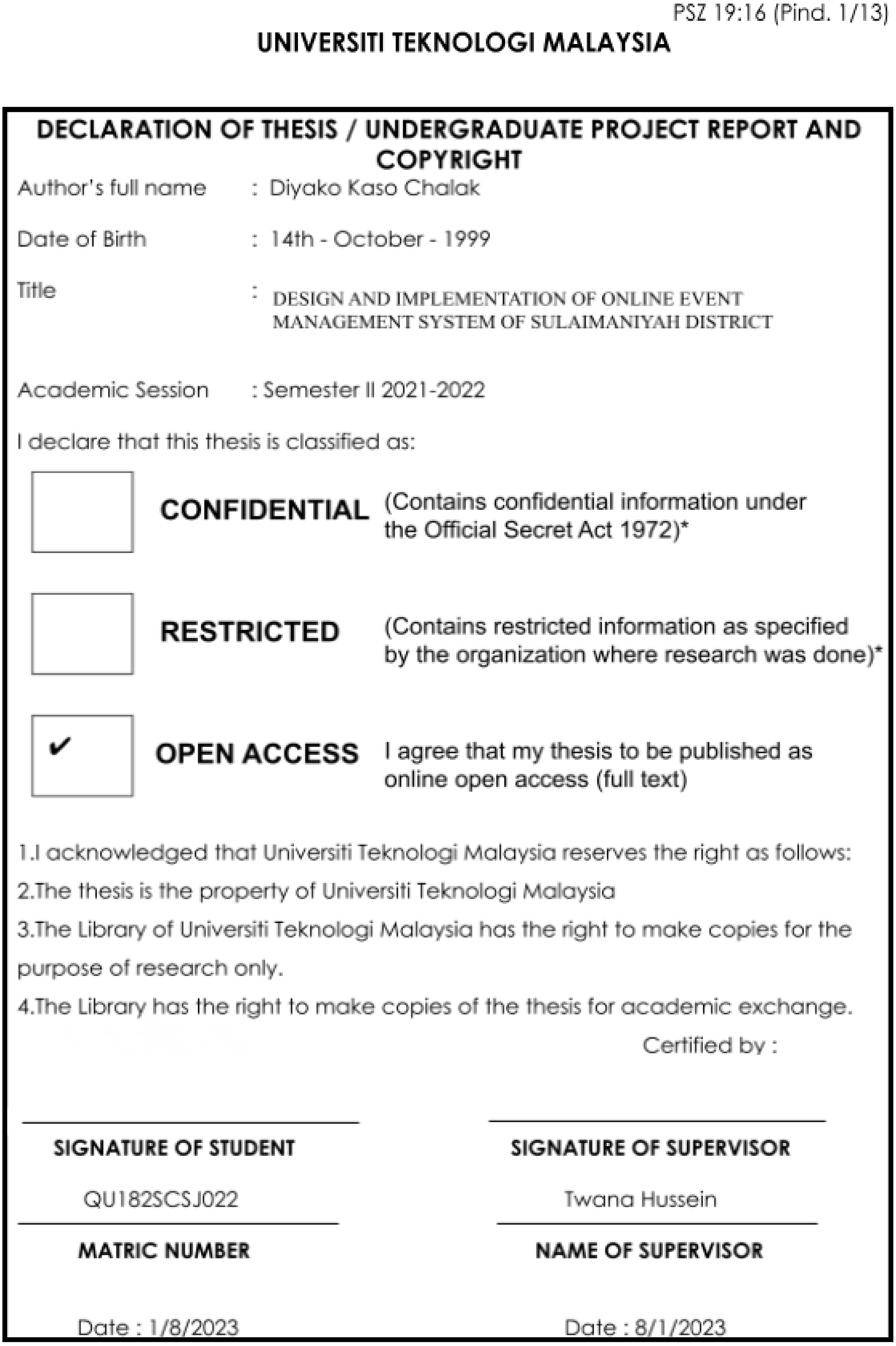
DESIGN AND IMPLEMENTATION OF ONLINE EVENT MANAGEMENT SYSTEM OF SULAIMANIYAH DISTRICT

Diyako Kaso Chalak

UNIVERSITI TEKNOLOGI MALAYSIA



PSZ 19:16 (Pind. 1/13)

UNIVERSITI TEKNOLOGI MALAYSIA

DECLARATION OF THESIS / UNDERGRADUATE PROJECT REPORT AND COPYRIGHT

Author's full name : Diyako Kaso Chalak

Date of Birth : 14th - October - 1999

Title : DESIGN AND IMPLEMENTATION OF ONLINE EVENT MANAGEMENT SYSTEM OF SULAIMANIYAH DISTRICT

Academic Session : Semester II 2021-2022

I declare that this thesis is classified as:

□ **CONFIDENTIAL** (Contains confidential information under the Official Secret Act 1972)*

□ **RESTRICTED** (Contains restricted information as specified by the organization where research was done)*

☑ **OPEN ACCESS** I agree that my thesis to be published as online open access (full text)

1.I acknowledged that Universiti Teknologi Malaysia reserves the right as follows:

2.The thesis is the property of Universiti Teknologi Malaysia

3.The Library of Universiti Teknologi Malaysia has the right to make copies for the purpose of research only.

4.The Library has the right to make copies of the thesis for academic exchange.

Certified by :

SIGNATURE OF STUDENT	SIGNATURE OF SUPERVISOR
QU182SCSJ022	Twana Hussein
MATRIC NUMBER	NAME OF SUPERVISOR
Date : 1/8/2023	Date : 8/1/2023

NOTES : If the thesis is CONFIDENTIAL or RESTRICTED, please attach with the letter from the organization with period and reasons for confidentiality or restriction

“I hereby declare that we have read this thesis and in my opinion this thesis is sufficient in term of scope and quality for the award of the degree of Bachelor of Computer Science (Software Engineering)”

Signature : ~~Twana H. Shareef~~

Name of Supervisor : Twana

Hussein Date : 25 June 2022

DESIGN AND IMPLEMENTATION OF ONLINE EVENT MANAGEMENT SYSTEM OF SULAIMANIYAH DISTRICT

Diyako Kaso Chalak

A thesis submitted in fulfilment of the
requirements for the award of the degree
of
Bachelor of Computer Science (Software Engineering)

School of Computing
Faculty of
Engineering
Universiti Teknologi Malaysia

JUNE 2022

DECLARATION

I declare that this thesis entitled *"Design and Implementation of an online event management system of Sulaimaniyah District"* is the result of my own research except as cited in the references. The thesis has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Diako e :

: Diyako Kaso Chalak

Date : 25 June 2022

DEDICATION

This thesis is dedicated to my father, who taught me that the best kind of knowledge to have is that which is learned for its own sake. It is also dedicated to my mother, who taught me that even the largest task can be accomplished if it is done one step at a time.

ACKNOWLEDGEMENT

In preparing this thesis, I was in contact with many people, researchers, academicians, and practitioners. They have contributed towards my understanding and thoughts. In particular, I wish to express my sincere appreciation to my main thesis supervisor, Mr.Twana Hussein for their guidance, advice and motivation. Without their continued support and interest, this thesis would not have been the same as presented here.

I am also indebted to Universiti Teknologi Malaysia (UTM) for funding my Ph.D study. Librarians at UTM, Cardiff University of Wales and National University of Singapore also deserve special thanks for their assistance in supplying the relevant literature.

My fellow students should also be recognised for their support. My sincere appreciation also extends to all my colleagues and others who have provided assistance at various occasions. Their views and tips are useful indeed. Unfortunately, it is not possible to list all of them in this limited space. I am grateful to all my family members.

ABSTRACT

This abstract presents the concept of designing and implementing an innovative online event management system, aimed at serving as an independent platform for event management in the Sulaymaniyah region. The primary goal of this system is to generate increased traffic and participation in the diverse range of events held within the city, while simultaneously promoting event awareness among residents and visitors. The proposed online event management system will offer a user-friendly interface with comprehensive functionalities to facilitate seamless event planning and organization. Event organizers will be able to create and manage their events through the platform, allowing them to reach a wider audience and increase event attendance. Attendees, on the other hand, will have easy access to a centralized database of events, enabling them to explore and engage in various activities occurring in Sulaymaniyah. The system will utilize cutting-edge technologies such as real-time notifications, personalized event recommendations, and social media integration to ensure effective event promotion and enhanced user engagement. Moreover, the platform will incorporate location-based features, enabling users to discover events happening nearby, further fostering community involvement and boosting local event participation. One of the key focuses of the proposed system is to promote lesser-known events that often struggle to gain visibility. By employing data analytics and user behavior tracking, the platform will intelligently showcase events based on user interests, preferences, and attendance history, thus elevating the visibility of a diverse array of events and enhancing their chances of success. To address potential challenges related to information accuracy and spamming, the system will implement rigorous event verification processes and user feedback mechanisms. This will ensure that only authentic and relevant events are listed, thereby enhancing the overall credibility of the platform. In conclusion, the online event management system aims to revolutionize event organization and promotion in the Sulaymaniyah region. By providing event organizers with an independent platform and offering attendees an accessible and comprehensive event directory, the system will drive more traffic to events and create a vibrant event culture within the city. Ultimately, this innovative solution seeks to connect people with their community and surroundings, fostering a sense of engagement and involvement in the diverse range of events held in Sulaymaniyah.

ABSTRAK

Abstrak ini membentangkan konsep mereka bentuk dan melaksanakan sistem pengurusan acara dalam talian yang inovatif, yang bertujuan untuk berkhidmat sebagai platform bebas untuk Pengurusan Acara di rantau Sulaymaniyah. Matlamat utama sistem ini adalah untuk menjana peningkatan lalu lintas dan penyertaan dalam pelbagai acara yang diadakan di dalam bandar, sambil pada masa yang sama mempromosikan kesedaran acara di kalangan penduduk dan pengunjung. Sistem pengurusan acara dalam talian yang dicadangkan akan menawarkan antara muka yang mesra pengguna dengan fungsi yang komprehensif untuk memudahkan perancangan dan organisasi acara yang lancar. Penganjur acara akan dapat mencipta dan mengurus acara mereka melalui platform, membolehkan mereka menjangkau khalayak yang lebih luas dan meningkatkan kehadiran acara. Peserta, sebaliknya, akan mempunyai akses mudah ke pangkalan data peristiwa berpusat, membolehkan mereka meneroka dan terlibat dalam pelbagai aktiviti yang berlaku di Sulaymaniyah. Sistem ini akan menggunakan teknologi canggih seperti pemberitahuan masa nyata, cadangan acara peribadi, dan integrasi media sosial untuk memastikan promosi acara yang berkesan dan penglibatan pengguna yang dipertingkatkan. Selain itu, platform ini akan menggabungkan ciri berasaskan lokasi, membolehkan pengguna menemui acara yang berlaku berdekatan, memupuk lagi penglibatan masyarakat dan meningkatkan penyertaan acara tempatan. Salah satu tumpuan utama sistem yang dicadangkan adalah untuk mempromosikan peristiwa yang kurang dikenali yang sering berjuang untuk mendapatkan penglihatan. Dengan menggunakan analisis data dan penjejakan tingkah laku pengguna, platform akan mempamerkan acara secara bijak berdasarkan minat pengguna, keutamaan dan sejarah kehadiran, sekali gus meningkatkan keterlihatan pelbagai acara dan meningkatkan peluang mereka untuk berjaya. Untuk menangani cabaran yang berpotensi berkaitan dengan ketepatan maklumat dan spam, sistem akan melaksanakan proses pengesahan peristiwa yang ketat dan mekanisme maklum balas pengguna. Ini akan memastikan bahawa hanya peristiwa yang sahih dan relevan disenaraikan, dengan itu meningkatkan kredibiliti keseluruhan platform. Kesimpulannya, sistem pengurusan acara dalam talian bertujuan untuk merevolusikan organisasi acara dan promosi di rantau Sulaymaniyah. Dengan menyediakan penganjur acara dengan platform bebas dan menawarkan peserta direktori acara yang boleh diakses dan komprehensif, sistem ini akan mendorong lebih banyak lalu lintas ke acara dan mewujudkan budaya acara yang bersemangat di dalam bandar. Akhirnya, penyelesaian inovatif ini bertujuan untuk menghubungkan orang ramai dengan komuniti dan persekitaran mereka, memupuk rasa penglibatan dan penglibatan dalam pelbagai acara yang diadakan di Sulaymaniyah.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Introduction

One of the most important signs of the livelihood of a society and the advancement of their cultures and mindsets is the number of activities that are done in that specific area or place that target multiple academic, arts, philosophical and musical tastes with the aim of nurturing the minds of that society and setting forth an environment where those tastes and abilities can grow and transform for the better. Events are an important part of the social life as it gathers multiple people from different academic and ethical and knowledge backgrounds together to interact on a common subject or theme in which the exchange of ideas might lead to brighter ideas and the networking among those people might lead to better life opportunities regardless of the event that has its own importance to open up the minds of the society and to drive them to a better destination to guide their train of thoughts.

Kurdistan Region in Iraq is a culturally rich region as it stands in the middle of the Iran , Turkey and Iraq those regions had interactions with our region throughout history and our cultures to some extent have borrowed beautiful cultures from each other whether they are in music , arts , Islamic arts , literature and Theatres and Clothing those culture borrowing is reflected in our culture and the activities that we do. Approximately all of Kurdistan is constantly full of Activities and events held to drive the society forwards and more specifically Sulaymaniyah city as it is the Capital of Culture and is the spearhead of cultural and artistic activities and events as most of the well-known and respected poets and literature writers come from this region and most of the activities are to be conducted in the city of Sulaymaniyah.

Unfortunately till now Kurdistan region and more specifically Sulaymaniyah City has not yet utilized the technological advancements to better improve the way the events are held and people to be notified. Tens of events are held weekly but only a little portion of the society knows about it due to the fact that events are published in the Bazaar by posting a big flyer that states the activity's name and the venue and the date which is not really efficient and the problem is in the knowledge background of the community members as they have not yet utilized the event publishing functions on the social media platform due to very poor understanding to those event publishing functions to be viewed by the majority of people of all kinds of interests and backgrounds which leads to a limited amount of participants who knew about the event and in turn leads to a very weak cultural movement. Hence the (Design and implementation of online event Management system**)** is Proposed.

Fortunately, technological advancements have led to many wonderful things that can be utilized in the favour of Human Beings such as Mobile Applications, Web Applications and Desktop Applications which all can make human lives easier and to allow us to better utilize the opportunities that we have in front of us. The (Design and implementation of online event Management system) will be a system that will address the issues mentioned above and will allow users of all sorts of backgrounds and technical knowledge to create their own accounts on the system and gives them the capability to create events by specifying the name of the events, the venue of the event and the date of the event as well as the description of the event to let the users know what this event is all about. The system will categorize the events based on specific titles and topics such as (art, theatre, conferences, fashion, festivals, sports, etc…) to make it easier for the person browsing through the system to better find what they are looking for. The system shall also allow the users to book their attendance for a specific event and to have their names on the list of the event. Our system will have an e-wallet payment method to buy tickets to certain theatre shows and movies or pay a fee for an event in general. Our system shall be the rise of the cultural movement by letting the majority of people know about how alive this city is in terms of the events it holds and to attract most of the population to attend these activities and events and in the future this system can be utilized in the touristic movement by offering multiple language support for the tourists to increase the tourism traffic to our city.

1.2 Problem Background

In the city of Sulaymaniyah, Typically the events are known and published by posting flyers and flex papers that state the name of events and the venue and also the date of the event in the Bazaar. Hanging flyers or flex papers in the Public places is not the most efficient way to let people know of the event and will not ensure a large number of participants to attend the event due to the following facts. First the public places might have tons of hanging flyers that will limit the number of people who will view the flyer to be notified of the event. Secondly the pedestrians (those who walk) might not focus on reading each and every flyer they see on the streets which will in turn lead to poor viewers. Thirdly the Pedestrian in the best case even if they saw the flyer they might only take a picture of it and then they might forget about it and the picture will be buried in tons of other pictures.

(Design and implementation of an online event Management system) system is the solution by driving enormous traffic to the events as many people will rely on one platform to view all the events available out here and will find their desired event based on the categories. It will act as a marketing platform that will market the events that have a fee on them and will help those organization to make a living with the events that they have organized such as movie festivals and theatres and artistic exhibitions which can be in the form of a ticket or a small entrance fee by incorporating an e-wallet payment methods .The system shall also allow various different people of different organizations and groups to create the events that they have organized in order to publish it and make the people aware of their event. The people will also be notified of the event by having the option to write the event on their google calendar accounts on their phone automatically from the system in order to remind them a day or two before the event or on the event day.

1.3 Project Aim

The aim of this project is to re-ignite the cultural movement in the city by developing an efficient and easy to use (Design and implementation of online event Management system) system to drive more traffic to the events and to make people aware of the events in order to increase the networking opportunities and to expand the opportunity spaces beyond the stereotypical jobs and ways of making a living and the classical way of networking with the aim of paving way for an opportunity filled city that has opportunities on every corner and every person from every background and major can make contribution to the cultural and artistic and academic movement in our city.

1.4 Project Objectives

The objectives of the project are:

(a) To elicit requirements of the Online Event Management System Of Sulaimaniyah District

(b) To design the Online Event Management System Of Sulaimaniyah District based on the requirements of the system

(c) To develop the Online Event Management System Of Sulaimaniyah District to address the requirements

(d) To test Online Event Management System Of Sulaimaniyah District for completeness and correctness

1.5 Project Scope

The scopes of the project are:

(e) The system will be developed only for the City of Sulaymaniyah in the initial phases

(f) The system development will use Mobile Application languages and frameworks such as dart programming language and the flutter framework for mobile application development.

(g) The system will be developed using Visual Studio Code code Editor.

1.6 Project Importance

I decided to develop the (Design and implementation of event creation and notification) system because I was one of those who wanted to be aware of the events that are taking place in our city but there wasn’t a single standardized way to browse the events so I can decide which one I want to attend.

What Motivated me to work on this project is the fact that this project is even bigger that what I would have wanted and can directly impact the city and change how things are taking place and this idea has so much potential in the future to be enlarged and transformed to a better idea by incorporating many other features in it such as e-wallet and payment methods and to drive traffic to the events and to encourage tourists to also participate In those events so we can all drive the cultural , toursitic , artistic and Academic movement forward and to see my city strive.

The benefits of this system can be seen in driving enormous traffic to the events held, providing an easy e-payment method to pay for an event or pay for tickets, allowing people to be aware of all the events around them and paves way for more publicity for organizations and artistic and academic groups.

The parties who will benefit from this system is potentially everyone as the participants and the organizers all will significantly benefit from this idea due to the fact that the organizers will better publish and publicize their events and it will draw in more participants and the users of our system will better browse and find their desired events to attend which increases the likelihood of them attending at least 1 event per week.

1.7 Report Organization

To summarize the above points and paragraphs, Kurdistan regional government and the City of Sulaymaniyah is lacking the proper methods of publishing events and attending it as it sees itself only in the form of street flyers publishing that leads to poor attendance of such events and in some cases not being aware of the event taking place. (Design and implementation of online event Management system) will work to change the way events are being published and to ease finding the desired events by browsing the system based on categories and will help the organizers gain more attendees and publish their events in a much more efficient manner. The system shall drive huge traffic to the events. In the next chapters we shall discuss the methodologies of development and the structural and behavioural models of the system and functional and non-functional requirements of the system and the whole explanation of the system shall end in chapter 5.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Event management systems are nowadays being used extensively to manage events of all sorts and types. The most important factor that led to the mass public usage of such event systems is innovation in live event technology (Brandon,2018). Event management systems are defined as systems that are used to manage and organize activities related to a certain event or events (Chrisantus Oden , n.d.). An event management software helps event organizers to register their events and to allow the customers to book their attendance (check-in) and to promote their events on the software. Some event management software only specializes in a type of functionality such as participant booking (check-in) this leads to a limited amount of functionality. Typical operations of an event management System are but not limited to (Event Registration, Event Marketing) (Brandon, 2018).

In this chapter we shall discuss the current systems that exist that have similar functionalities to the (Design and implementation of online event Management system) that have been developed and implemented in various other domains. This chapter is going to discuss those systems in terms of their domains and where have they been implemented and what are the services that are being provided by those systems and how these systems compare with each other in terms of their functionality as well as their negative and positive aspects.

The technical aspects of the development of our own proposed system shall be discussed in terms of the programming language that is going to be used to program the functionalities of the system, such as dart programming language and the flutter framework and such technology shall be discussed, labelling their performance and security and the features that they are possessing.

2.2 Comparison between existing systems

The online event management system that we have proposed is not the first idea to be implemented in such domains but rather there have been multiple systems developed that were implemented in various domains with various types of characteristics and functionalities. The domain of event management is so broad and the implementation can be very broad ranging in all types of events that exist in the community to a very narrowed down domain that can include one type of event such as (sport , movies , workshops , conferences , etc…) the implementation of these systems will provide a guideline and a broader view into the characteristics of such systems and will guide the path of development and potentially adding characteristics from these systems and to be improved further based on the original idea of the systems. An implementation of these systems are the following:

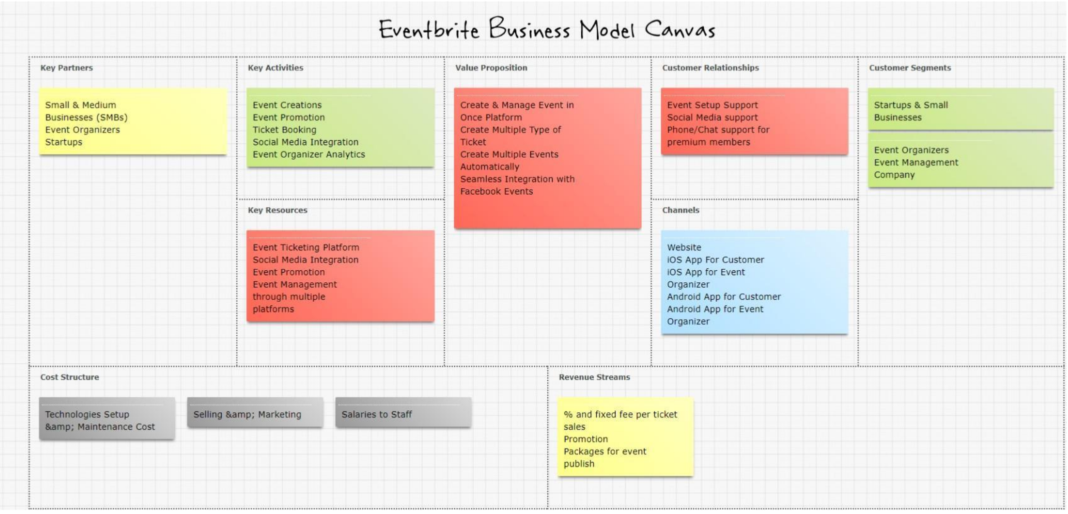
1. Eventbrite mobile application
2. Fever mobile application
3. Eventzilla mobile application
4. Facebook Events Mobile Application

2.2.1 Eventbrite Mobile Application overview.

Due to the diverse implementation of event management systems , the implementation differs from domain to domain. The Eventbrite system apart from managing events it is a platform that specializes in ticket sales to event organizers. Eventbrite is one of the largest ticketing platforms that is successful. Many event organizers worldwide use platforms like Eventbrite to manage their events and to promote their events and to boost their ticket sales for their paid events across many platforms such as web and mobile. Eventbrite platform only makes money when users book a paid event by booking (buying) tickets for the event (Gaurav, 2018).

Eventbrite event ticking plans come in three packages which are the premium package, the professional package and the essential package. For the essential package the Eventbrite platform charges 1% for every ticket purchased by the customer added to it is 99 cents per paid ticket this means if a ticket price is 10$ then the amount of money that Eventbrite makes is 1.09$ (Gaurav, 2018).

Most event organizers use Eventbrite because it makes the process of managing tickets, invitations, reminders, attendee's details and event promotion much easier Eventbrite can be integrated with other social media platforms such as Facebook and the organizer's followers can be aware of the events through Facebook events and book through Facebook. As a conclusion Eventbrite is a platform to set up events and sell tickets online (Gaurav, 2018).



Eventbrite Business Model Canvas

Key Partners

Small & Medium
Businesses (SMBs)
Event Organizers
Startups

Key Activities

Event Creations
Event Promotion
Ticket Booking
Social Media Integration
Event Organizer Analytics

Key Resources

Event Ticketing Platform
Social Media Integration
Event Promotion
Event Management
through multiple
platforms

Value Proposition

Create & Manage Event in
Once Platform
Create Multiple Type of
Ticket
Create Multiple Events
Automatically
Seamless Integration with
Facebook Events

Customer Relationships

Event Setup Support
Social Media support
Phone/Chat support for
premium members

Channels

Website
iOS App For Customer
iOS App for Event
Organizer
Android App for Customer
Android App for Event
Organizer

Customer Segments

Startups & Small
Businesses

Event Organizers
Event Management
Company

Cost Structure

Technologies Setup
&amp; Maintenance Cost

Selling &amp; Marketing

Salaries to Staff

Revenue Streams

% and fixed fee per ticket
sales
Promotion
Packages for event
publish

Figure 2.1: Eventbrite Business Model Canvas

The key points to notice in the Eventbrite business model canvas are the (key activities) which state the main activities and operations of the system and the (value proposition) which is the value the system gives back in return to the customer for using the system. The main features of this system are (event creation, event promotion, ticket booking, social media integration and event organizer analytics). The values that this system gives in return are (Creating and managing events in one platform, Creation of multiple types of tickets, Create multiple events, seamless integration with Facebook events or any other social media application).

2.2.1.1 Eventbrite advantages

One of the main advantages is the fact that this system allows the user to post and create events free of charge despite having the pricing plans.it is easy to use by users who are using the system for the first time in a way that is easy to choose the event and enter your email and pay using online methods if needed very easily. The Eventbrite notifies the users of the events that they have chosen to attend by creating reminders for the date of the events.

2.2.1.2 Eventbrite disadvantages

Disadvantages of Eventbrite it does not have a special module for handling the online events and mainly focuses on physical events that are taking place.it lacks the integration with CRM software, CRM software are software that equips you with tools that allows you to automate sales and contact vendors and asset and resource managements but Eventbrite lacks the integration with those types of software. The event payment is only given when the event ends, this can cause a problem when an event organizer wants to gain revenue from the tickets in order to promote the event for the upcoming months.

2.2.2 Fever mobile application overview

Fever (Fever Up) is an entertainment discovery platform which helps both the traveller and the local people to find events and book them with the intention of finding the best events and experiences in a specific city. These events can range from concerts and art exhibitions to cinema and Food Festivals. Fever offers services in many countries and it focuses on notifying the users of the newest social events and entertainment events happening in their city. Fever is a platform that provides popular events held by third parties in a specific city as well as the events organized by individuals. Fever enables you to browse lists of events available in a specific city as well as applying filters to the events you are browsing by applying category, date and area filters and you can also post your own events and details, the events are put into categories which are (Candlelight concerts, concerts and music festivals, activities and games, food experiences, theatre and comedy, cinema, culture and arts) ((Raeesha,2022).

2.2.2.1 Fever advantages

A few Advantages of Fever is that Fever is relatively new but it is one of the leading platforms today offering events that is in tune with everyone's taste and preferences from food enthusiasts to music lovers The majority of those experienced an event with fever up have enjoyed it and left satisfying reviews of their experiences and indicated that those events were well organized and fun and exciting (Raeesha,2022).

2.2.2.2 Fever disadvantages

A few disadvantages of fever are When the event organizers book an event in Feverup even if for any reason their event gets cancelled or delayed then Feverup does not offer any kind of money refund to the organizers but only provides a voucher that allows the Organizers to organize another event for the same amount of money previously paid and no need to pay a second time for this event in case the first event got cancelled or delayed. The customer support has been indicated as very poor due to the majority of the reviews that have been written on the matter of customer service indicating that the customer service is unresponsive and sorting and making changes in any existing event or booking would take a very long time to be processed by the customer service unlike a few customers who said that their enquiries have been processed quickly (Raeesha,2022).

2.2.3 Eventzilla Mobile Application

Eventzilla is an event management system that automates the process of creating and registering for an event in an efficient way. EventZilla provides the event organizer with easy user interfaces to manage attendees lists and create landing pages and screens in which the attendees can sign up for a particular event with ease. The event organizer is given complete freedom to organize the attendee form in which the organizer can create questionnaires and input fields of all sorts to get the necessary information from the attendee such as special food requests or song requests to the DJ at a party event. Eventzilla allows the organizers to send emails to the attendees based on the type of their ticket to promote their event further in which the organizer can choose the ticket type for their event whether free or paid event. (Lizzie, 2019).

2.2.3.1 Eventzilla advantages

A few advantages of Eventzilla is that Eventzilla’s user interface is very easy to get into as the basic layout is straightforward. Eventzilla’s event creation functionality lets the organizers create events with ease by incorporating maps, audio-visuals, texts and pictures. Event Signup is easy by publishing the link of the registration form to the users and the users can easily register and book their attendance for that event. Eventzillas payment method is easy as the system is integrated easily with e-payment methods such as PayPal and Stripe.

2.2.3.2 Eventzilla disadvantages

A few disadvantages of Eventzilla is the amount of options it has for customization and tweaking the settings for tickets, events, and information that sometimes causes confusion to the customers especially those who have not spent much time with the system. The URL that is assigned to an event cannot be changed and modified by the users which causes frustrations when the organizers want a simple customized URL to promote their events. The payment operations sometimes have bugs and need further maintenance and modification.

2.2.4 Facebook Events Mobile Application overview

Facebook is one of the most popular social media platforms today to be used to connect and communicate with customers and consumers. Many brands prefer Facebook over the traditional microblogging or microsites to show their presence online instead they resort to Facebook pages. Facebook offers many functionalities to its users whether businesses or individuals and one of the examples of these functionalities is the Facebook Webinars. Facebook webinars are an effective tool due to the fact that it feels like a one-on-one interaction in which it feels personal to the users and can be viewed by thousands of other customers to engage and get more information and have sessions like Questions and answers for the customers. The webinar can be initiated by specifying the date of the seminar and to choose an interesting title that reflects the nature of the seminar and then choosing whom you'd like to send the invitation to. (Marie, 2012).

2.2.4.1 Facebook Events Advantages

A few advantages of Facebook events are attendees of a certain event can alert and notify and invite their Facebook friends of the upcoming event. Facebook events can be used as a market research tool in which you can learn more about your customer's preferences and views and likes based on the comments that they have left or the engagement type of the customer. The archive functionality allows you to save everything in the "past event" section. (Marie, 2012).

2.2.4.2 Facebook Events disadvantages

A few disadvantages of Facebook Events are Facebook events allows you to only work with one time zone meaning your webinar or event will be shown based on the time zone of one country and this can cause problems if your organization is international and hosts global events to many different countries. Facebook removing messaging all your page followers has restricted the promotion of the events that are taking place and brands and organizations need to rely on other means to promote their events or to use Facebook's own advertising means to advertise for their events. At the beginning of the social media presence of the organization or the brands the event participants will be very small or in some cases non-existent due to the events and webinars being virtual. (Marie, 2012).

Table 2.1: Comparison between proposed system and existing systems

	Eventbrite App	Feverup App	Facebook Events App	Eventzilla App	My System
Main uses	Event Creation and Ticket sales	Browsing and creating Events	Creating virtual events such as webinars and options to notify people of physical events taking place	to create events and manage event fee payments and to promote events via email.	To browse the and create events and have the functionality for Ticket sales
Pricing packages	Free, Premium , Professional , Essential	Free, Special member , Exclusive member	None , for event promotion need to use Facebook Advertisement	Free , Paid	Not Currently Chosen
Usability	Good user interface you can intuitively navigate through the system	Easy to start using the system but to master the functionalities prior training videos are encouraged	Easy to use by users of social media as the options are relatable	Overall easy to use but some settings have many options which causes confusion	The system shall be easy be use with an intuitive User Interface

Main Features	Creating events and selling tickets with ease for your event	Creating events and browsing events based on categories in specific cities and booking special feverup events	Creating events whether physical or virtual and to invite Facebook friends and Facebook page followers to your online and physical events.	Creating events and managing the payment for such events through the use of PayPal and Stripe payment methods and promotion of events through email	Creating Events with the main focus on physical events and a potential support for online events
Event/Task Creation Features	Yes	Yes	Yes		Yes
Browsing Events/Tasks	`Yes	Yes	Yes	Yes	Yes
Event/Task Filtration	Yes	Yes	Yes	Yes	Yes
Cloud Storage Limitations	Yes 2,000 Calls per Hour	No	No	No	No
Invoice and Receipt Generation	Yes (sent through email)	Yes	Yes	Yes	Yes (will be sent through email)
Communication tools such as Chatboxes	Yes	Yes	Yes	No	Yes Especially for Q&A sessions
E-commerce Tools	Yes, such as embedded code to sell on websites and the use of	No	No	No	Not chosen yet

	promotiona l codes for discounts and selling on partner websites				
Event Promotion and notifications	Yes	Yes	Only via Facebook Advertisements	Yes	Yes
Social Media Integrations	Yes	No	Yes but only with a business page or account	Yes	No
Supports online Payment	Yes	Yes	Yes for virtual events as well.	Yes	Yes
Customer Support	Yes	Yes	Yes	Yes	Yes

2.3 Literature Review of Technology Used

2.3.1 Flutter Framework for Mobile Application Development overview

The flutter framework is a cross-platform framework that was designed and developed by the Google Inc. that allows the developers to use one codebase for multiple platforms and providing "Native" performance on those platforms, this enables the developers to run their codes on multiple platforms instead of creating "Native" applications for multiple platforms which leads to managing multiple codebases and a much higher technical cost shall be paid by the companies. (Sebastian, 2020).

2.3.2 Development using Flutter Framework for Mobile Application Development

in essence the Flutter framework reduces the cost of development and maintenance by developing one codebase that runs on multiple platforms, the way this is done is by creating platform specific flutter widgets on the flutter app that offer the “Native” functionality of the target platform allowing developers to access the hardware functions and services of the target platform by manipulating the operations of the (audio , sensors , cameras , etc…) by using the dart programming language as the main programming language of the flutter application. (Sebastian, 2020).

2.3.3 Dart Programming Language overview

Dart programming language is a class-based, object-oriented language that was designed and created by Google Inc. as a replacement for JavaScript with better performance and extensibility to be implemented in large-scale google projects. Dart programming language has many advantages in terms of its structure and usage over JavaScript in which Dart offers re-usability of code which leads to much less development time and cost and offers increased performance. Dart can be run on web browsers as it automatically converts to JavaScript when used in the Web context.(Afaf,2020).

2.3.4 Programing using Dart Programming Language

Dart programming language is used to create web, backend, mobile applications. Dart programming language needs to be written inside of the flutter app in a code editor and a virtual device emulator such as Android studio. Dart is very similar to the other object oriented programming languages such as Java, C, C++ and offers various runtime options for different platforms. Apart from offering the typical object oriented data structures and control flows used in object oriented programmes, Dart programming language offers a variety of data structures from various programming languages such as (Lists, Maps, Arrays, Hashmaps, etc…). (Sanjib,2020).

2.4 Chapter Summary

In this chapter we discuss the different implementations of the Online Event Management System and review all the current existing systems of those sorts and how they were implemented. Eventbrite in event creation and management and providing ticketing sales, Feverup that provided users with the options to freely create events and publish them and Eventzilla with the event promotion and creation capabilities and its integration with e-payment methods and lastly Facebook events that is used by social media users to create physical and online events and notify their page followers and the Facebook user can invite their Facebook friends.

The systems were reviewed based on their descriptions of what those systems were and what operations they performed and we discussed their advantages and disadvantages in terms of their functionality and the opinions of the customer were taken into consideration as well.

We finally discussed the technologies that will be used in my current system and have discussed what they are and how do they work by referring to scientific and technical specification documents

CHAPTER 3

SYSTEM DEVELOPMENT METHODOLOGY

3.1 Introduction

Every software must start with a clear and focused goal in mind and a plan to execute the development. Every process development goes through a similar plan or a flow of operations and processes; these processes are called the software development life cycle (SDLC). The usage of Software development life cycle in development processes lead to a clearer process of operations and reduces time of development and reduces the cost of development. Software development life cycle is very beneficial for the overall environment of the workspace as it sets the responsibilities of each development team and provides communication channels between the stakeholders of the system as well as the development team. However, over the years many frameworks for the software development life cycle have been introduced in order to tackle different project requirements and complexities as the SDLC cannot accommodate all types of projects on its own (Jory, 2019).

In this chapter we shall discuss the chosen methodology that has been chosen for the development of the (Design and implementation of online event Management system), what are three processes that are involved in the chosen methodology discussed step by step and where each part of the project fits in within the processes described. The technologies that shall be used for the development of the project shall be discussed and explained as to how it may be used and what part of the system it shall be used in. The software and the hardware requirement required for the execution of the project on the user's end device shall be discussed as well. Finally, the summarization of the chapter will be labeled at the end of the document to review the document in terms of what has been discussed.

3.2 Methodology Choice and Justification

Software development methodologies can be considered a solution that addresses the business requirement of the business community in terms of faster and more lightweight methods of software development especially in the rapidly growing Internet software and mobile application industries. The Agile methodology movement has agreed on a unified approach to the agile methodology by establishing a manifesto called the "Agile Software Development Manifesto". (Abrahamsson et al, 2002).

The chosen Agile methodology framework used for the project development is the (Scrum). Scrum simply put is a process control framework of development in which teams use to develop products in iterations each iteration a shippable product is created called an "increment" which corresponds to a software feature in an iterative process called "sprint". short-cycle development and product testing is the main focal point of Scrum framework in which it emphasizes the continuous learning and continuous testing and continuous review of each increment developed by the team and simplifies complex problems to much simpler ones (Larman & Vodde , 2016).

Scrum Master is responsible for how the process of development is being executed and ensure that the Scrum processes are going smoothly and that the methodology for the product has been followed with the aim of improving the product quality of the product developed (Owens, 2014).

Product Owner is the person who is responsible for setting and clarifying the necessary requirements for development and to make sure the team is on the right track and determines "what" to be done and "when" is the deadline but not responsible for "how" it should be done (Layton, 2015).

Scrum Development Team is Responsible for developing the product and creating the increments of the product based on the plans planned for in the product backlog of the Sprint Cycle and the team is responsible for designing and planning and implementing the increments as the teams are cross functional who have diverse technical backgrounds and skills(Owens,2015).

Customer or the stakeholder is the person who is impacted or can impact the project requirements. The stakeholder can be from within the organization or can be from outside of the organization or can simply be an investor in your project (Layton, 2015).

Based on the above definitions that have been mentioned regarding the Scrum Agile Methodology framework the reason why this framework has been chosen is because of the flexibility it offers in terms of its operations and process and the way the framework utilizes the time and resources that are available and plans for every single iteration and feature in its product backlog. Scrum organizes the tasks that needs to be achieved in a Sprint backlog in which the sprint (plans the task , builds the task , tests the task and reviews the tasks) for each one of the features or set of tasks which is very preferable as you will have plenty of focus on one aspect of the development instead of wandering around all the features and attempting to tackle them all at once, this ensures maintainability and versatility in the development process by continuously reviewing and planning and testing the product increment in iterations increases both the reliability of the system and the management of the process of development.

3.3 Phases of the Chosen Methodology

The phases that make up the Scrum Agile Methodology Framework consist of 3 phases which are (Pre-Game Phase, Development Phase, Post-Game Phase) as stated in (Abrahamsson et al ,2002).

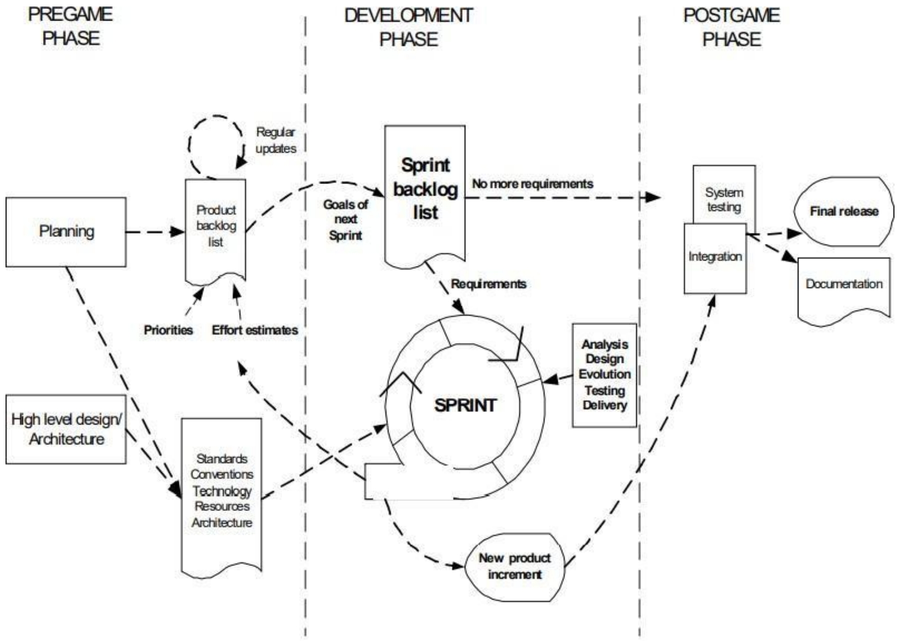


Figure 4. Scrum Process

Figure 3.1: Processes inside of the Scrum Agile Methodology Framework received from (Abrahamsson et al, 2002)

3.3.1 The (Pre-game) Phase of Scrum

The pre-game phase mainly consists of two sub-phases that consist of (planning, architecture and high level design). In the planning phase all the current requirements that are known that may come from the customers or the developers or any other stakeholder are put in units called product backlog. (Abrahamsson et al, 2002).

The product backlog is a list of requirements and features and fixes that have been provided and organized by the product owner who is responsible for putting those requirements into the product backlog that ultimately acts as the “to-do” list of the development team working on the system that contains the necessary requirements that needs to exist in the system (Claire, 2018).

In the architecture phase the system architecture is chosen based on the tasks and items of the product backlogs these designs are then reviewed and a meeting is held in order to agree on one of the proposals that are presented .If the tasks needed to be done are a modification to an existing system then those changes are put into consideration and the risks and the problems they impose are evaluated in order to ensure the modifications will not cause any defects in the system (Abrahamsson et al , 2002).

In (Design and implementation of online event Management system) the pre-game phase will consist of the Customer requirements which are going to be the problems faced by the community in terms of the event creation and attendance of those events and their problems are going to be turned into user stories for more clarification and put in the product backlog in which it contains the set of the system features such as the (payment method) , (categories of the events) , (event management options such as creating events) along with the customer requirements elicited to turn into features and each feature shall be simplified to smaller and more manageable tasks and their priorities and deadlines shall be set within the 2-3 weeks sprints and the architecture phase will depend on the architecture that the system shall follow.

3.3.2 The (Development) Phase of Scrum

The development phase which is often referred to as the “game phase” is the most unpredictable part of the development phase in which the factors of (development time frame, system quality, task requirements, needed resources, the implementation technologies and tools needed) play a very risky and important part in the development decisions as these factors may often be changed during the development (Abrahamsson et al , 2002).

Sprint planning is the process of discussing and labeling the priorities of the product backlog tasks after they have been organized and labeled and the team starts with the highest priority tasks in the product backlog and determine how to achieve the goal of the task by asking the product owner for clarifications on the tasks to get a clearer understanding and get the bigger picture (Jason, 2022).

The development team then chooses the features and tasks that are going to be developed and worked on in the current sprint cycle and are organized and put in the sprint backlog. (Claire, 2018).

The act of developing and achieving the features is called Sprints In which each Sprint aims at developing a small increment to the existing increments by producing and building the current functionality labeled in the sprint backlog for implementation and a new product increment will be added to the finished increments. (Abrahamsson et al , 2002).

Daily scrum is another one of the activities that happen during the sprints in which the team gather around and have a meeting for a short period of time in which the team members explain and discuss their current achievements and progress that they have reached so far as well as the obstacles they are facing in order to brainstorm the solutions and keep the team members on track and up to date with the project progress (Claire,2018).

In this part of the development process the (Design and implementation of online event Management system) shall have its tasks and features ready and each feature is going to be planned and simplified to simpler tasks and put in the sprint backlog to be developed in the period of 2-3 weeks in order to add to the increments of the system incrementation by incrementation to ensure that each feature is planned out well and built and tested and reviewed before moving on to the next iteration for developing the next increment.

3.3.3 The (Post-game) Phase of Scrum

The post-game phase is the final phase of the development of the system and the following tasks that shall be conducted are (system testing, Integration testing, sprint review , sprint retrospective) as well as documenting the system after that the system shall be ready to be delivered (Abrahamsson et al ,2002).

Sprint review is the process of reviewing the current progress reached by the development team in which the demo of the system is presented to the peers and the customers along with the sprint backlog items that have been done and developed then the product owner can review the presented demo and decide whether or not to release this increment and the backlog items that are left unchecked shall go back to the product backlog to be re-planned and prioritized for the next sprint cycle (Claire , 2018).

Sprint retrospective is the activity of pondering over the mistakes that happened during the previous sprint and to plan out and improve those steps as well as taking into consideration the customer feedback and to improve the user stories provided by the customers to prepare for the next sprint cycle (Jason, 2018).

In the post-game phase the (Design and implementation of online event Management system) project shall be checked for completeness of their requirements and also will be tested in order to ensure its conformance to the previously mentioned requirements stated by the stakeholders using the (system testing) and to check whether the components of the system are interacting well with each other especially in terms of their interfaces by conducting the (integration testing) on the system if those tests have passed successfully and the mistakes done during the sprint cycle shall be reviewed and improved for the next sprint cycles and the user stories in the sprint retrospective shall be pondered upon and improved to ensure a better and more efficient development in the next sprint cycles.

3.4 Technology Used Description

3.4.1 Flutter Framework for Mobile Application Development overview

The flutter framework is a cross-platform framework that was designed and developed by the Google Inc. that allows the developers to use one codebase for multiple platforms and providing "Native" performance on those platforms, this enables the developers to run their codes on multiple platforms instead of creating "Native" applications for multiple platforms which leads to managing multiple codebases and a much higher technical cost shall be paid by the companies. (Sebastian , 2020).

3.4.2 Development using Flutter Framework for Mobile Application Development

in essence the Flutter framework reduces the cost of development and maintenance by developing one codebase that runs on multiple platforms, the way this is done is by creating platform specific flutter widgets on the flutter app that offer the "Native" functionality of the target platform allowing developers to access the hardware functions and services of the target platform by manipulating the operations of the (audio , sensors , cameras , etc…) by using the dart programming language as the main programming language of the flutter application.(Sebastian, 2020).

3.4.3 Dart Programming Language overview

Dart programming language is a class-based, object-oriented language that was designed and created by Google Inc. as a replacement for JavaScript with better performance and extensibility to be implemented in large-scale google projects. Dart programming language has many advantages in terms of its structure and usage over JavaScript in which Dart offers re-usability of code which leads to much less development time and cost and offers increased performance. Dart can be run on web browsers as it automatically converts to JavaScript when used in the Web context.(Afaf,2020).

3.4.4 Programing using Dart Programming Language

Dart programming language is used to create web, backend, mobile applications. Dart programming language needs to be written inside of the flutter app in a code editor and a virtual device emulator such as Android studio. Dart is very similar to the other object oriented programming languages such as Java, C, C++ and offers various runtime options for different platforms. Apart from offering the typical object oriented data structures and control flows used in object oriented programmes, Dart programming language offers a variety of data structures from various programming languages such as (Lists, Maps, Arrays, Hashmaps, etc…).(Sanjib,2020).

3.5 System Requirement Analysis

Minimum requirements

a) Processor : Quad-core 2.4 Ghz
b) Chipset : Exynos 1280
c) Ram : 2 GB
d) Storage : 64 GB

e) Network : LTE

Recommended requirements

f) Processor : Octa-core 2.4 Ghz
g) Chipset : Qualcomm SM7325 Snapdragon 778G 5G
h) Ram : 4 GB
i) Storage : 128 GB
j) Network : 5G

3.6 Chapter Summary

In this chapter we have introduced the agile development methodology and the agile development methodology framework of Scrum in which we discussed the overview of the Scrum framework as well as the justification as to why we chose that specific framework then the phases inside of the Scrum framework was discussed that is generally divided into 3 section (pre-game , development , post-game) phases in which inside each section we have their own activities such as (product backlog , architecture , sprint planning , sprint review , sprint retrospective) in the end of the document we have labeled briefly the technology that is going to be used for the development of the system as well as the necessary software and hardware requirements to run the Mobile application.

CHAPTER 4

REQUIREMENT ANALYSIS AND DESIGN

4.1 Introduction

in this chapter the structural and the behavioral aspects of the (Design and Implementation of Online Event Management System) shall be discussed by putting forth the modeling of the system using the Unified Modelling Language (UML) Diagrams in which each type of diagram describes a different aspect of the system in terms of its functionality , in terms of the system's overall structure and finally the relationship between the database entities that reside inside of the system by using (Enterprise Architect) modeling tool.

4.2 Requirement Analysis

for this part the main diagrams that have been used, describe the behavioural aspect of the system in terms of the functionality that the system shall offer, we begin by showcasing the (Use Case) diagram which comprises of the general view of the system's functionality and we dive in further by incorporating (Sequence diagrams) and (Activity diagrams) to explain the functionalities on a deeper level.

4.2.1 Use Case Diagram of the System

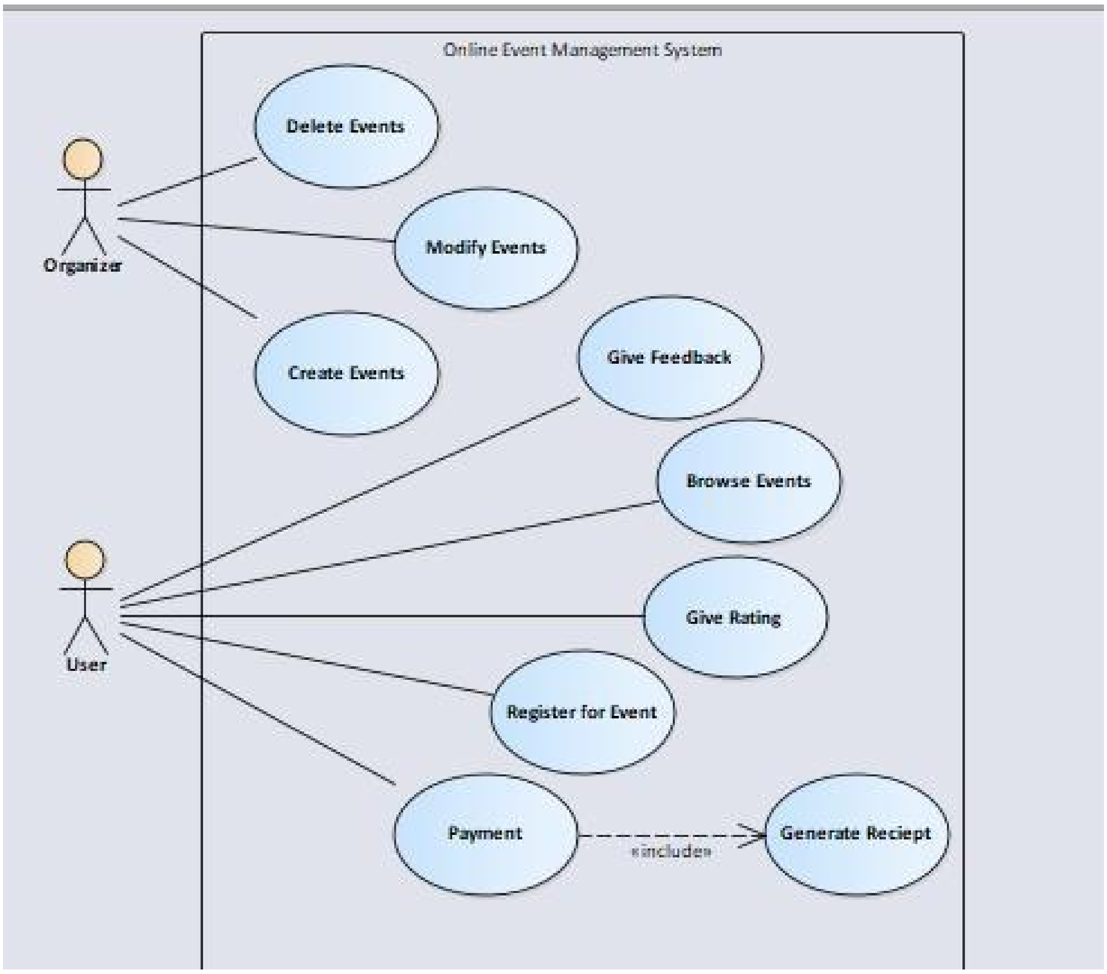


Figure 4.1: Use Case diagram of the System

the above diagram is the use case diagram of the system that contains two types of actors (Organizer) and (User) in which the Organizer can (create , delete , modify) event whereas the User can (browse events , register for an event , make payments and get a receipt through generate receipt)

4.2.2 Sequence Diagrams of the System

4.2.2.1 Sequence Diagram for Modify Events Use Case

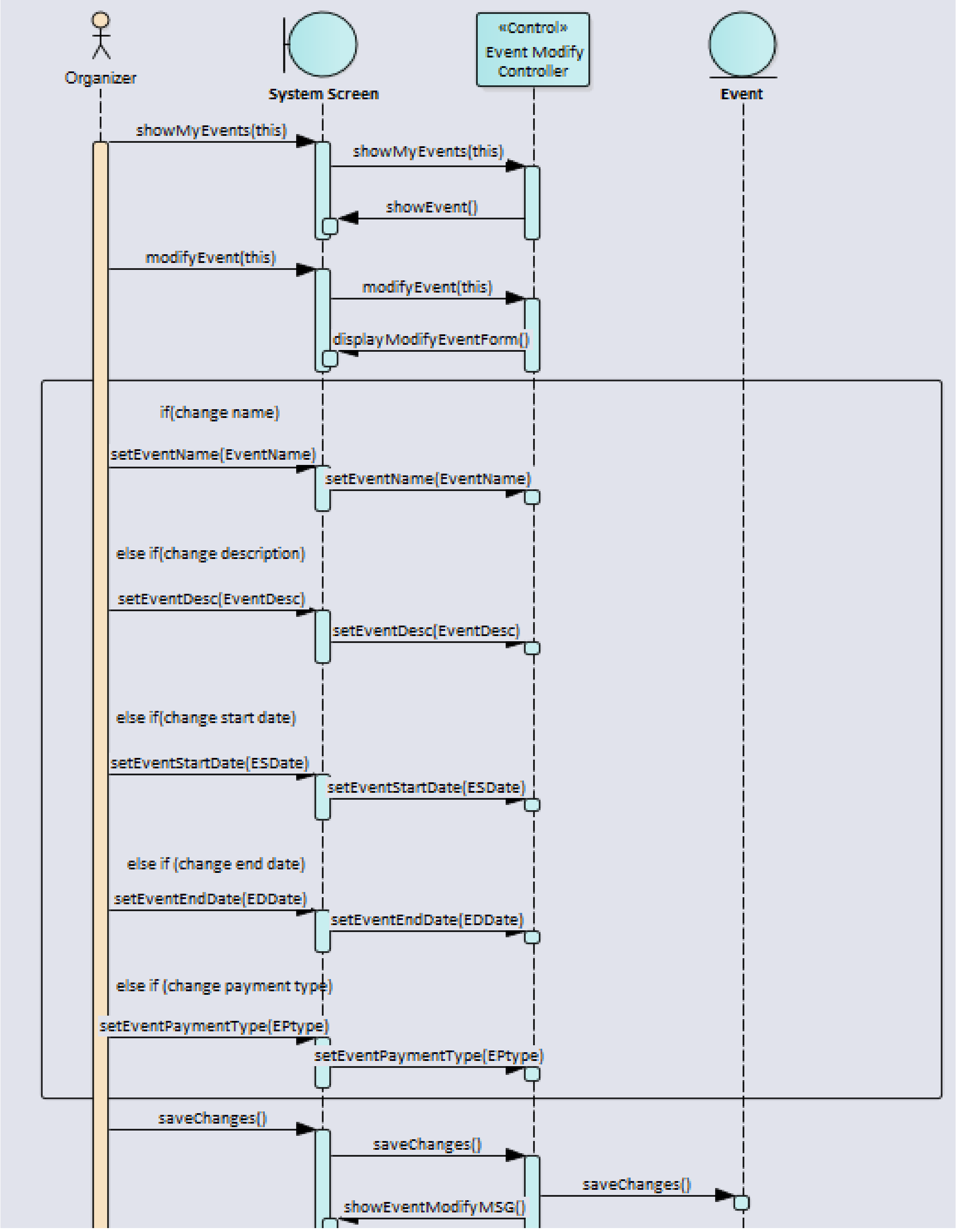


Figure 4.2 : sequence diagram for modify events use case

the above diagram is the sequence diagram for the modify events use case in which the User shall be able to modify the information of the event that they have created by clicking on the modify icon visible on the event created, The User then shall be taken to the form where he/she could modify the content of the input fields such as (Event Name , Event Description , etc…)

4.2.2.2 Sequence Diagram for Create Events Use Case

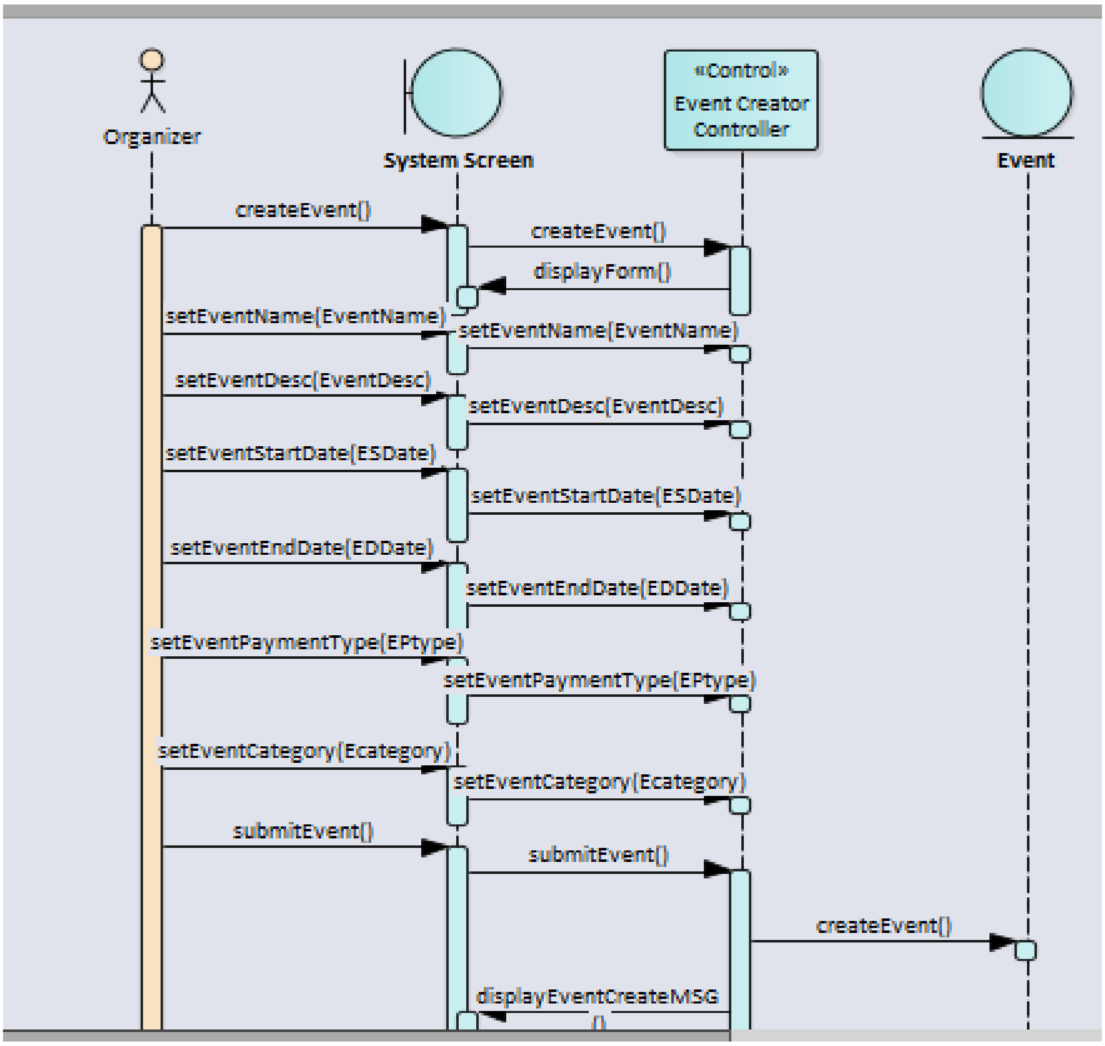


Figure 4.3 : sequence diagram for create events Use Case

the above diagram is the sequence diagram for the create events use case in which the User clicks on (create new event) and they are taken to the form in which they can input the necessary information about the event by inputting (event name , event description , event start date , event end date , event category , event payment type) to indicate the necessary information for the specified event and clicking on the submit event button in order to create the event and then the User will be shown an alert saying that states that the event they have created have been successfully created.

4.2.2.3 Sequence Diagram for Delete Events Use Case

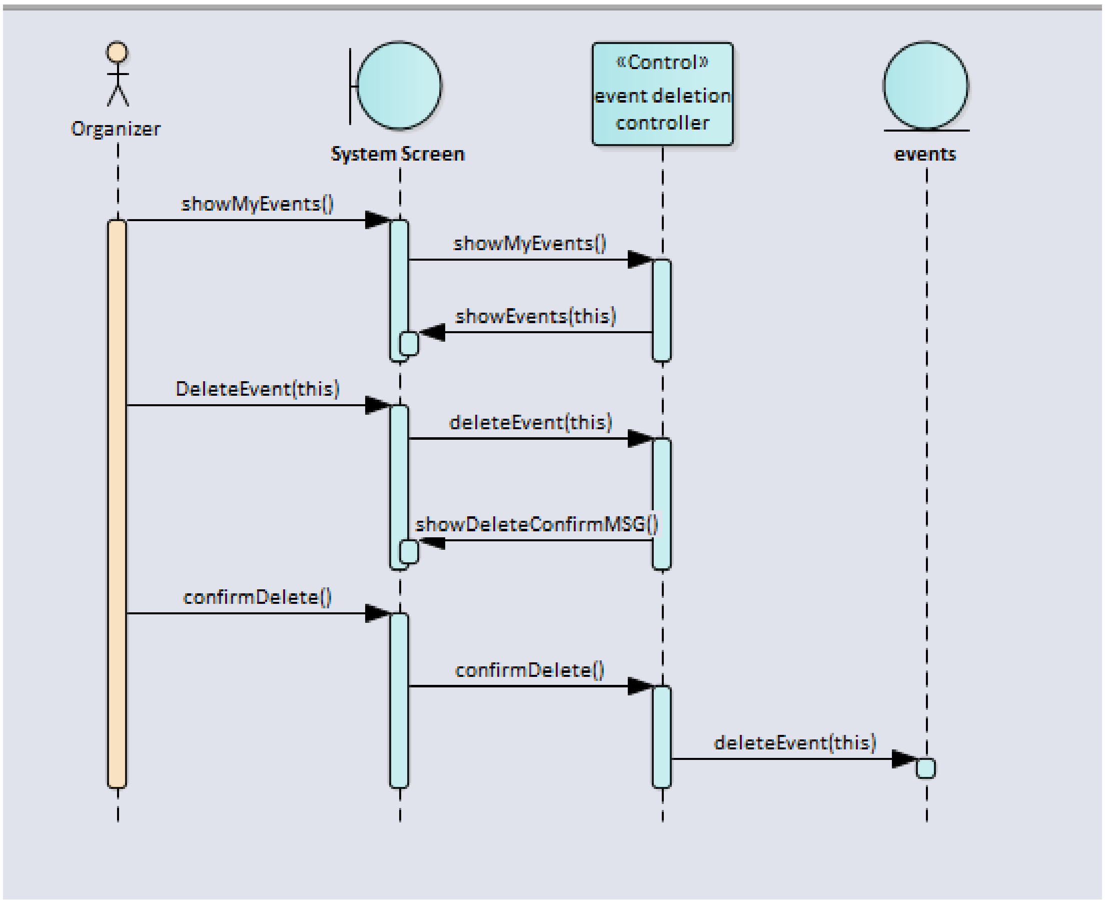


Figure 4.4 : sequence diagram for the delete events use case

in the above diagram we have the sequence diagram for the delete events use case in which the user first is shown all of the events that they have created and the user shall then be able to delete any one of those events by clicking on the delete icon on the event to delete the event but before the deletion he is shown a confirm message to confirm the deletion when the user confirms the delete , the event is deleted.

4.2.2.4 Sequence Diagram for Browse Events Use Case

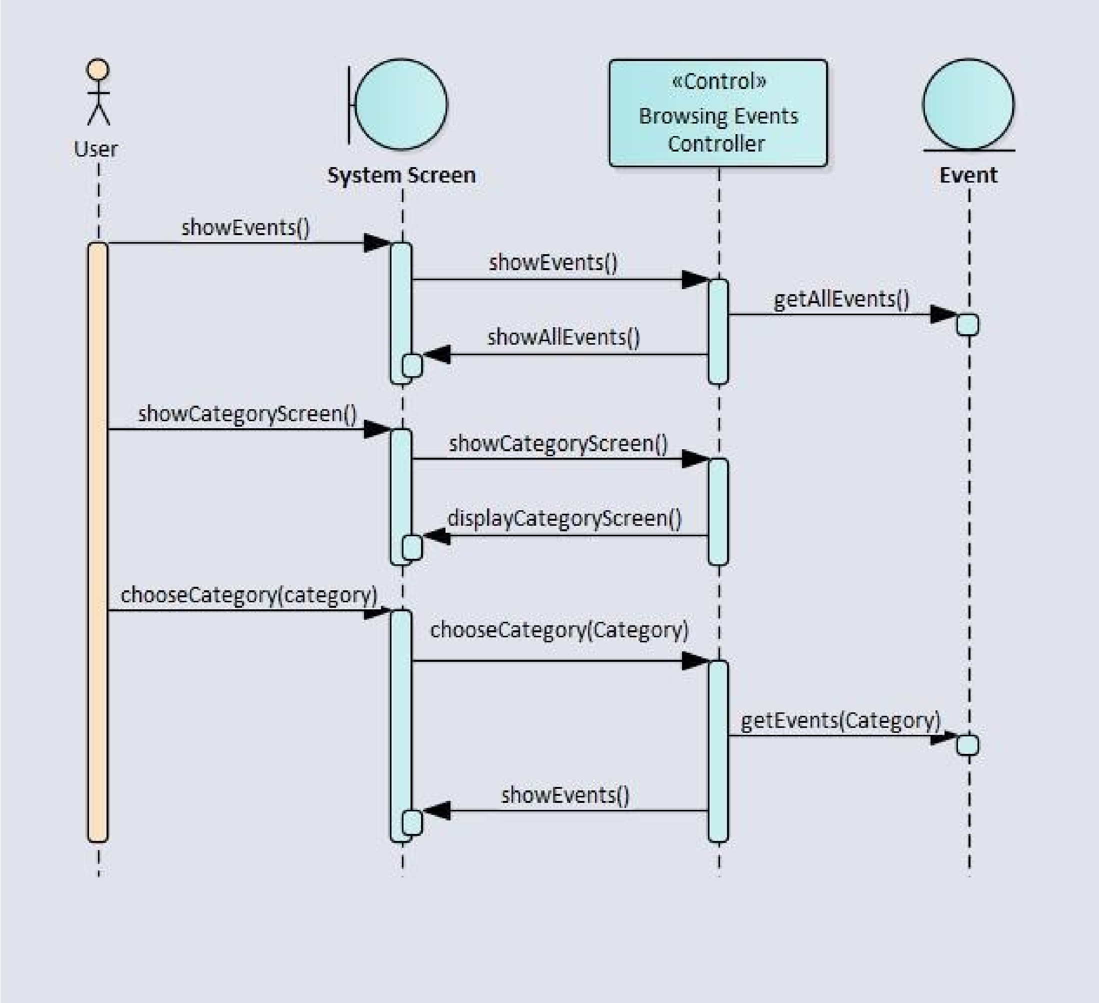


Figure 4.5 : sequence diagram for the browse events use case

in the above diagram we have a sequence diagram for the browse events use case in which the user is shown all of the events that are available in the system then as previously mentioned in the previous chapters the user can select the category of events to be shown hence the user picks a category and the events for that chosen category is shown and displayed to the user.

4.2.2.5 Sequence Diagram for Register for Event Use Case

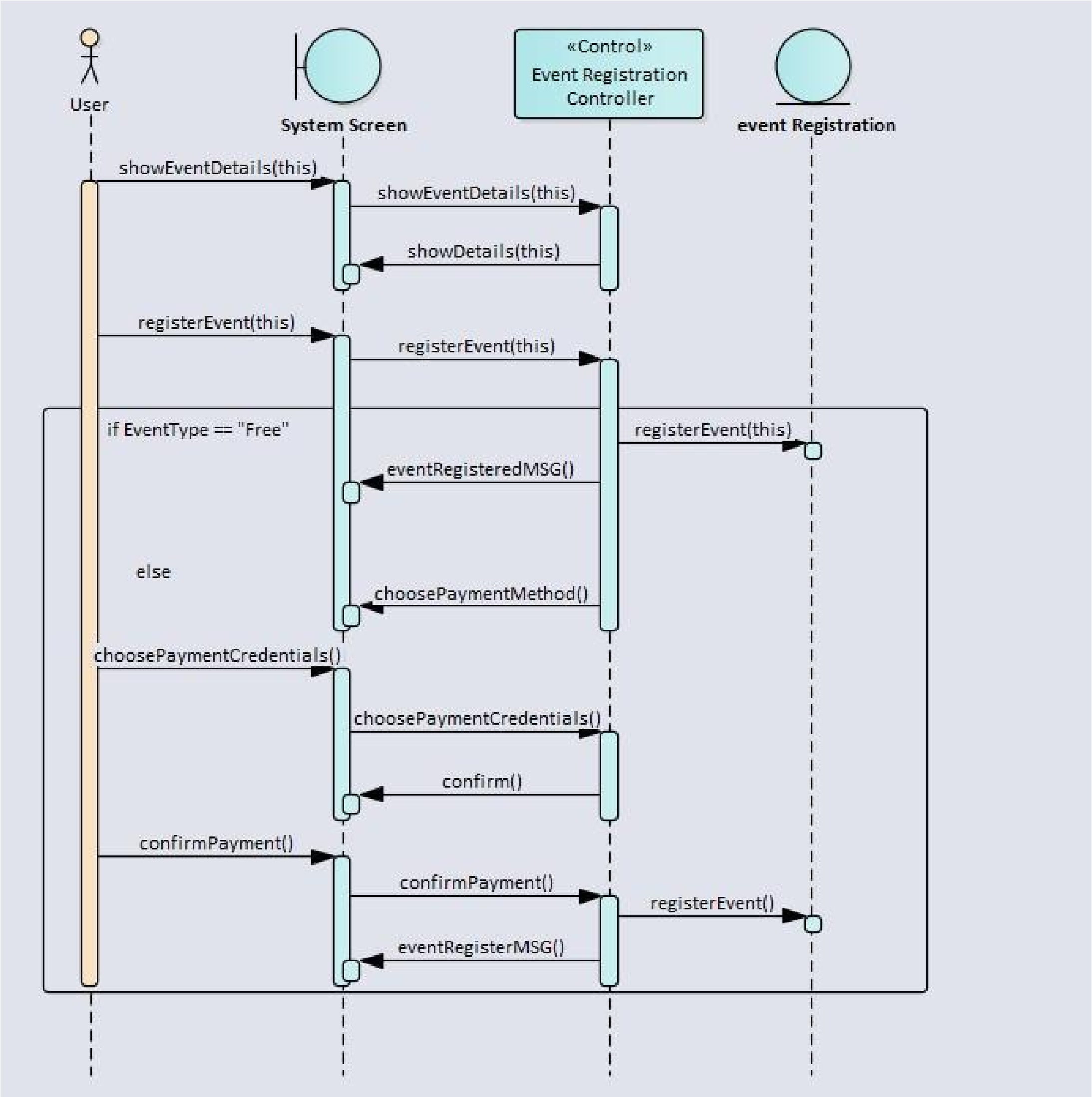


Figure 4.6 : sequence diagram for Register for Event Use case

in the above diagram we can see a sequence diagram for the register for event use case in which the User clicks on show event details and the then he is taken to the details of the event that contains all of the necessary information regarding the event and he is given the choice to register for this specific event the user then proceeds by clicking on the register for event button , then if the payment type of the event is (free) he is simply registered for the event and no further actions are needed , but if the payment type of the event is (paid) then he must choose the payment card or method he shall pay with and then The User picks one and then he shown a confirm message to confirm the payment , then the User confirms the payment.

4.2.2.6 Sequence Diagram for Payment Use Case

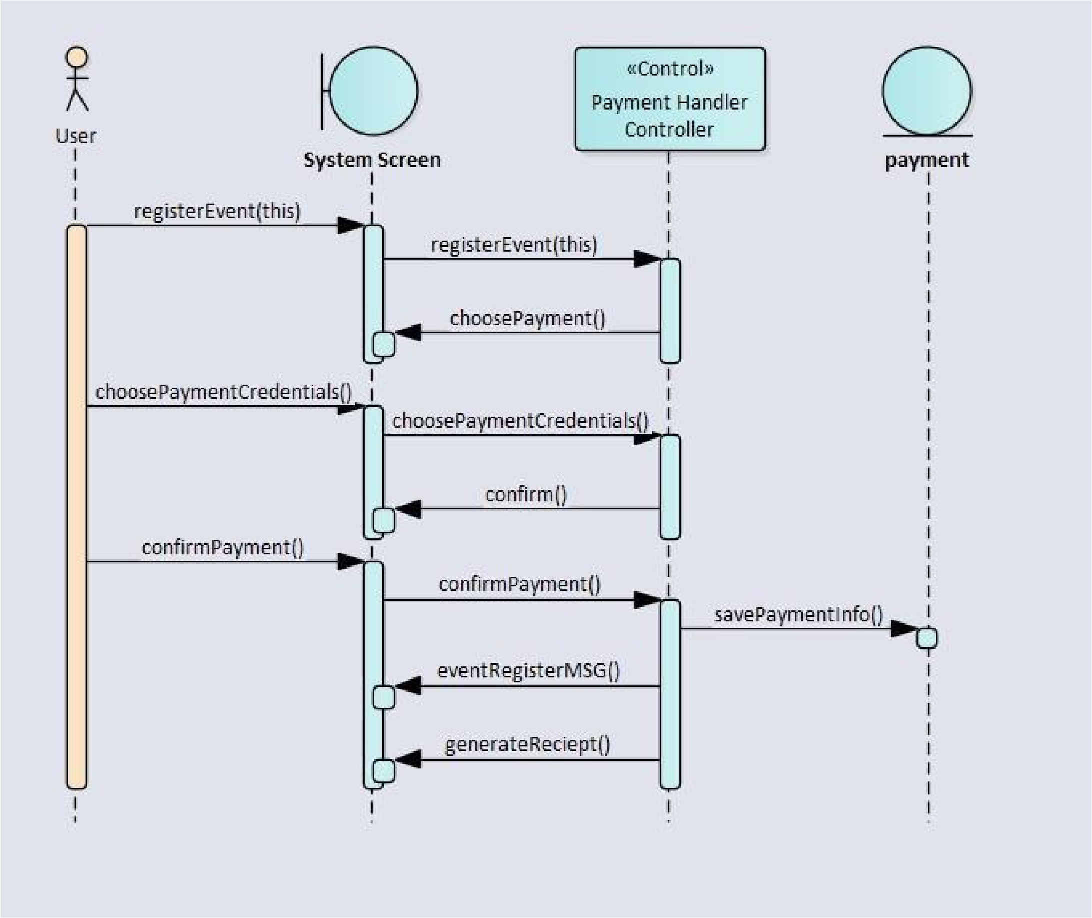


Figure 4.7 : sequence diagram for payment use case

in this diagram we are given the sequence diagram for the payment use case in which it takes place for (paid) events when registering for an event in which the payment method or payment card is chosen by the user and the payment is confirmed by the user this leads to registration of the event and the payment transaction shall be saved in the database as proof of payment and the event registered message shall be shown to the user along with the generation of a receipt.

4.2.2.7 Sequence Diagram for Generate Receipt Use Case

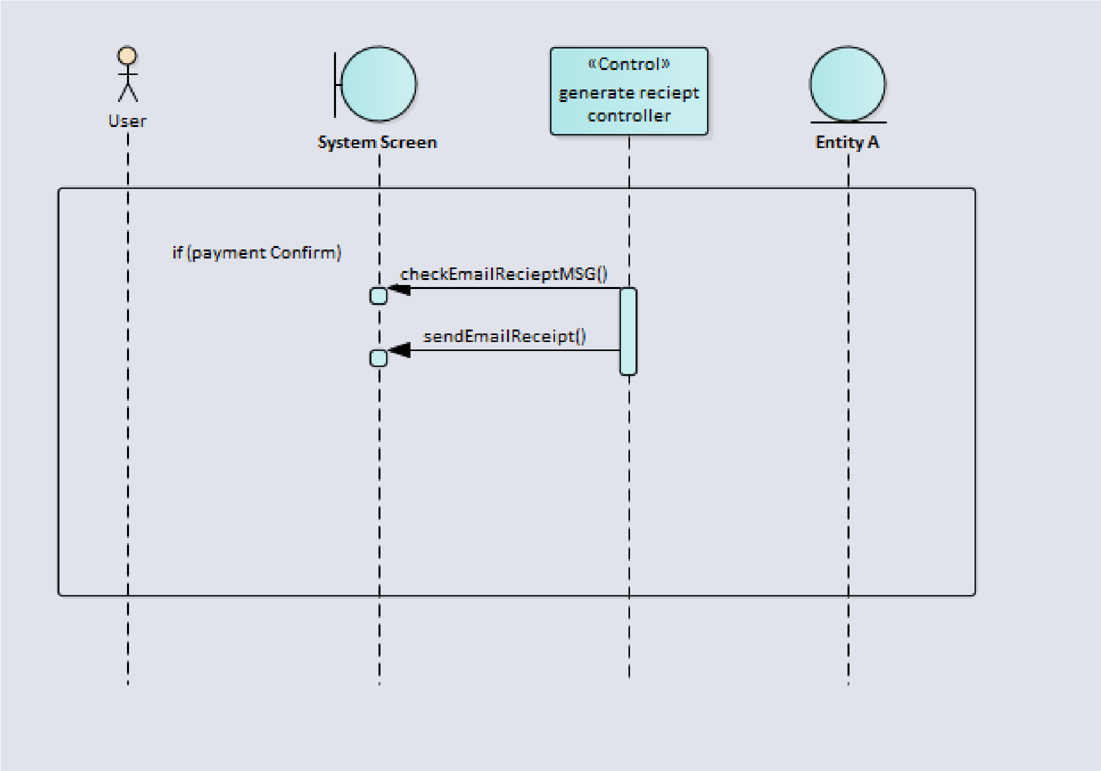


Figure 4.8 : sequence diagram for the generate receipt use case

In this diagram sequence diagram for generating receipts is shown in which once the payment is confirmed a message that tells the user to check their email for a receipt is shown to the user along with sending the receipt to the user's email.

4.2.2.8 Sequence Diagram for Give Feedback Use Case

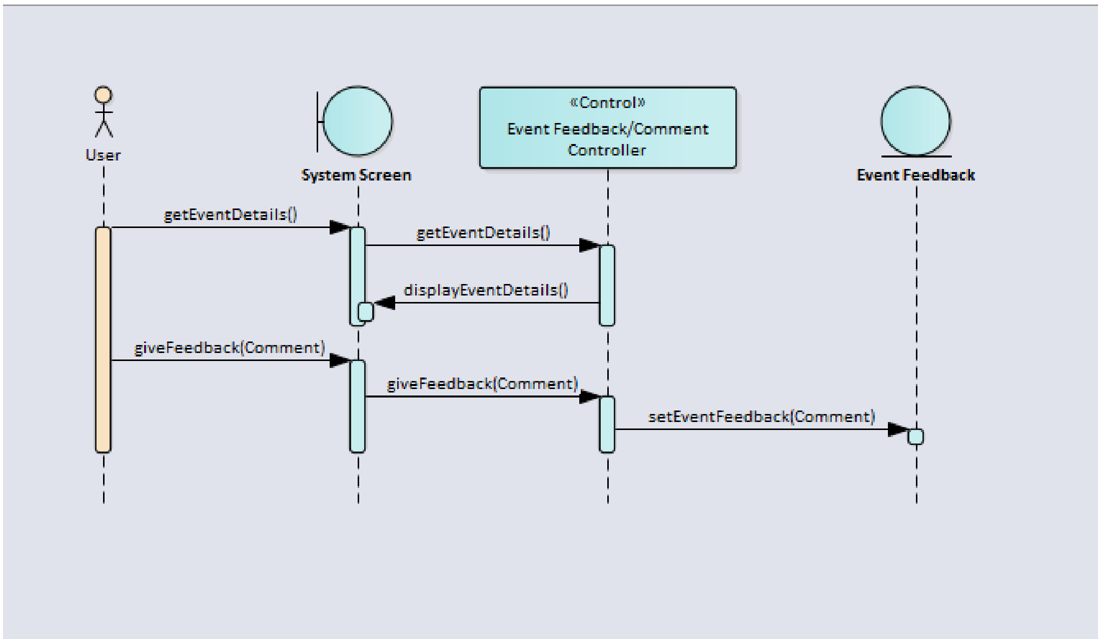


Figure 4.9 : sequence diagram for the Give Feedback use case

in this sequence diagram the user will click on the event to get the full details of the event and then gives feedback/comment on the event in the specified field and the comment or feedback shall be saved

4.2.2.9 Sequence Diagram for Give Rating Use Case

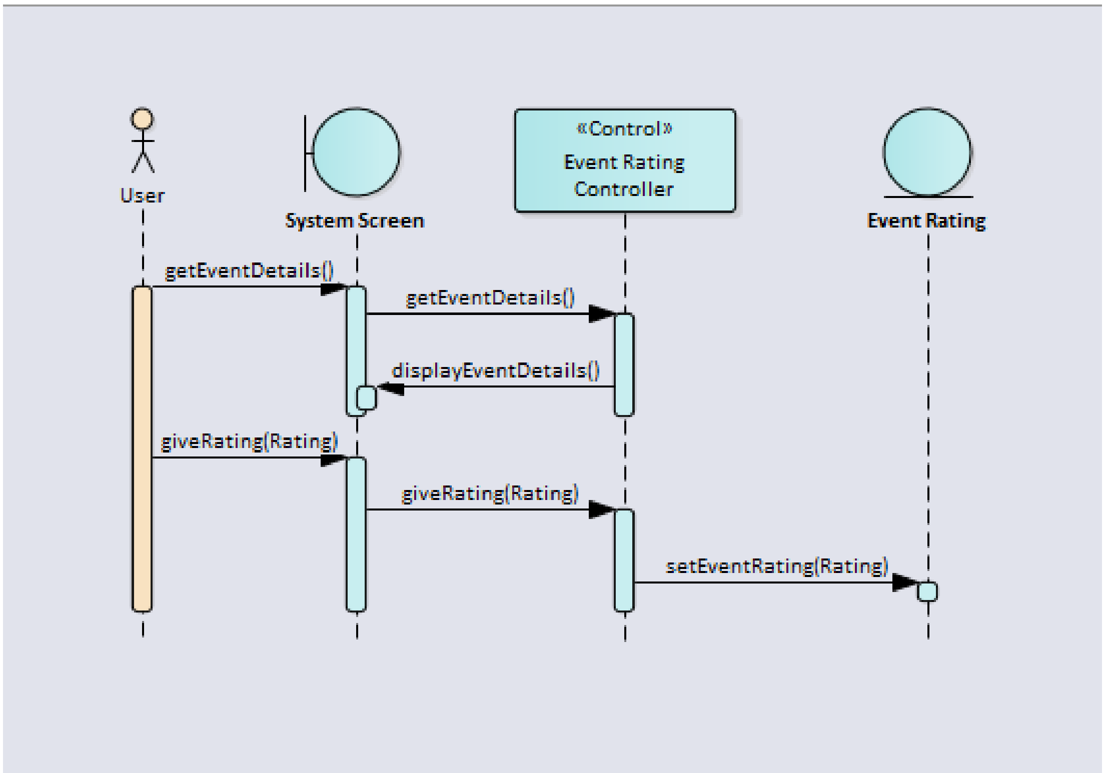


Figure 4.10 : sequence diagram for the Give Rating use case

in this sequence diagram the User clicks on the event to get the full details of the event and then chooses a rating from the specified field to rate the event and then the rating is saved.

4.2.3 Activity Diagrams of the System

4.2.3.1 Activity Diagram for the Modify Events Use case

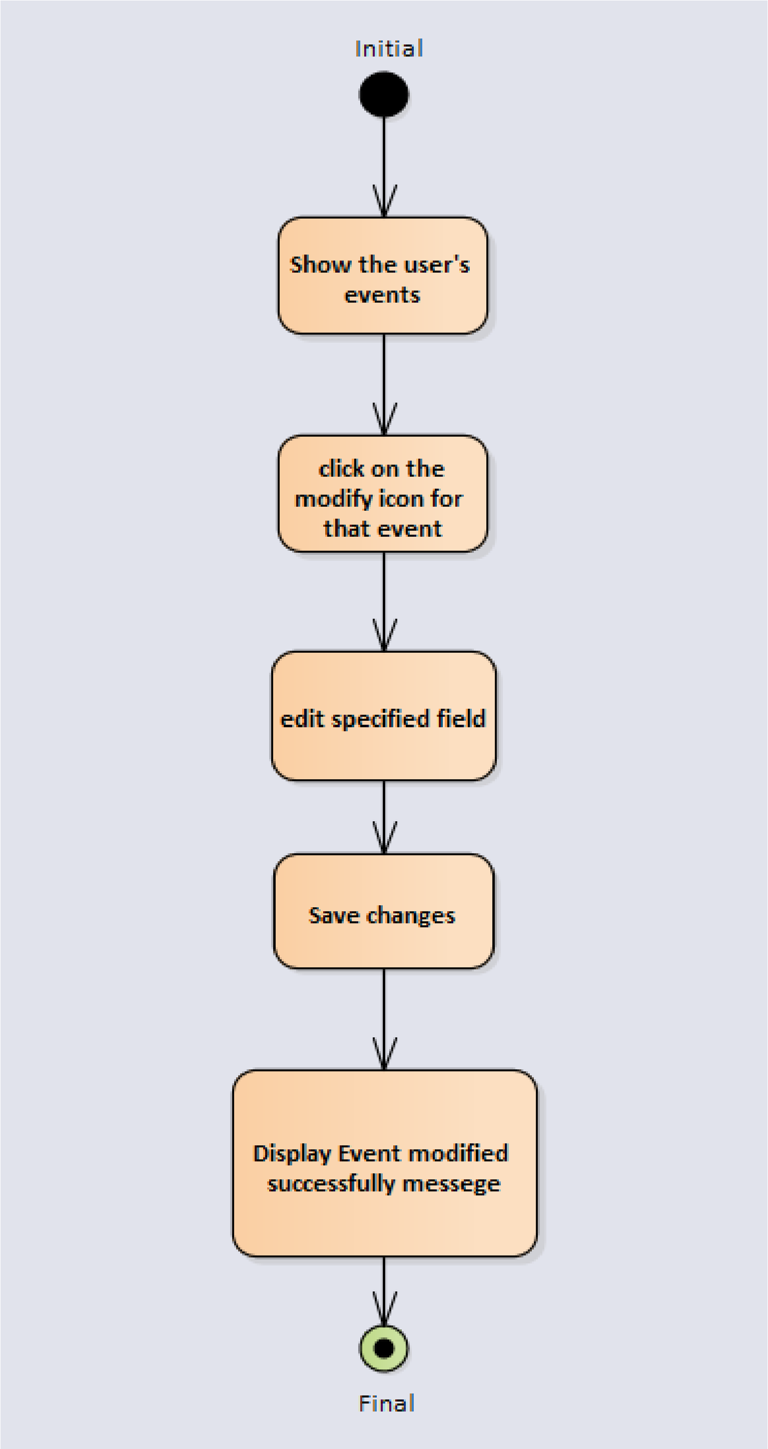

Initial
Show the user's events
click on the modify icon for that event
edit specified field
Save changes
Display Event modified successfully messege
Final

Figure 4.11 : activity diagram for the modify events use case

In this activity diagram for the modify events use case the user's events are shown and the modify icon for the event is clicked to be taken to the input form in which the data of the event can be edited by the user.

4.2.3.2 Activity Diagram for the Create Events use case

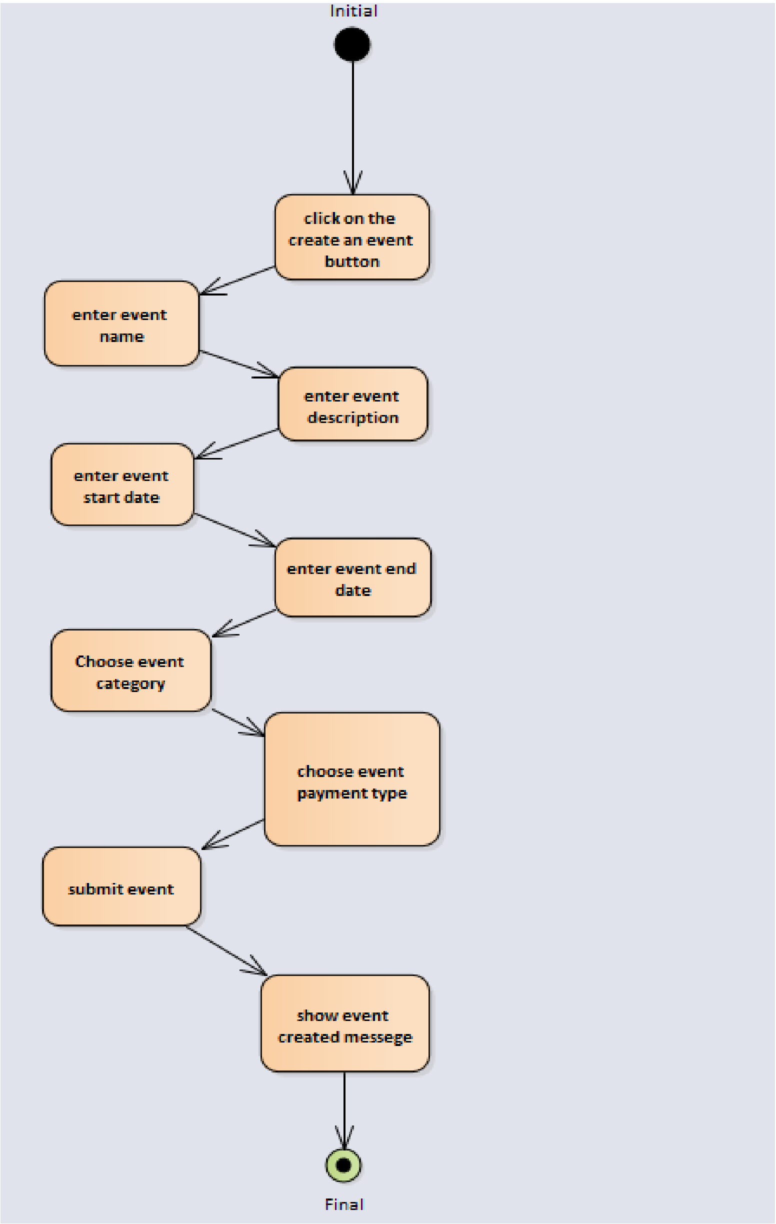


Figure 4.12 : activity diagram for the create events use case

4.2.3.3 Activity Diagram for the Delete Events use case

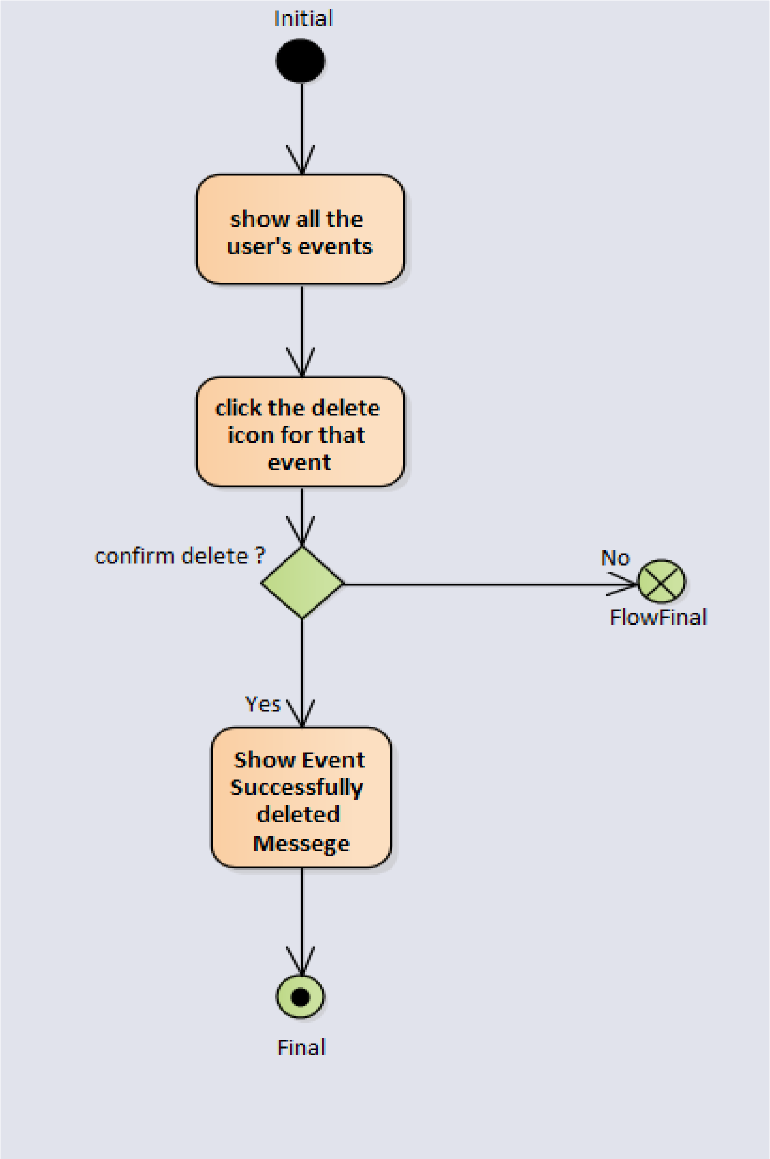

Initial
show all the user's events
click the delete icon for that event
confirm delete ?
No
FlowFinal
Yes
Show Event Successfully deleted Messege
Final

Figure 4.13 : activity diagram for the delete event use case

4.2.3.4 Activity Diagram for the Browse Events use case

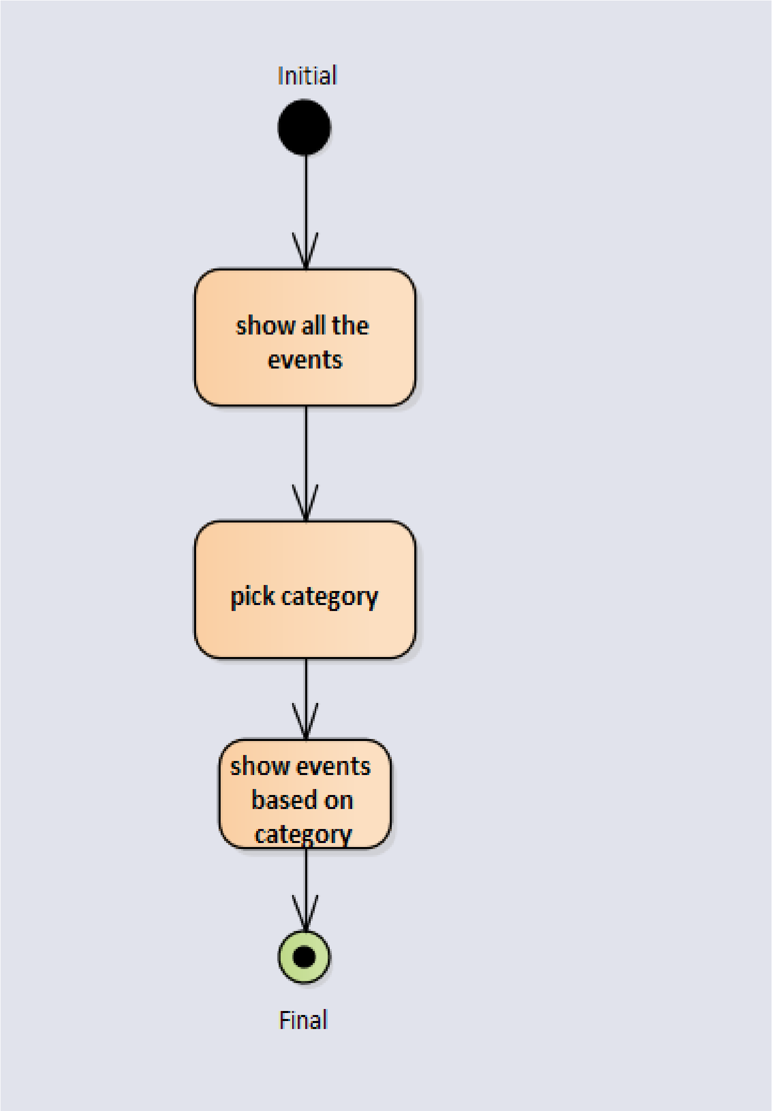
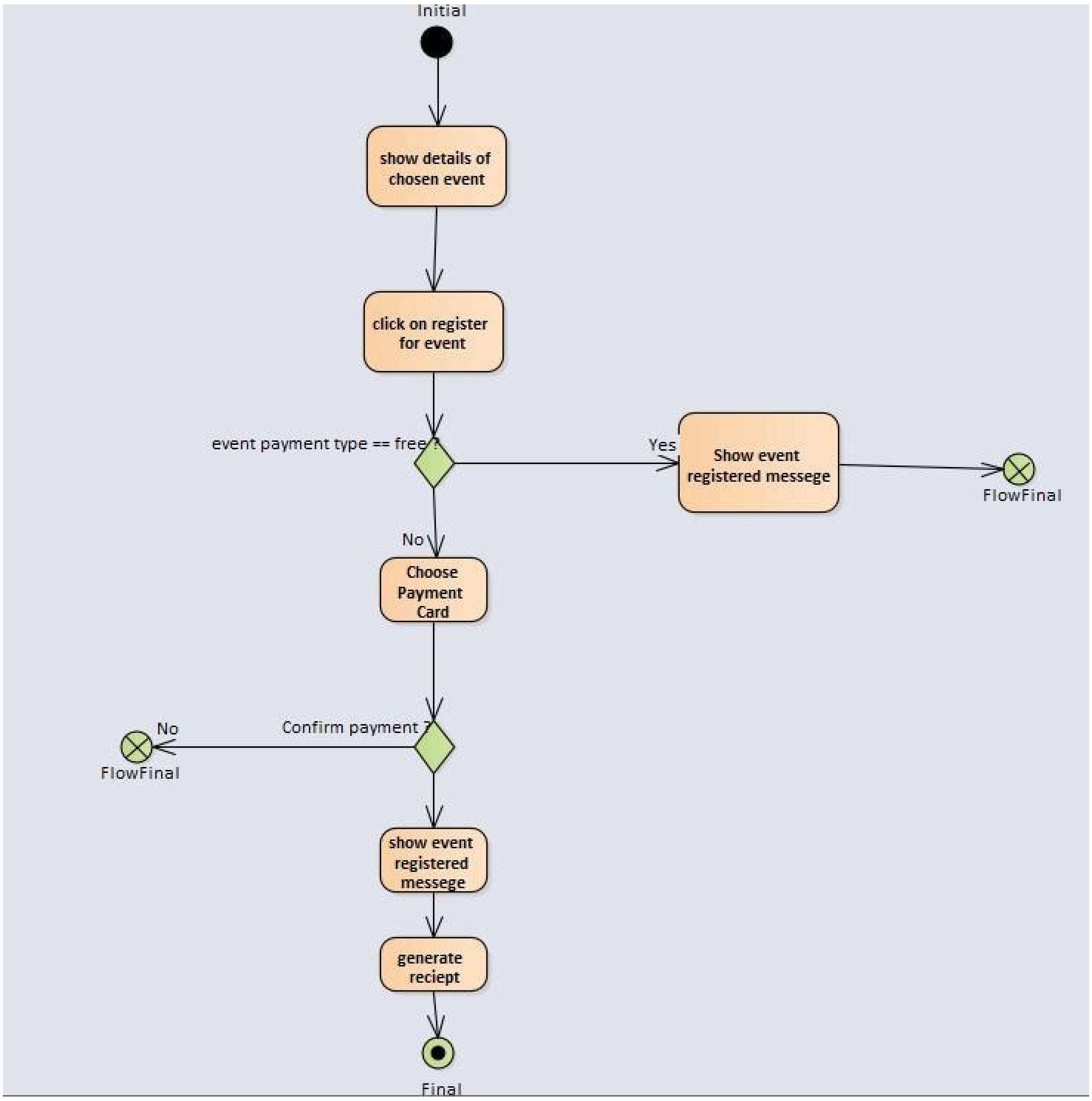


Figure 4.14 : activity diagram for the browse events use case

4.2.3.5 Activity Diagram for the Register for Event use case


Initial
show details of chosen event
click on register for event
event payment type == free ?
Yes
Show event registered messege
FlowFinal
No
Choose Payment Card
No
Confirm payment ?
FlowFinal
show event registered messege
generate reciept
Final

4.15 : activity diagram for the register for event use case

in this activity diagram for register for an event the full details of the event are shown to the user and the user clicks on the register for event button in which the user is registered is the payment type is free if not then the user shall be given the option to choose a payment card to pay once that is chosen he is shown a confirm message to confirm the payment once it’s confirmed the user is confirmed.

4.2.3.6 Activity Diagram for payment use case

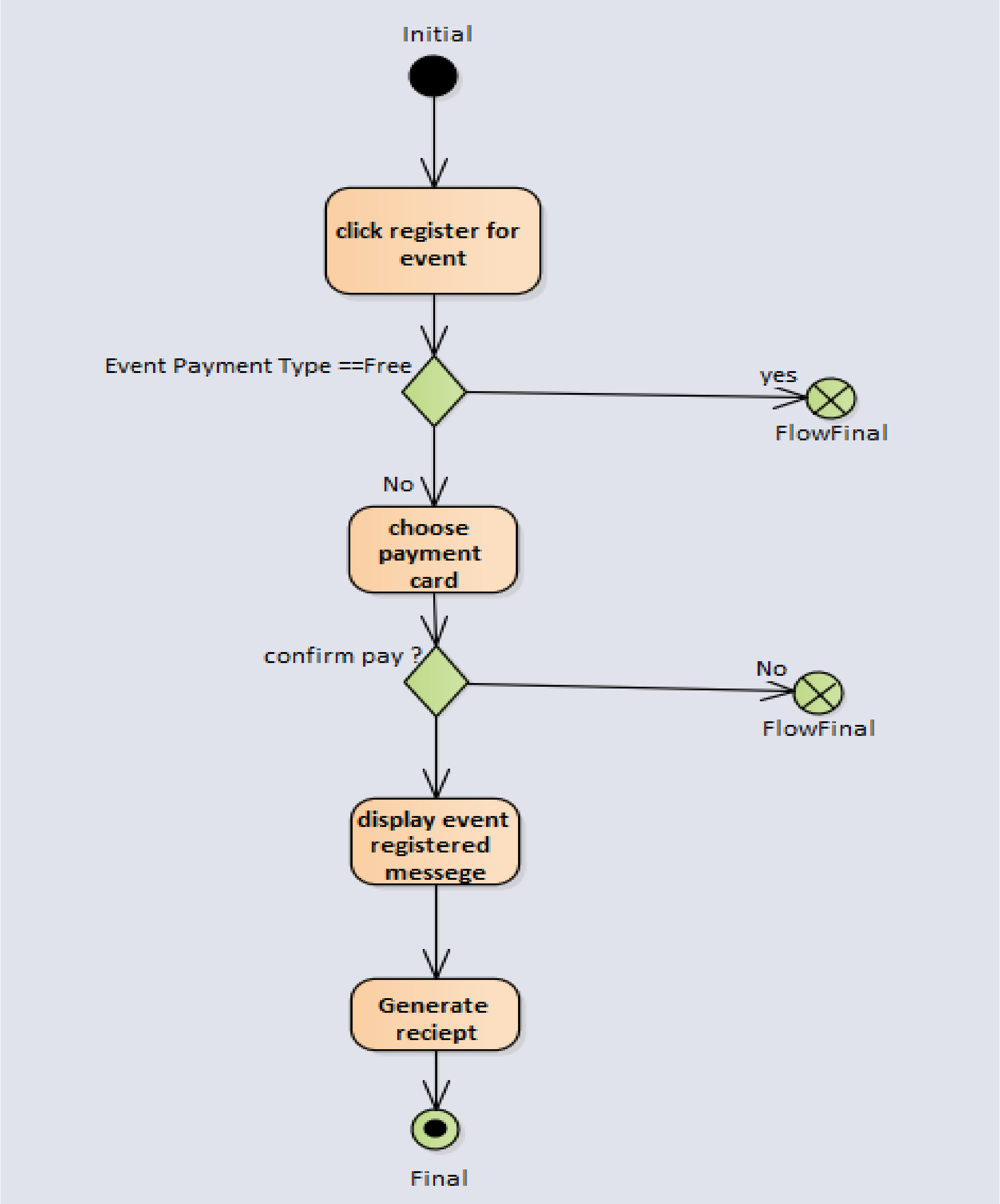


Figure 4.16 : activity diagram the payment use case

4.2.3.7 Activity Diagram for Generate Receipt use case

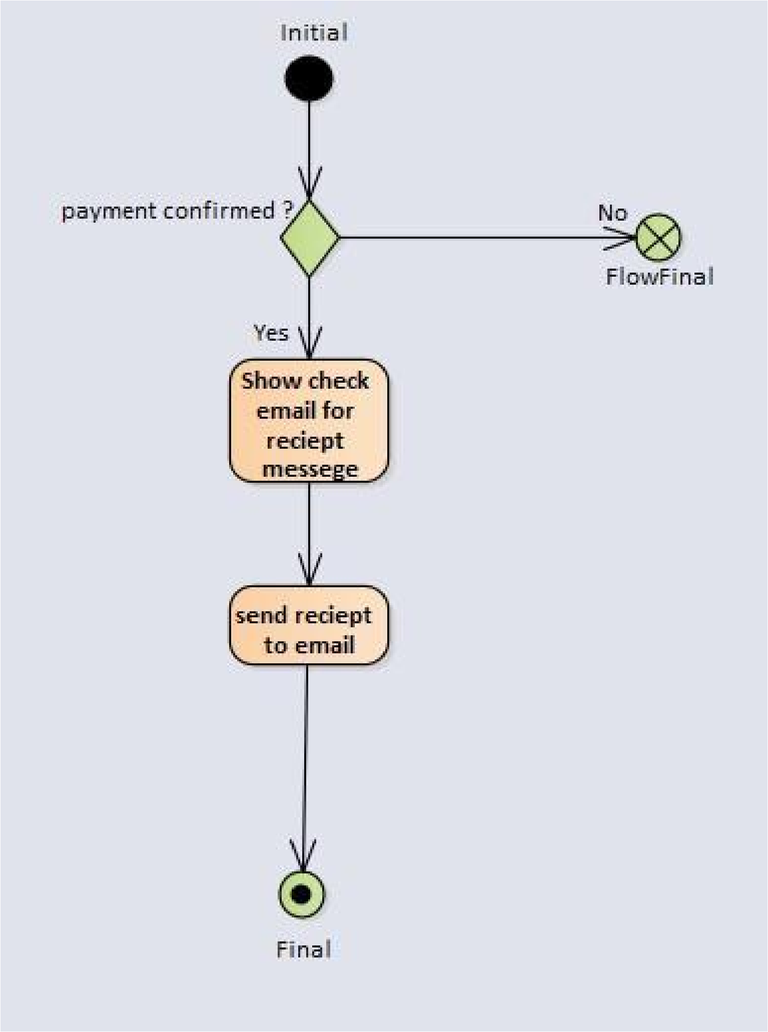
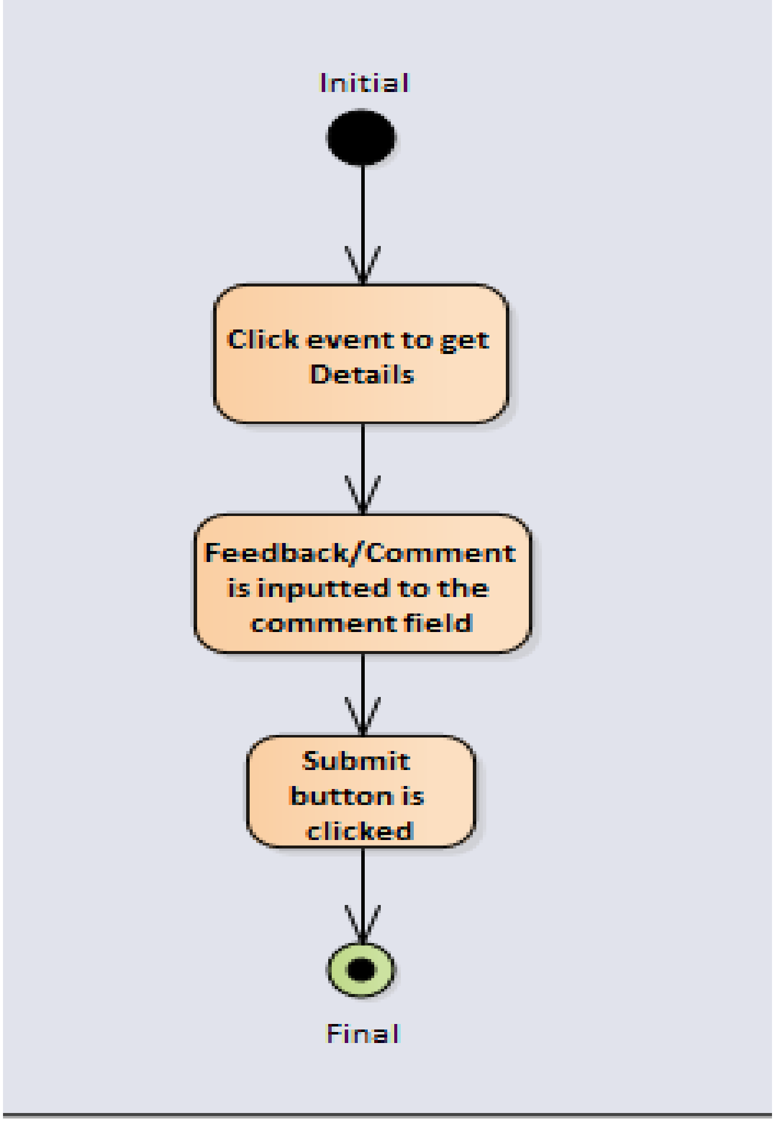


Figure 4.17 : activity diagram for generate receipt use case

4.2.3.8 Activity Diagram for Give Feedback use case



Initial
Click event to get Details
Feedback/Comment is inputted to the comment field
Submit button is clicked
Final

Figure 4.18 : activity diagram the Give Feedback use case

4.2.3.9 Activity Diagram for Give Rating use case

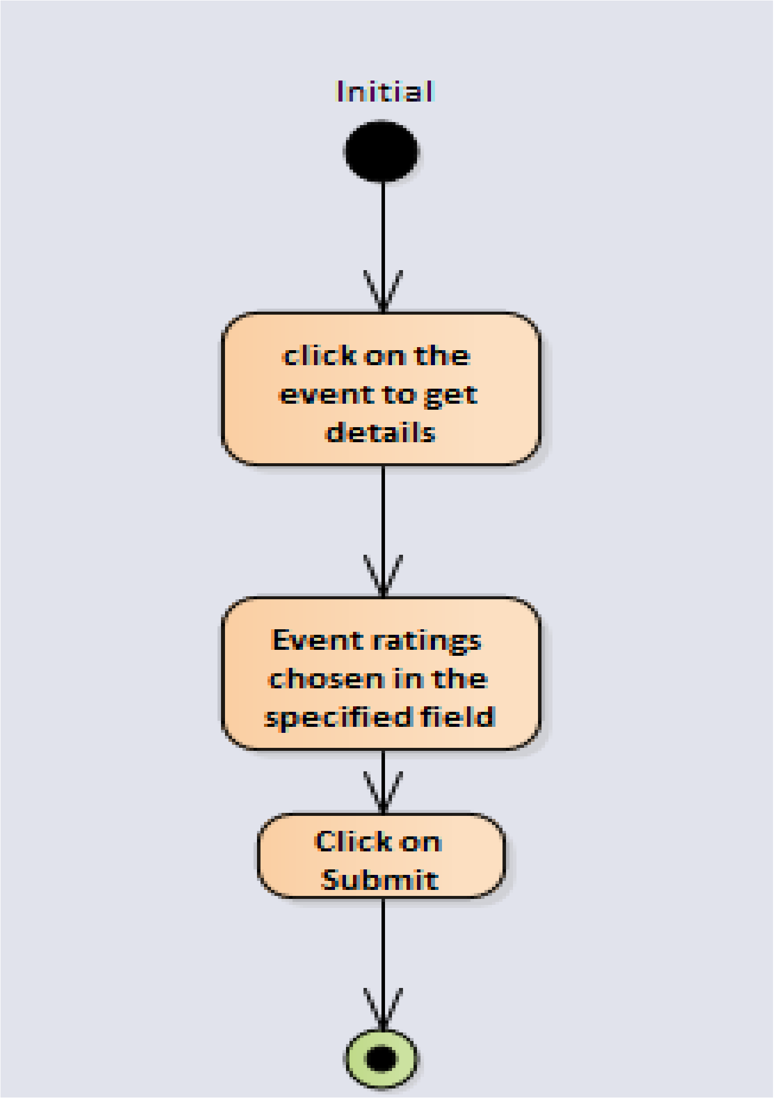

Initial
click on the event to get details
Event ratings chosen in the specified field
Click on Submit

Figure 4.19 : activity diagram the Give Rating use case

4.3 Project Design

For the project design the structural aspect of the system shall be viewed by using an OOP class diagram that labels the attributes and the operations of the classes in the proposed system along with their relationships.

4.3.1 Class Diagram of the System

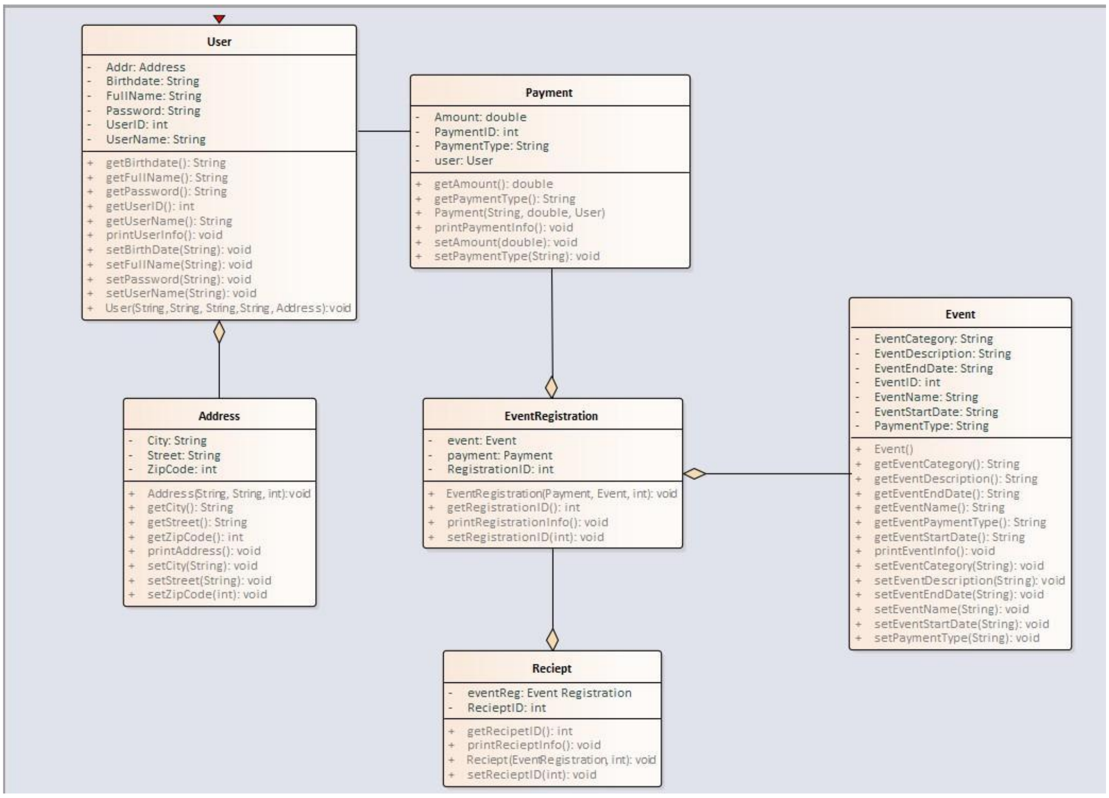

User
- Addr: Address
- Birthdate: String
- FullName: String
- Password: String
- UserID: int
- UserName: String
+ getBirthdate(): String
+ getFullName(): String
+ getPassword(): String
+ getUserID(): int
+ getUserName(): String
+ printUserInfo(): void
+ setBirthDate(String): void
+ setFullName(String): void
+ setPassword(String): void
+ setUserName(String): void
+ User(String,String, String,String, Address):void
Payment
- Amount: double
- PaymentID: int
- PaymentType: String
- user: User
+ getAmount(): double
+ getPaymentType(): String
+ Payment(String, double, User)
+ printPaymentInfo(): void
+ setAmount(double): void
+ setPaymentType(String): void
Event
- EventCategory: String
- EventDescription: String
- EventEndDate: String
- EventID: int
- EventName: String
- EventStartDate: String
- PaymentType: String
+ Event()
+ getEventCategory(): String
+ getEventDescription(): String
+ getEventEndDate(): String
+ getEventName(): String
+ getEventPaymentType(): String
+ getEventStartDate(): String
+ printEventInfo(): void
+ setEventCategory(String): void
+ setEventDescription(String): void
+ setEventEndDate(String): void
+ setEventName(String): void
+ setEventStartDate(String): void
+ setPaymentType(String): void
Address
- City: String
- Street: String
- ZipCode: int
+ Address(String, String, int):void
+ getCity(): String
+ getStreet(): String
+ getZipCode(): int
+ printAddress(): void
+ setCity(String): void
+ setStreet(String): void
+ setZipCode(int): void
EventRegistration
- event: Event
- payment: Payment
- RegistrationID: int
+ EventRegistration(Payment, Event, int): void
+ getRegistrationID(): int
+ printRegistrationInfo(): void
+ setRegistrationID(int): void
Reciept
- eventReg: Event Registration
- RecieptID: int
+ getRecipetID(): int
+ printRecieptInfo(): void
+ Reciept(EventRegistration, int): void
+ setRecieptID(int): void

Figure 4.20 : class diagram of the system

4.4 Database Design

In the database design the database entities are labelled along with their primary keys and foreign keys in a normalized table by using the entity relationship diagram to showcase the entities along with their attributes and their relationships.

4.4.1 Entity Relationship diagram for the system

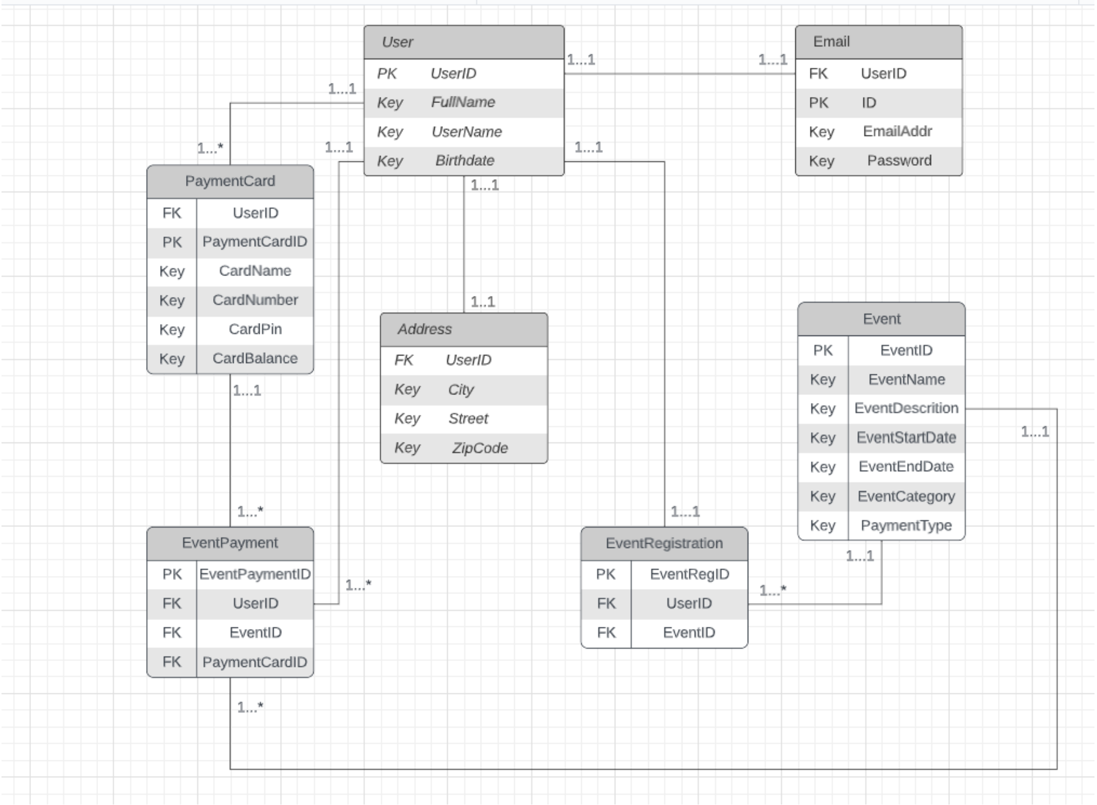


Figure 4.21 : entity relationship diagram for the system

4.5 Interface Design

In this part the structure of the (Design and Implementation of Online Event Management System) is shown and indicates how it will be structured in the future.

Figure 4.22 is the login page of the system in which the User logs in to the system by inserting username and password.

Figure 4.23 is the main screen of the system after the login by the user in which all the events of the system are shown in each category in which the user can browse; they can click on a specific category on this screen or the category tab can be used to view all the categories of the system.

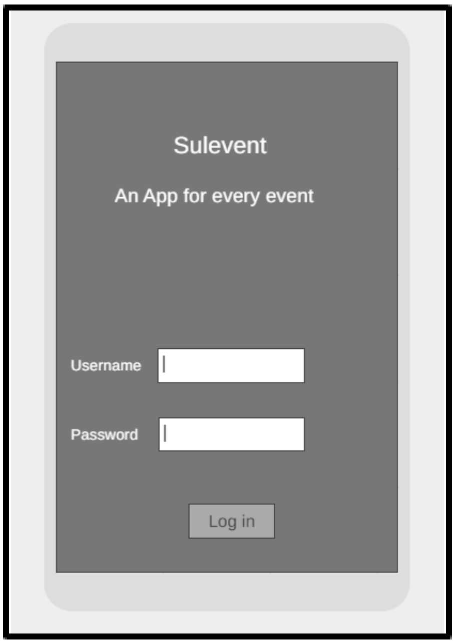


Figure 4.22 : Login Page

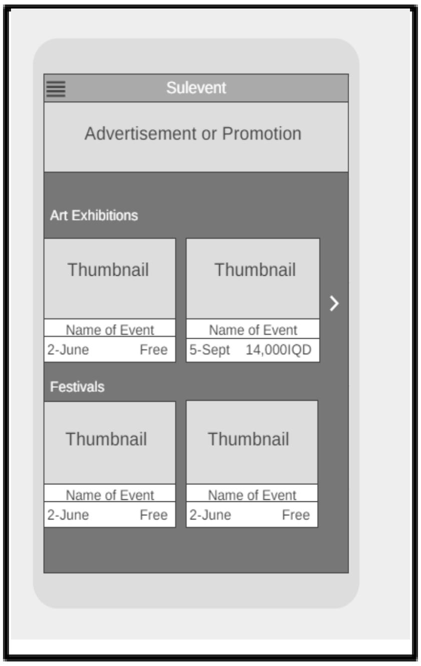


Figure 4.23 : Main Screen

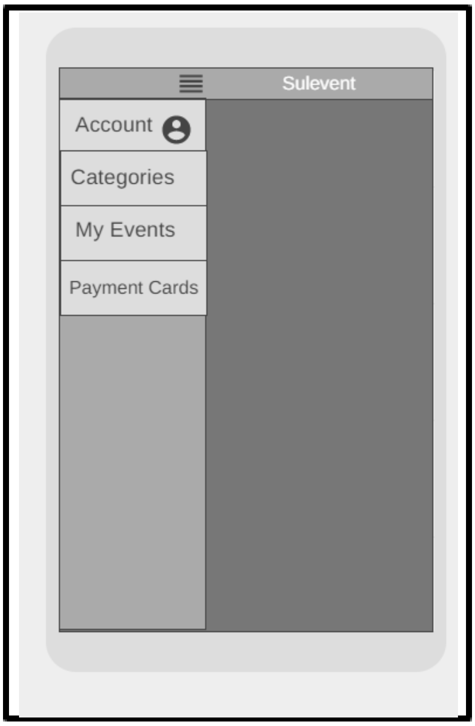


Figure 4.24 : Slider Menu

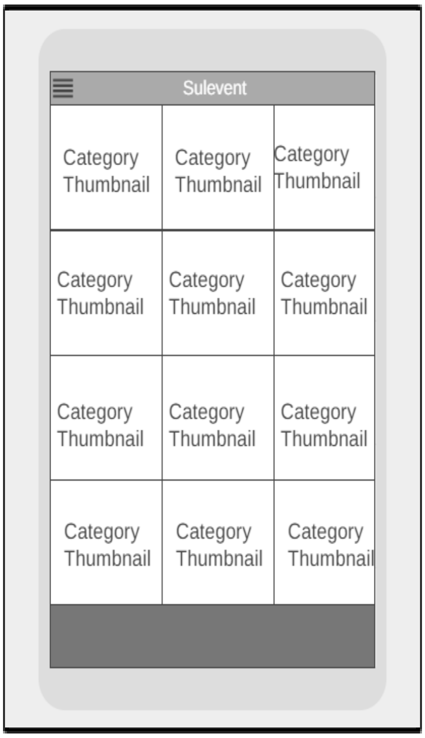
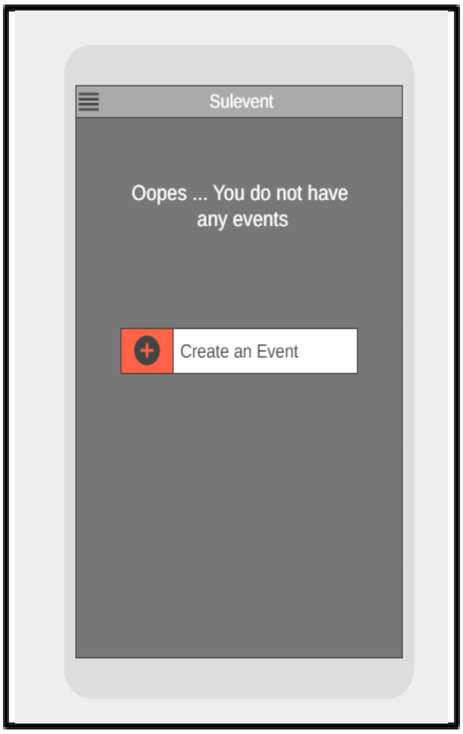


Figure 4.25 : Categories

Figure 4.24 is the slider menu that contains the options in the system such as viewing the account of the user or viewing the categories by clicking on the category tab or show the events of the user by clicking on the my events tab , the user can also choose the payment cards and insert their information by clicking on the payment cards tab.

Figure 4.25 shows The user after clicking on the categories in the slider menu they are shown these tiles that contain a thumbnail of each category and once clicked it shows the events related to that certain category.



Oopes ... You do not have any events

Figure 4.26 : Org. Home

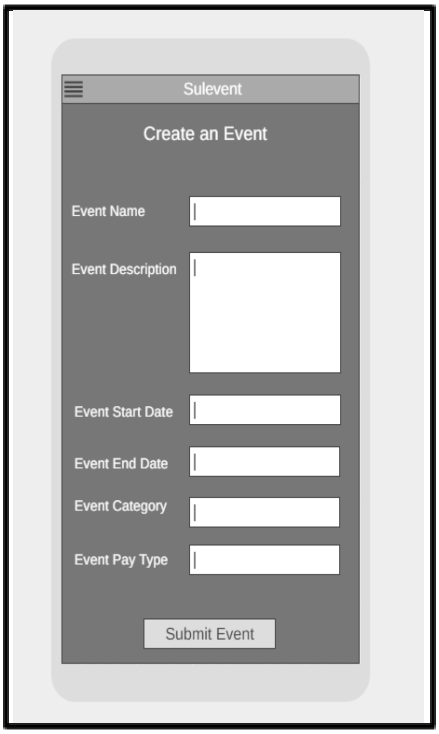


Figure 4.27 : Create Event

Figure 4.26 shows menu option is shown after the my events tab has been clicked by the user of the system in which the events that the user has created are shown here but in this case there are no events for that user hence the message “Oops…You do not have any events” is shown along with the choice of creating an event in which the user is taken to the input form for the creation of the event once they click on the create an event button.

Figure 4.27 is the input form for creating an event once the create an event button is clicked by the user it allows the user to input the name of the event , the description for the event , the event's start date , the event's end date , the event's category and the event pay type whether free or paid event. Then the submit event button is clicked in order to create the event.

In Figure 4.28 screen is shown once the user clicks on the my events tab in the slider menu in which the events of the user are shown and they are given the ability to either delete the event or to modify the existing information for that specific event.

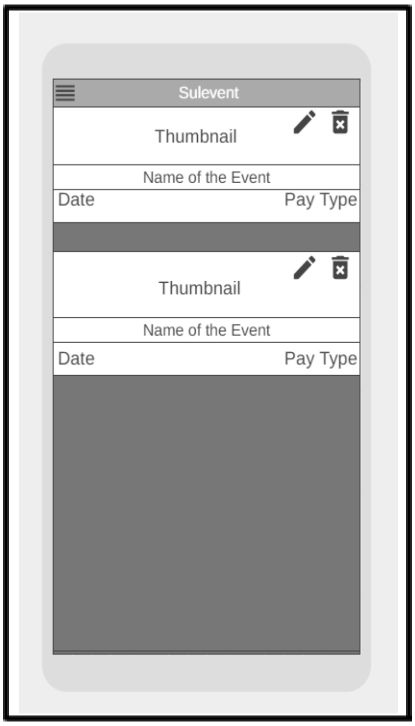


Figure 4.28 : my events

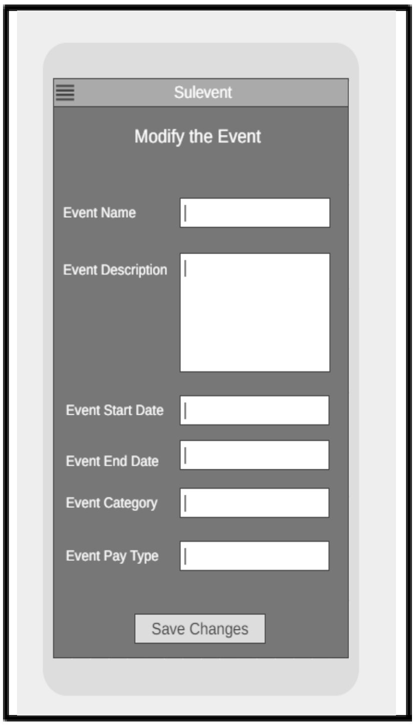


Figure 4.29 : modify events

Figure 4.29 screen is shown when the user clicks on the modification button in the my events tab to edit the information of the event that has been created, the user can then modify the event name , event description , the event start date , the event end date and the event category and the event pay type whether free or paid.

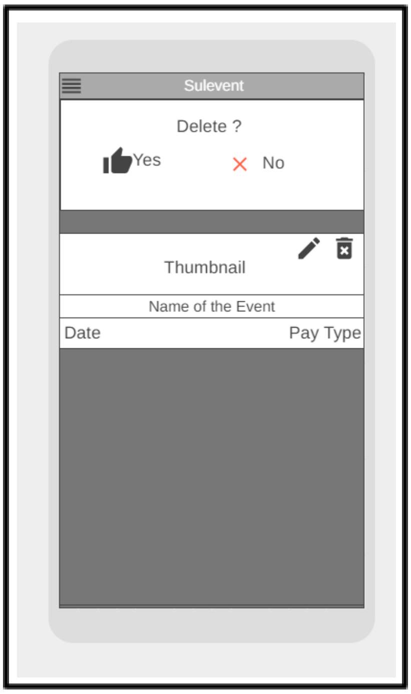


Figure 4.30 : confirm delete event

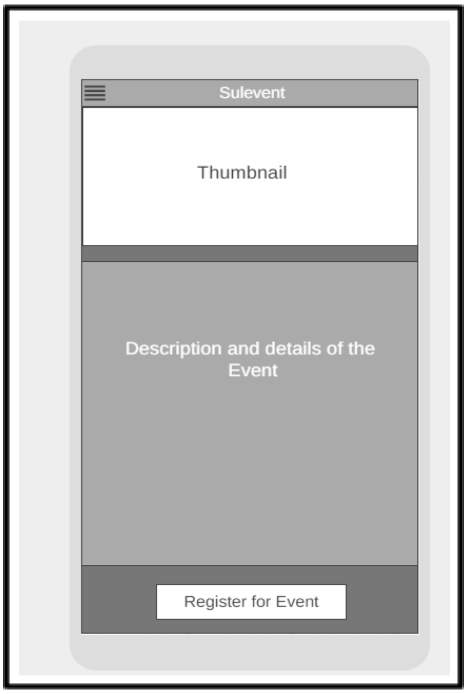
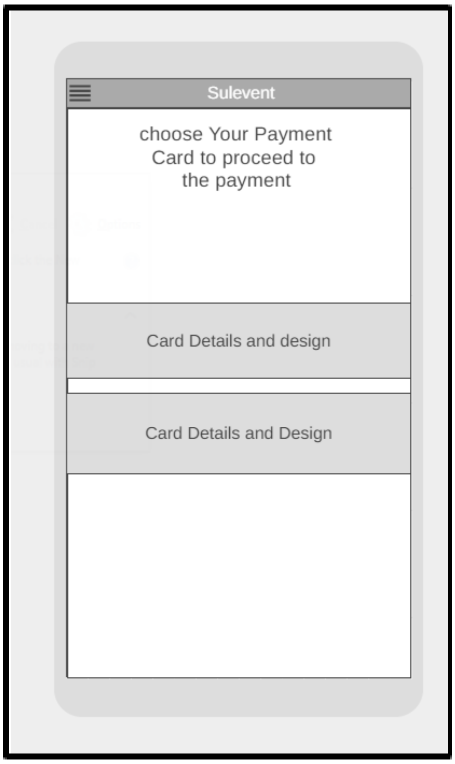


Figure 4.31 : register for event

Figure 4.30 message is shown when the delete icon is clicked and two choices are given to the user to either say no to refuse the deletion of the event or to say yes to confirm the deletion of the event.

Figure 4.31 screen is shown when the event is clicked to show the full details of the event as well as giving the user the ability to register for the event by clicking on the register for event button in which the user will be registered if the pay type of the event is free , however if not free then the user is given the choice to choose their payment card.



Sulevent

choose Your Payment Card to proceed to the payment

Figure 4.32 : payment cards to choose from

4.6 Chapter Summary

in this chapter the diagrams and the modelling of the system has been put forth in which the use case diagram covers the overall functionality of the system and the sequence diagram and activity diagrams are used to show the behavioural aspect of the system by labelling the actions of the system sequentially , the class diagram labels the structural aspect of the system by showing the classes and their attributes and the operations performed on those classes and the relationship between them and the entity relationship diagram is used to show the database entities and the relationship between them. Finally the User Interface's structure is shown to indicate how the system shall be structured in terms of the user interface.

Chapter 5

Implementation and Testing

5.1 Introduction

Software testing is a critical stage of the software development process that involves assessing a software system or application to find problems, faults, and possible areas for improvement. Its objectives are to ensure the software's quality, and operation as well as to confirm that it satisfies the required standards and user expectations. Software testing is the practice of finding and fixing problems with software before it is made available to end users when it is deployed and published to be used by the vast majority. It includes a variety of approaches, methodologies, and tools (Beizer, 1990).

Software testing is described by Beizer (1990) as "the process of executing a program or system with the intention of finding errors." It includes running test cases or scenarios, comparing the current output to the output expected, and examining inconsistent outputs to find flaws or potential problems. The objective is to ensure that the program serves the intended function, performs correctly under any situation given to it and offers a positive user experience in addition to finding and fixing the bugs.

Throughout the whole software development lifecycle, software testing performs a variety of functions. It aids in early error detection and correction by developers, lowering the costs By boosting trust in the system's performance and behavior, testing also improves software reliability. It helps to ensure that the program conforms to the rules and regulations as well as industry standards and functional requirements (Myers et al, 2011).

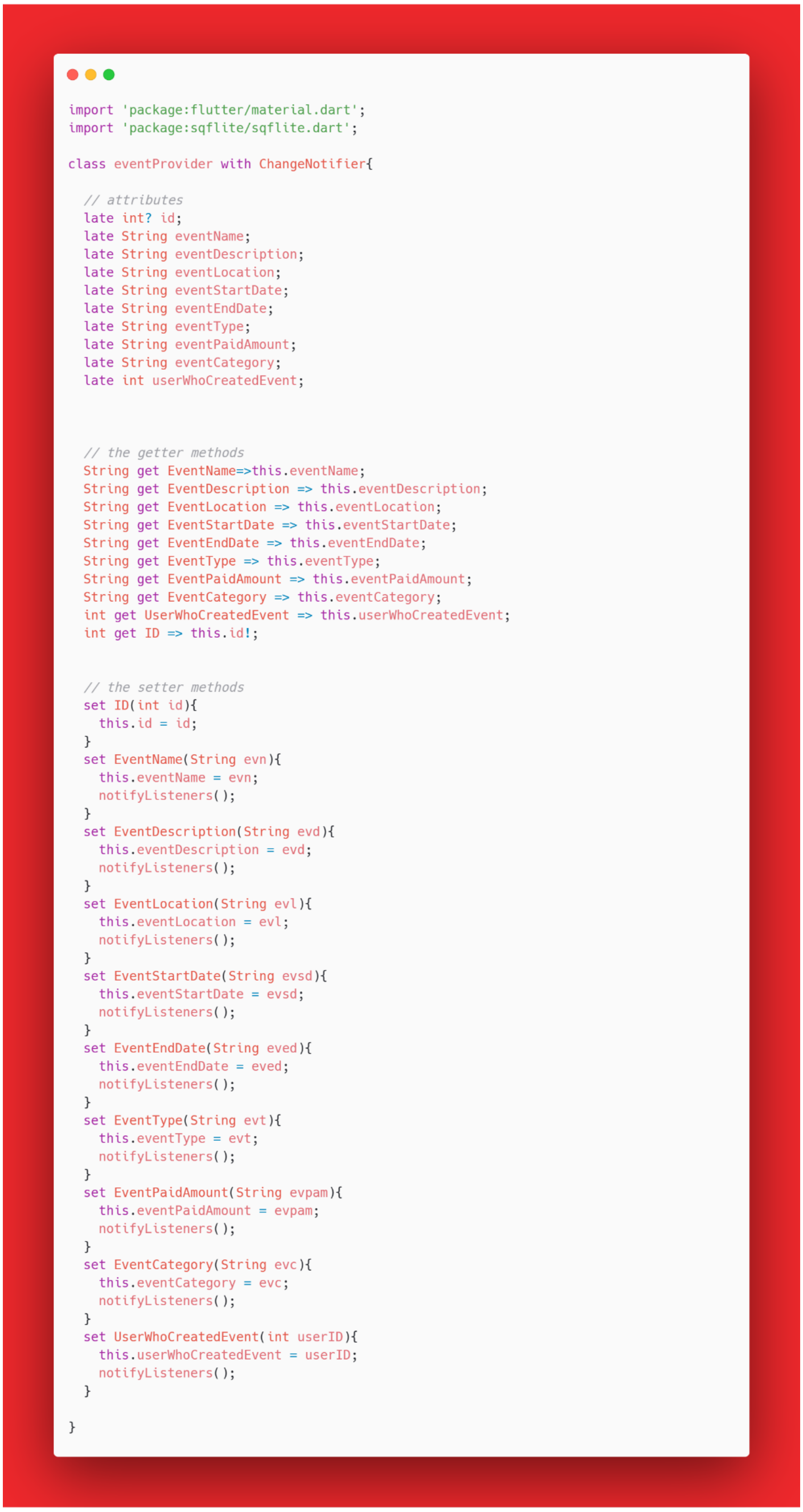
In software testing, there are a number of important ideas and methods. To ensure thorough coverage of the system during testing, objectives must be clearly defined before test techniques and test cases are designed. The process of running test cases, documenting outcomes. To automate repetitive processes, boost productivity, and speed up the testing process, experts use specialized technologies. Aiming for maximal coverage to reduce risks, test coverage evaluates how thoroughly the software has been tested (Pressman, 2014).

The testing process comprises different testing techniques, including non-functional testing and functional testing, which assess the system's performance, security, usability, and other quality features. There are also specific methods, such as testing, integration testing, system testing, and acceptance testing, each targeting different aspects of the software (Pressman, 2014).

Implementing software source code is the process of turning the requirements and logic of a software system into a functional and executable program. To convert source code into an executable format that can be run on a computer or other device, it must be written, compiled, and linked in a software called the compiler. Implementation of software is essential for converting abstract ideas into usable software applications and is a crucial stage in the software development life cycle (Sommerville, 2016).

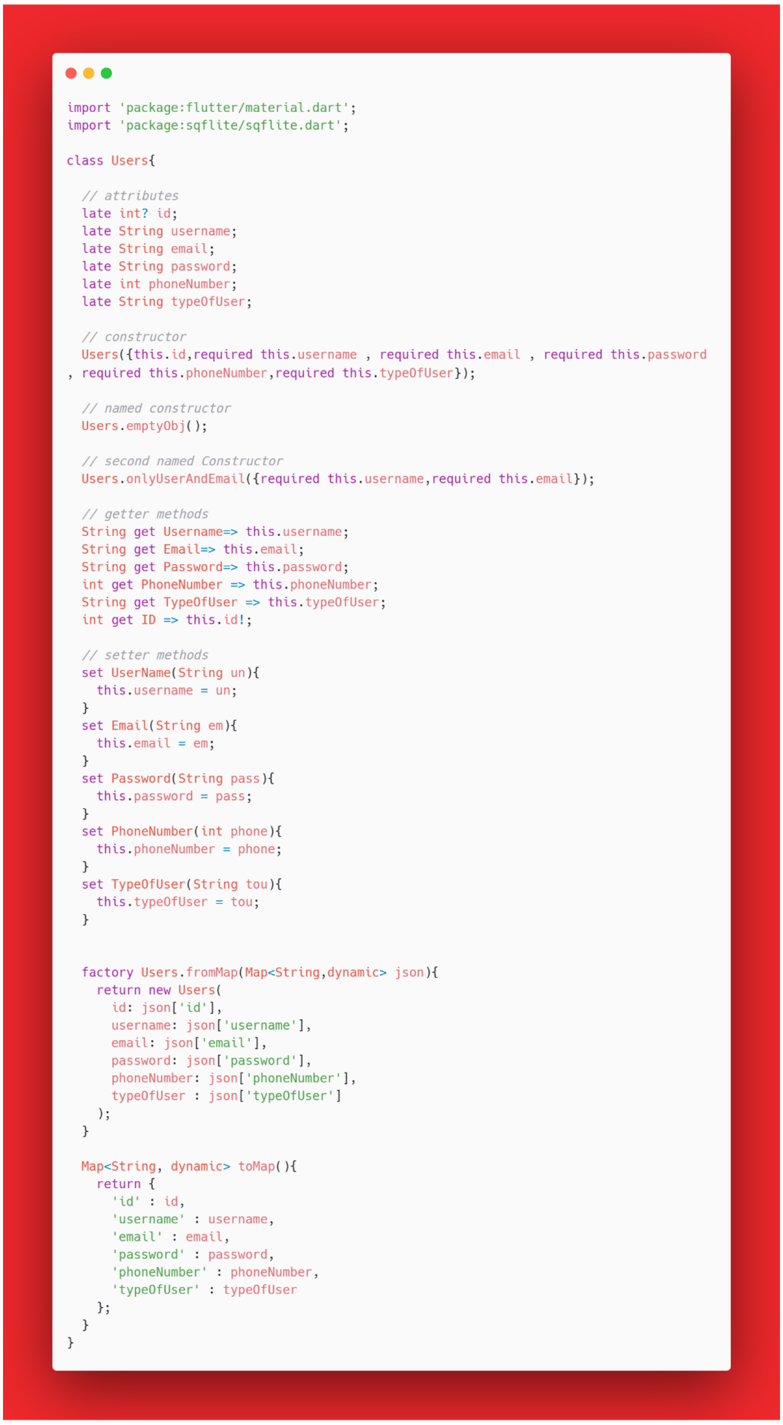
Software implementation, according to Sommerville (2016), is "the process of converting a system specification into an executable system." In order to come up with the intended functionality and behavior of the software system, the relevant lines of code related to a specific functionality elicited from the requirements must be written. The software design phase's design choices, algorithms, and data structures must be converted into a programming language that the computer can understand and interpret and use during the implementation phase.

5.2 Coding of System Main Functions



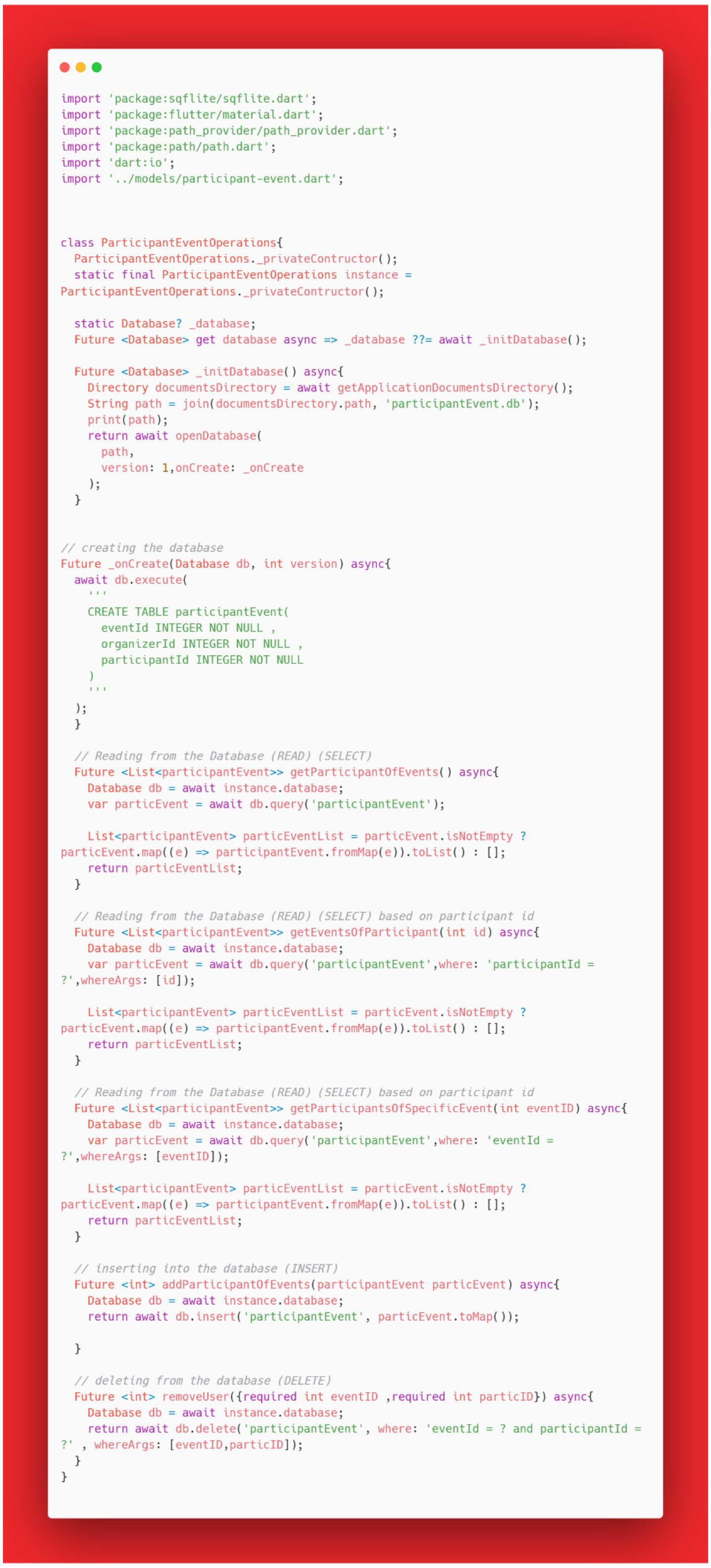
```
import 'package:flutter/material.dart';
import 'package:sqflite/sqflite.dart';

class eventProvider with ChangeNotifier{

  // attributes
  late int? id;
  late String eventName;
  late String eventDescription;
  late String eventLocation;
  late String eventStartDate;
  late String eventEndDate;
  late String eventType;
  late String eventPaidAmount;
  late String eventCategory;
  late int userWhoCreatedEvent;



  // the getter methods
  String get EventName=>this.eventName;
  String get EventDescription => this.eventDescription;
  String get EventLocation => this.eventLocation;
  String get EventStartDate => this.eventStartDate;
  String get EventEndDate => this.eventEndDate;
  String get EventType => this.eventType;
  String get EventPaidAmount => this.eventPaidAmount;
  String get EventCategory => this.eventCategory;
  int get UserWhoCreatedEvent => this.userWhoCreatedEvent;
  int get ID => this.id!;


  // the setter methods
  set ID(int id){
    this.id = id;
  }
  set EventName(String evn){
    this.eventName = evn;
    notifyListeners();
  }
  set EventDescription(String evd){
    this.eventDescription = evd;
    notifyListeners();
  }
  set EventLocation(String evl){
    this.eventLocation = evl;
    notifyListeners();
  }
  set EventStartDate(String evsd){
    this.eventStartDate = evsd;
    notifyListeners();
  }
  set EventEndDate(String eved){
    this.eventEndDate = eved;
    notifyListeners();
  }
  set EventType(String evt){
    this.eventType = evt;
    notifyListeners();
  }
  set EventPaidAmount(String evpam){
    this.eventPaidAmount = evpam;
    notifyListeners();
  }
  set EventCategory(String evc){
    this.eventCategory = evc;
    notifyListeners();
  }
  set UserWhoCreatedEvent(int userID){
    this.userWhoCreatedEvent = userID;
    notifyListeners();
  }

}
```

Figure 5.1 Event Provider



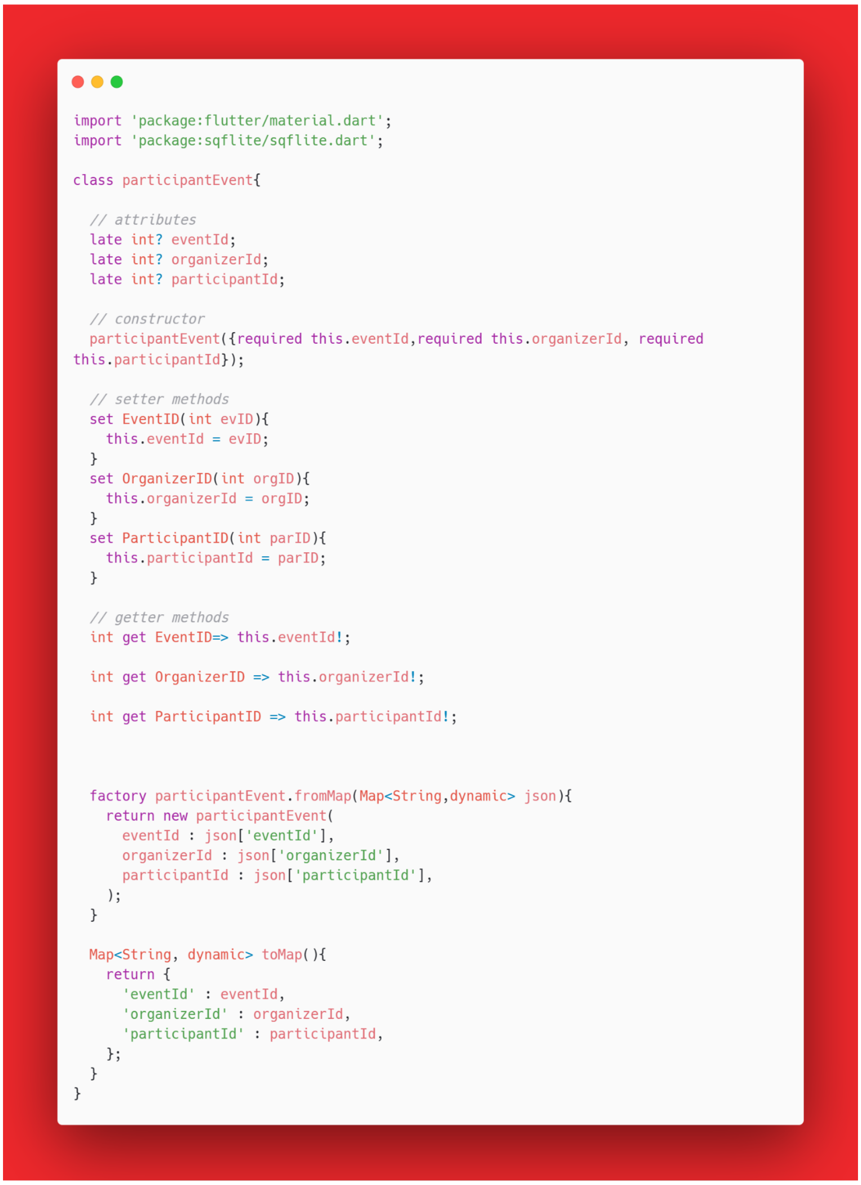
```
import 'package:flutter/material.dart';
import 'package:sqflite/sqflite.dart';

class Users{

  // attributes
  late int? id;
  late String username;
  late String email;
  late String password;
  late int phoneNumber;
  late String typeOfUser;

  // constructor
  Users({this.id,required this.username , required this.email , required this.password
, required this.phoneNumber,required this.typeOfUser});

  // named constructor
  Users.emptyObj();

  // second named Constructor
  Users.onlyUserAndEmail({required this.username,required this.email});

  // getter methods
  String get Username=> this.username;
  String get Email=> this.email;
  String get Password=> this.password;
  int get PhoneNumber => this.phoneNumber;
  String get TypeOfUser => this.typeOfUser;
  int get ID => this.id!;

  // setter methods
  set UserName(String un){
    this.username = un;
  }
  set Email(String em){
    this.email = em;
  }
  set Password(String pass){
    this.password = pass;
  }
  set PhoneNumber(int phone){
    this.phoneNumber = phone;
  }
  set TypeOfUser(String tou){
    this.typeOfUser = tou;
  }


  factory Users.fromMap(Map<String,dynamic> json){
    return new Users(
      id: json['id'],
      username: json['username'],
      email: json['email'],
      password: json['password'],
      phoneNumber: json['phoneNumber'],
      typeOfUser : json['typeOfUser']
    );
  }

  Map<String, dynamic> toMap(){
    return {
      'id' : id,
      'username' : username,
      'email' : email,
      'password' : password,
      'phoneNumber' : phoneNumber,
      'typeOfUser' : typeOfUser
    };
  }
}
```

Figure 5.2 Users Model



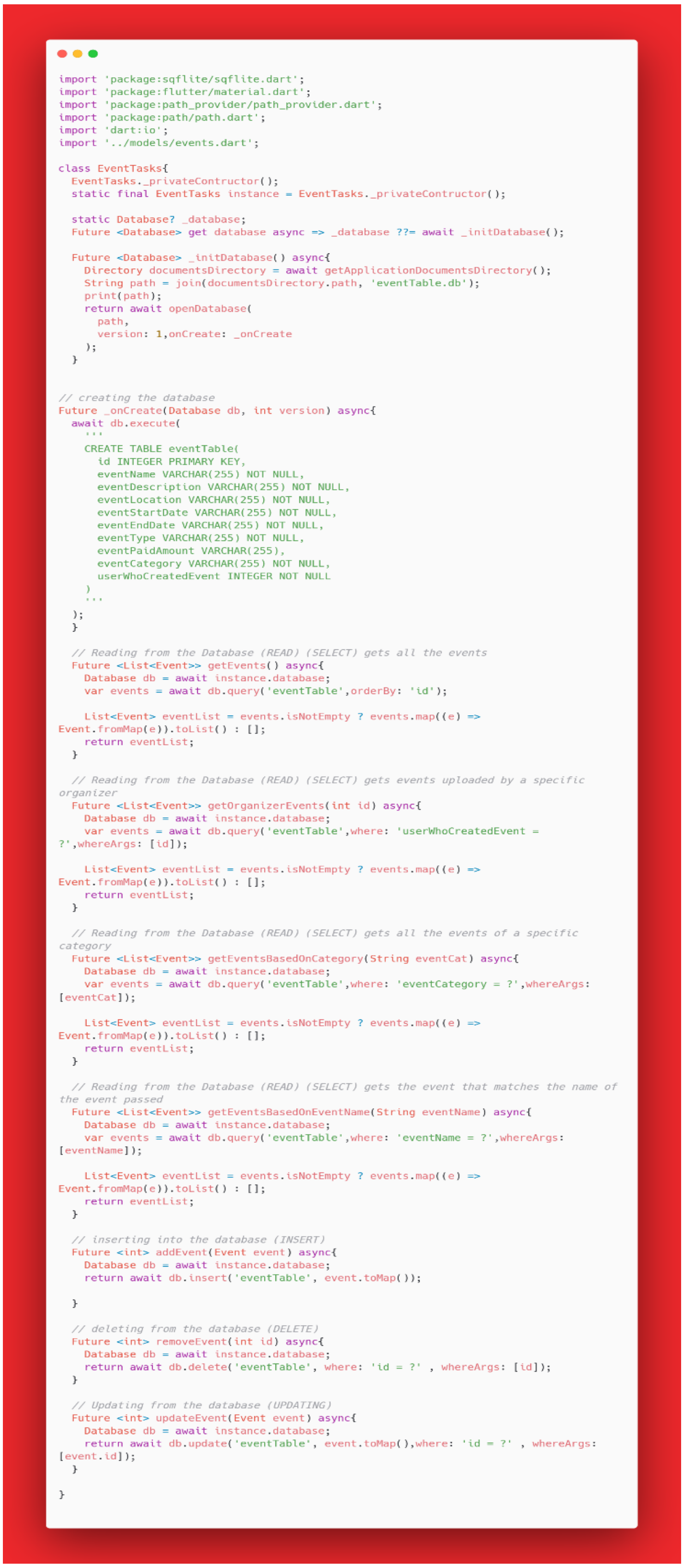
```dart
import 'package:sqflite/sqflite.dart';
import 'package:flutter/material.dart';
import 'package:path_provider/path_provider.dart';
import 'package:path/path.dart';
import 'dart:io';
import '../models/participant-event.dart';



class ParticipantEventOperations{
  ParticipantEventOperations._privateContructor();
  static final ParticipantEventOperations instance =
ParticipantEventOperations._privateContructor();

  static Database? _database;
  Future <Database> get database async => _database ??= await _initDatabase();

  Future <Database> _initDatabase() async{
    Directory documentsDirectory = await getApplicationDocumentsDirectory();
    String path = join(documentsDirectory.path, 'participantEvent.db');
    print(path);
    return await openDatabase(
      path,
      version: 1,onCreate: _onCreate
    );
  }


// creating the database
Future _onCreate(Database db, int version) async{
  await db.execute(
    '''
    CREATE TABLE participantEvent(
      eventId INTEGER NOT NULL ,
      organizerId INTEGER NOT NULL ,
      participantId INTEGER NOT NULL
    )
    '''
  );
  }

  // Reading from the Database (READ) (SELECT)
  Future <List<participantEvent>> getParticipantOfEvents() async{
    Database db = await instance.database;
    var particEvent = await db.query('participantEvent');

    List<participantEvent> particEventList = particEvent.isNotEmpty ?
particEvent.map((e) => participantEvent.fromMap(e)).toList() : [];
    return particEventList;
  }

  // Reading from the Database (READ) (SELECT) based on participant id
  Future <List<participantEvent>> getEventsOfParticipant(int id) async{
    Database db = await instance.database;
    var particEvent = await db.query('participantEvent',where: 'participantId =
?',whereArgs: [id]);

    List<participantEvent> particEventList = particEvent.isNotEmpty ?
particEvent.map((e) => participantEvent.fromMap(e)).toList() : [];
    return particEventList;
  }

  // Reading from the Database (READ) (SELECT) based on participant id
  Future <List<participantEvent>> getParticipantsOfSpecificEvent(int eventID) async{
    Database db = await instance.database;
    var particEvent = await db.query('participantEvent',where: 'eventId =
?',whereArgs: [eventID]);

    List<participantEvent> particEventList = particEvent.isNotEmpty ?
particEvent.map((e) => participantEvent.fromMap(e)).toList() : [];
    return particEventList;
  }

  // inserting into the database (INSERT)
  Future <int> addParticipantOfEvents(participantEvent particEvent) async{
    Database db = await instance.database;
    return await db.insert('participantEvent', particEvent.toMap());

  }

  // deleting from the database (DELETE)
  Future <int> removeUser({required int eventID ,required int particID}) async{
    Database db = await instance.database;
    return await db.delete('participantEvent', where: 'eventId = ? and participantId =
?' , whereArgs: [eventID,particID]);
  }
}
```

Figure 5.3 : ParticipantEvent Database



```dart
import 'package:flutter/material.dart';
import 'package:sqflite/sqflite.dart';

class participantEvent{

  // attributes
  late int? eventId;
  late int? organizerId;
  late int? participantId;

  // constructor
  participantEvent({required this.eventId,required this.organizerId, required
this.participantId});

  // setter methods
  set EventID(int evID){
    this.eventId = evID;
  }
  set OrganizerID(int orgID){
    this.organizerId = orgID;
  }
  set ParticipantID(int parID){
    this.participantId = parID;
  }

  // getter methods
  int get EventID=> this.eventId!;

  int get OrganizerID => this.organizerId!;

  int get ParticipantID => this.participantId!;



  factory participantEvent.fromMap(Map<String,dynamic> json){
    return new participantEvent(
      eventId : json['eventId'],
      organizerId : json['organizerId'],
      participantId : json['participantId'],
    );
  }

  Map<String, dynamic> toMap(){
    return {
      'eventId' : eventId,
      'organizerId' : organizerId,
      'participantId' : participantId,
    };
  }
}
```

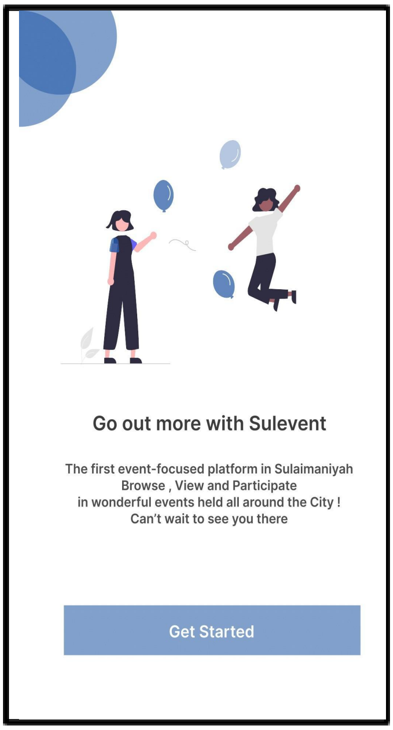
Figure 5.4 : ParticipantEvent Model



```dart
import 'package:sqflite/sqflite.dart';
import 'package:flutter/material.dart';
import 'package:path_provider/path_provider.dart';
import 'package:path/path.dart';
import 'dart:io';
import '../models/events.dart';

class EventTasks{
  EventTasks._privateContructor();
  static final EventTasks instance = EventTasks._privateContructor();

  static Database? _database;
  Future <Database> get database async => _database ??= await _initDatabase();

  Future <Database> _initDatabase() async{
    Directory documentsDirectory = await getApplicationDocumentsDirectory();
    String path = join(documentsDirectory.path, 'eventTable.db');
    print(path);
    return await openDatabase(
      path,
      version: 1,onCreate: _onCreate
    );
  }


// creating the database
Future _onCreate(Database db, int version) async{
  await db.execute(
    '''
    CREATE TABLE eventTable(
      id INTEGER PRIMARY KEY,
      eventName VARCHAR(255) NOT NULL,
      eventDescription VARCHAR(255) NOT NULL,
      eventLocation VARCHAR(255) NOT NULL,
      eventStartDate VARCHAR(255) NOT NULL,
      eventEndDate VARCHAR(255) NOT NULL,
      eventType VARCHAR(255) NOT NULL,
      eventPaidAmount VARCHAR(255),
      eventCategory VARCHAR(255) NOT NULL,
      userWhoCreatedEvent INTEGER NOT NULL
    )
    '''
  );
  }

  // Reading from the Database (READ) (SELECT) gets all the events
  Future <List<Event>> getEvents() async{
    Database db = await instance.database;
    var events = await db.query('eventTable',orderBy: 'id');

    List<Event> eventList = events.isNotEmpty ? events.map((e) =>
Event.fromMap(e)).toList() : [];
    return eventList;
  }

  // Reading from the Database (READ) (SELECT) gets events uploaded by a specific
organizer
  Future <List<Event>> getOrganizerEvents(int id) async{
    Database db = await instance.database;
    var events = await db.query('eventTable',where: 'userWhoCreatedEvent =
?',whereArgs: [id]);

    List<Event> eventList = events.isNotEmpty ? events.map((e) =>
Event.fromMap(e)).toList() : [];
    return eventList;
  }

  // Reading from the Database (READ) (SELECT) gets all the events of a specific
category
  Future <List<Event>> getEventsBasedOnCategory(String eventCat) async{
    Database db = await instance.database;
    var events = await db.query('eventTable',where: 'eventCategory = ?',whereArgs:
[eventCat]);

    List<Event> eventList = events.isNotEmpty ? events.map((e) =>
Event.fromMap(e)).toList() : [];
    return eventList;
  }

  // Reading from the Database (READ) (SELECT) gets the event that matches the name of
the event passed
  Future <List<Event>> getEventsBasedOnEventName(String eventName) async{
    Database db = await instance.database;
    var events = await db.query('eventTable',where: 'eventName = ?',whereArgs:
[eventName]);

    List<Event> eventList = events.isNotEmpty ? events.map((e) =>
Event.fromMap(e)).toList() : [];
    return eventList;
  }

  // inserting into the database (INSERT)
  Future <int> addEvent(Event event) async{
    Database db = await instance.database;
    return await db.insert('eventTable', event.toMap());

  }

  // deleting from the database (DELETE)
  Future <int> removeEvent(int id) async{
    Database db = await instance.database;
    return await db.delete('eventTable', where: 'id = ?' , whereArgs: [id]);
  }

  // Updating from the database (UPDATING)
  Future <int> updateEvent(Event event) async{
    Database db = await instance.database;
    return await db.update('eventTable', event.toMap(),where: 'id = ?' , whereArgs:
[event.id]);
  }

}
```

Figure 5.5 : EventTasks Database

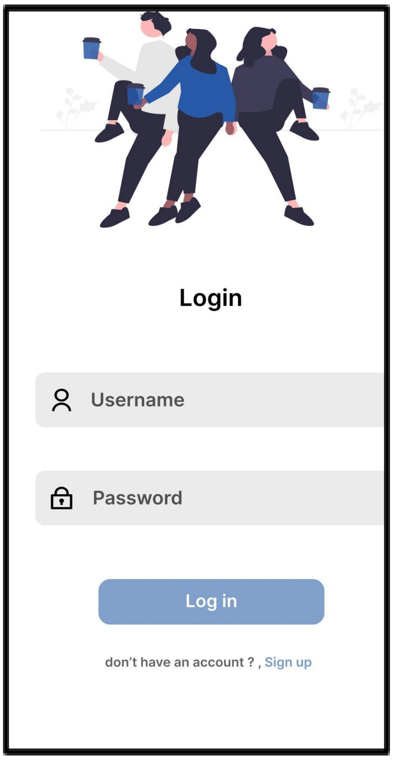
5.3 Interfaces of System Main Functions



Go out more with Sulevent

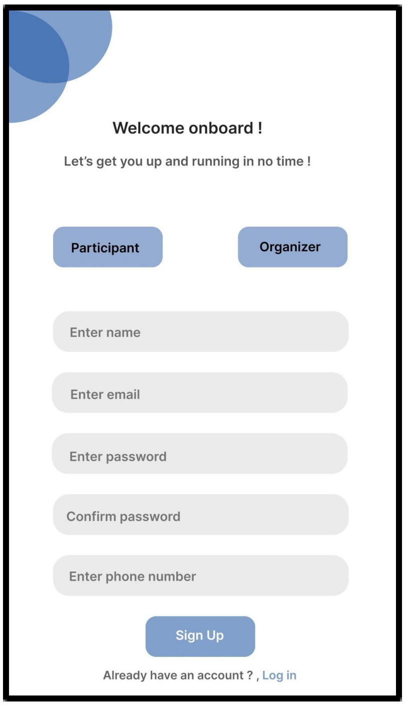
The first event-focused platform in Sulaimaniyah
Browse , View and Participate
in wonderful events held all around the City !
Can't wait to see you there

Figure 5.7 ; Start Page



don't have an account ? , Sign up

Figure 5.8 : Login Page

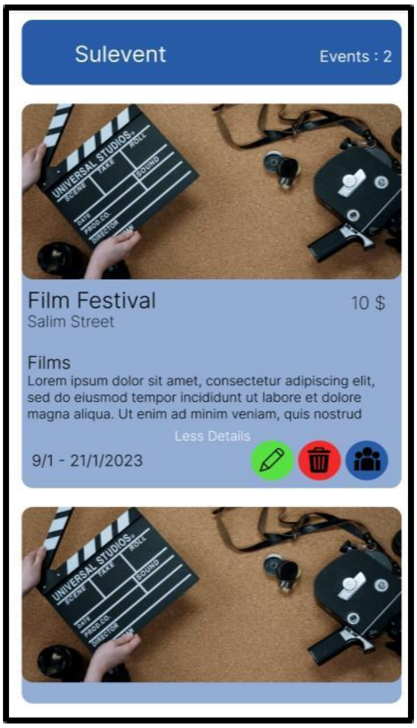


Welcome onboard !

Let's get you up and running in no time !

Already have an account ? , Log in

Figure 5.9 : Sign Up Page



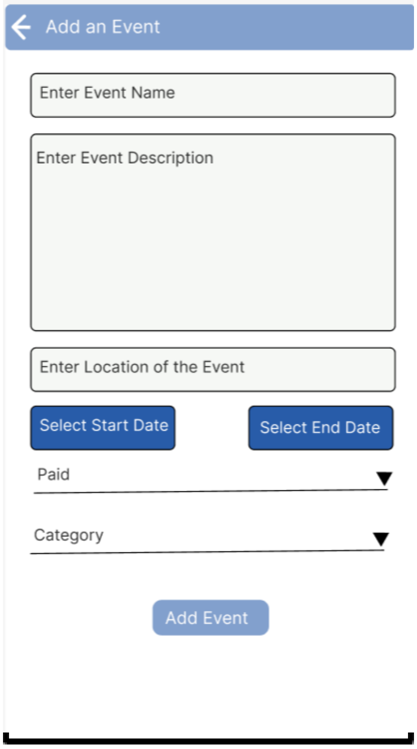
Sulevent Events : 2

Film Festival 10 $
Salim Street

Films
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud

9/1 - 21/1/2023

Figure 5.10 : Org. Home Page


Add an Event

Enter Event Name

Enter Event Description

Enter Location of the Event

Select Start Date

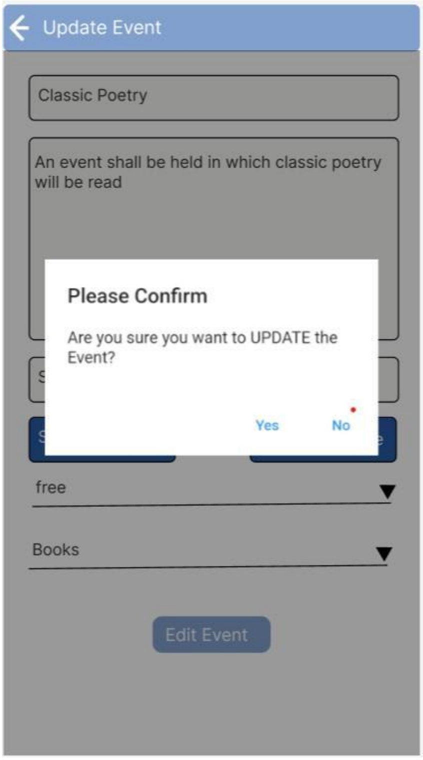
Select End Date

Paid ▼

Category ▼

Add Event

Figure 5.11 : Create Event


Update Event

Classic Poetry

An event shall be held in which classic poetry will be read

Please Confirm

Are you sure you want to UPDATE the Event?

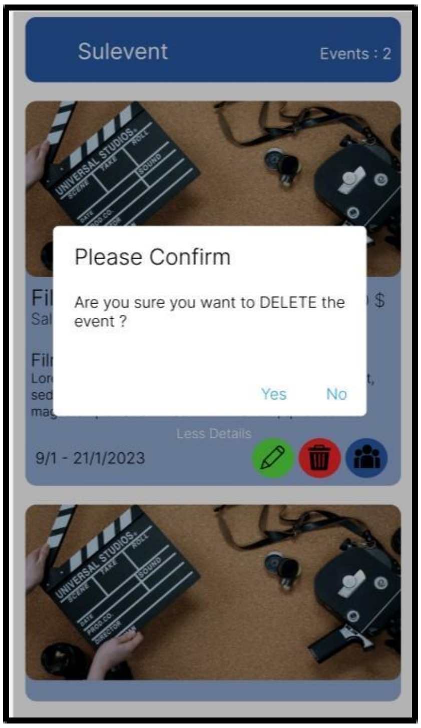
Yes No

free ▼

Books ▼

Edit Event

Figure 5.12 : Edit Event


Sulevent Events : 2

Please Confirm

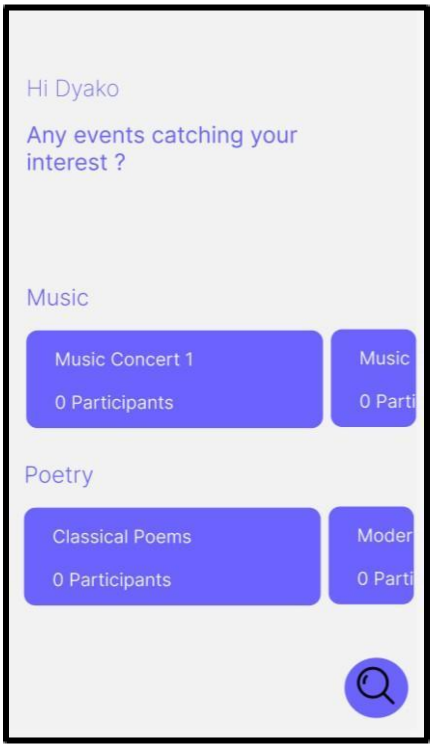
Are you sure you want to DELETE the event ?

Yes No

Less Details

9/1 - 21/1/2023

Figure 5.13 : Delete Event


Hi Dyako

Any events catching your interest ?

Music

Music Concert 1
0 Participants

Music
0 Parti

Poetry

Classical Poems
0 Participants

Moder
0 Parti

Figure 5.14 : Partic. Home

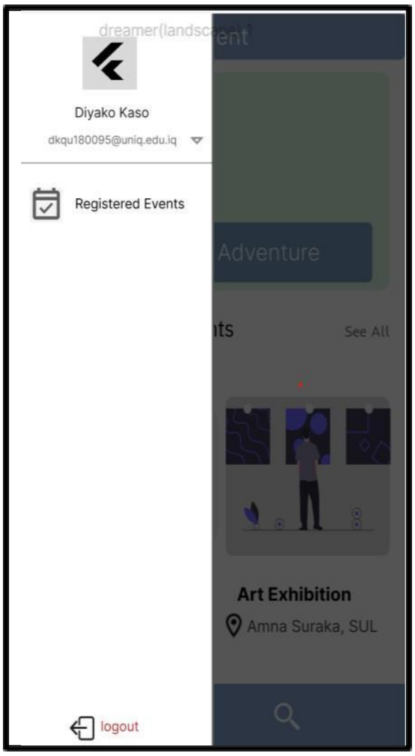
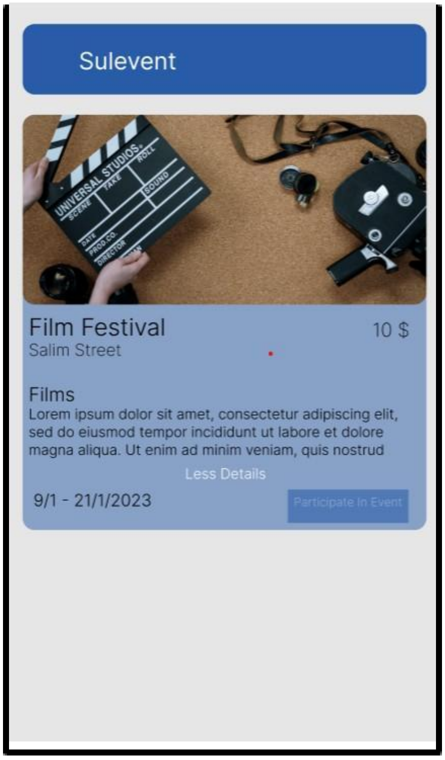


Figure 5.15 : Partic. Drawer



Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud

Figure 5.16 : View Event

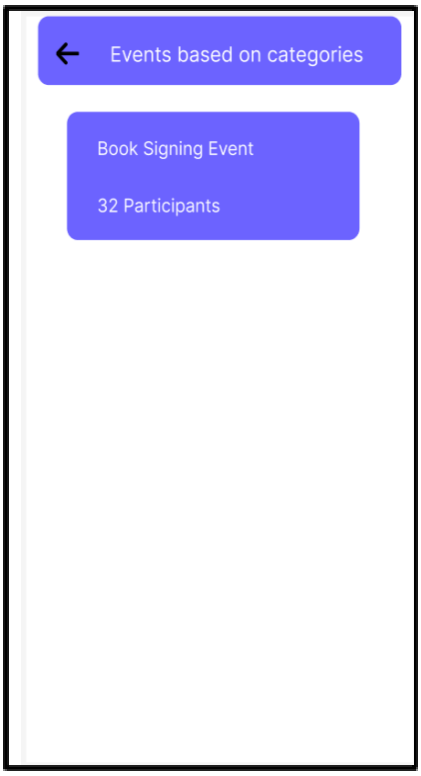


Figure 5.17 : Search Category

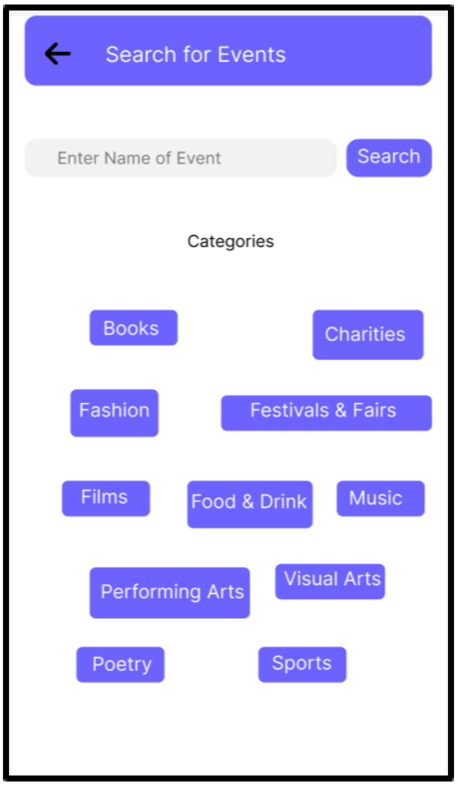


Figure 5.18 Search Page

5.4 Testing

Software Testing is a very important phase in the software development life cycle, it deals with the internal structure of the software application and dives deep into the processes and methods that run the system. Software Testing wants to identify multiple aspects of the code including bugs, errors and software defects that leads to system failure and software crash, in order to make sure that the software follows and meets the requirements that have been elicited by the user and how they would like the system to behave in terms of functionality. The software testing paves way for software developers to ensure that the system is reliable, efficient and robust when it has been deployed in it's domain to serve it's purpose which leads to minimizing the software failures and crashes and makes sure the customer receives the software they have asked for and meets the requirements.

According to Beizer (1990) The process of software testing is crucial for identifying bugs and removing defects and errors, This in turn will lead to a better system and a more reliable and efficient system which shall improve the overall quality of the software.

Software Testing consists primarily of Black Box and White Box testing. Black Box testing is a software testing technique that tests the behavior and functionality of the system without giving too much importance to the internal structure of the code rather it gives the system the input and predicts the output. The black box testing technique is praised for it's flexibility when it comes to different teams of software developers when they are testing the system they can test it without knowledge of the components and functions of the system. Another important point in black box testing is that the testers can test the system to make sure the system behaves exactly as it is expected according to the software requirements.

Beizer (1990), Praises the black box testing technique for it's importance in detecting software defects and bugs in order to make sure the system is safe and reliable to use.

Chen et al. (2012), promotes the usefulness of black box testing for making sure the system is robust and correct in terms of the output it generates based on the input of the users.

The following subsection discusses further how black box testing is used to test the system against different metrics and boundaries in order to make sure the system is correct and reliable to use.

5.5 Black Box Testing

The Testing phase requires specific testing methods and approaches that tests for the user’s input and makes sure that the input value the user has inputted into the software does not cause the software to crash. A software testing approach called boundary value analysis (BVA) which is a form of (black box) testing technique that tests the input inputted by the user and checks the output shall be used for the testing phase. Boundary Value Analysis (BVA) searches for flaws or faults at the edges or extreme values of input that has been inputted by the user into the system. BVA assists in evaluating the system's behavior and determining if it correctly handles boundary values by concentrating on three key areas which are (Invalid | Valid | Invalid). To enhance test coverage and raise the performance of software systems, this strategy is frequently used in software testing. (Beizer,1990).

The primary objective of BVA is to identify faults that are present just at the boundaries. BVA helps in the system testing phase by making sure that the system successfully handles extreme values, edge cases, and unusual conditions by choosing test cases at the boundaries. This method improves testing and increases the chance of finding bugs. (Kaner et al, 1999).

5.6 System Flow

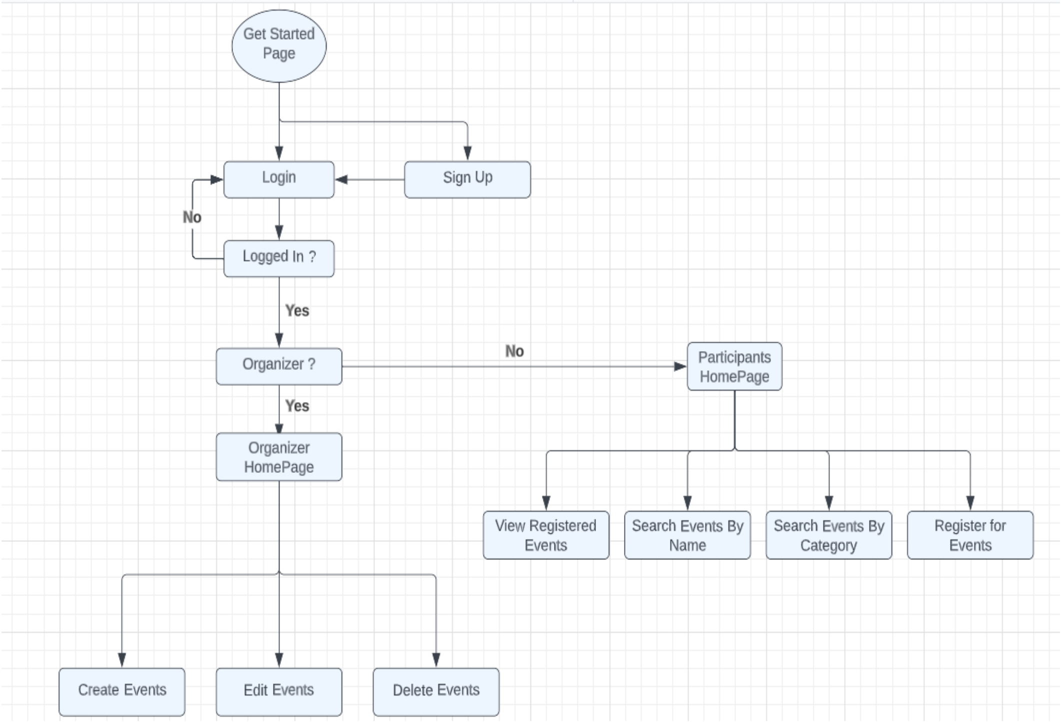


Figure 5.19 : System flow diagram

5.7 Input Output Verification

In order to verify if the user’s input does not cause any defects or exceptions within the system, we need to put the Black Box testing technique which is Boundary Value Analysis (BVA) to the test.

We have 4 main windows that contain fields for user to input data and which are prone to erroneous data and they are (Login Page , SignUp page).

Table 5.1 refers to the BVA testing of (Login Page) in which the conditions are as follows :

- 3 < Email < 30 – Valid

- Email < 3 – Invalid
- Email > 30 – Invalid
- 4 < Password < 30 –Valid
- Password < 4 –Invalid
- Password > 30 –Invalid

Table 5.1 : Login BVA Testing

Test Case ID	Input Data	Expected Results	Actual Results	Pass / Fail
TC 1_01	Email : Diyako@gmail.com Password : 1234	Input Accepted	Input Accepted	Pass
TC 1_02	Email : ThisisAveryLongThisisAveryLongEmail@gmail.com Password : 1234	Input Rejected	Input Rejected	Fail
TC 1_03	Email : oc@gmail.com Password : 1234	Input Rejected	Input Rejected	Fail
TC 1_04	Email : Roc@gmail.com Password : @534@534@534@534@534@534@534@534@534@534	Input Rejected	Input Rejected	Fail
TC 1_05	Email : Asm@gmail.com Password : 23	Input Rejected	Input Rejected	Fail
TC 1_06	Email : MAC@gmail.com Password : 12345	Input Accepted	Input Accepted	Pass

Table 5.2 refers to the BVA testing of (Sign Up Page) in which the conditions are as follows :

- 3 < Name< 20 – Valid
- Name< 3 – Invalid
- Name> 20 – Invalid
- 3 < Email < 30 – Valid
- Email < 3 – Invalid
- Email > 30 – Invalid
- 4 < Password < 30 –Valid
- Password < 4 –Invalid
- Password > 30 –Invalid
- 4 < ConfirmPassword< 30 –Valid
- ConfirmPassword< 4 –Invalid
- ConfirmPassword> 30 –Invalid
- PhoneNumber == 11 – Valid
- PhoneNumber < 11 – Invalid
- PhoneNumber > 11 – Invalid

Table 5.2 : Sign Up BVA Testing

Test Case ID	Input Data	Expected Results	Actual Results	Pass / Fail
TC 2_01	Name : Diyako Email : Diyako@gmail.com Password : 1234 Confirm Password : 1234 PhoneNumber : 07701234567	Input Accepted	Input Accepted	Pass

TC 2_02	Name : Naz Email : ThisisAveryLongThisisAveryLongEmail@gmail.com Password : 1234 Confirm Password : 12345 Phone Number : 07705543366	Input Rejected	Input Rejected	Fail
TC 2_03	Name : Karwan Email : oc@gmail.com Password : 1234 Confirm Password : 12345 PhoneNumber : 0770555	Input Rejected	Input Rejected	Fail
TC 2_04	Name : ArivanKamalQaderAghaHussein Email : Nitorg@gmail.com Password : 12344545	Input Rejected	Input Rejected	Fail

	Confirm Password : 12344545 PhoneNumber : 07705557744			
TC 2_05	Name : Karwan Jamal Email : KarwanJ@gmail.com Password : Karwan123 Confirm Password : Karwan123 PhoneNumber : 07705834567	Input Accepted	Input Accepted	Pass

5.8 User Testing

User Acceptance Testing (UAT) is a form of testing that is directly conducted by the target users of the developed system to ensure that the functionality and processes that have been developed for the system does match the requirements and expectations set by the target users for the system. (Bertolino et al , 2018).

User Acceptance Testing (UAT) focuses on testing the system in order to measure the effectiveness and efficiency of the developed system and to what extent is the system usable by testing it’s Usability to make sure it meets the requirements. (Bertolino et al , 2018).

User Acceptance Testing (UAT) is aimed at ensuring that the system performs all of the operations that have been required by the Users of the system by completing all of the functions and tasks efficiently that allows the user to reach their desired goals. (Li et al , 2016).

Usability testing in (UAT) is related to the testing of the system against the metrics that are used for determining the Usability of the system, this includes determining the user-friendliness of the User Interface (UI) elements and design in order to ensure smooth functioning of the system operations and ease of navigation between different windows and be able to access all the necessary data and functions in order to reach the desired outcome. (Li et al , 2016).

Acceptance Criteria testing in (UAT) performs a checkup for the software against those conditions that have been determined by the stakeholders and organization that states the software completely fulfills the standards and expectations of the organization and hence the software is complete and no more development needs to be done. (Li et al , 2016).

Before the testing phase starts we can define two user profiles that are potential users of the system software that shall be using the system to perform their tasks related to event management and promotion. Table 5.3 shows the user profile of those two potential users.

Table 5.3 : Potential User Profiles

Details	**Business Owner**	**Chief Executive Officer**
Age	35	44
Gender	Female	Male
Experience with Technologies	Beginner	Intermediate
Experience with mobile application	The user does not have a lot of technical background to use complex applications and have only used applications like youtube and whatsapp and facebook (not utilizing all the functionalities)	The user has an intermediate understanding of technology and complex application and has used complex softwares like Microsoft Office 365 and collaboration softwares.

In need of an event management system ?	Yes	Yes
Usage of the System	The business owner is in need of an event management system in order to promote their new business events like publishing new products or product show cases in which the user shall use the system to promote their events and manage the attendees who are going to attend that event.	The CEO is in need of an event management system in order to prepare and organize conferences and meetings and product planning meetings and team progress meetings in order to make the employees aware of meeting times and venues and see who is going to attend the meeting and who cannot.

Table 5.4 illustrates the different tasks that can be performed by both potential users of our system. The steps of the execution for each task has been labeled accordingly as well as the result of what shall be the outcome of those steps when they are executed in the sequence. The event shall be labeled (pass) if the user performed all of the steps specified and was able to reach their desired goal. The (comments/suggestions) section shall be labeled in order to give comments about different elements of the user experience.

Table 5.4 : User Acceptance Testing (UAT) tasks

Te s t No.	**Description of Task**	**Steps of Execution**	**Result**	**Pass/ Fail**	**Comment s/ Suggestio ns**
1.	Create A new Account	press signup button choose type of user enter name enter email enter password enter confirmation password enter phone number press on signup	a new account has been created for the user	Pass	The process was straightforw ard as the input fields had icons (metaphors) that indicated the purpose of the input field and what to input

2.	Login with your account	enter email enter password press on login button	User has logged in **Alternate Route :** The user has entered wrong email or password	Pass	NA
3.	create event on organizer's home page	press the + (plus) icon enter name of event enter description of event enter location of event select event start date select event end date select type of event if (eventTyp e == paid) enter amount select event category press the (add event) button	a new event has been created and uploaded to the system	Pass	The + (plus) icon in the home page of the organizer does contribute to the accessibility of the function and makes it easy to find and press in order to be taken to the add event window and enter the event details, each input field of the add event window had (hint texts) that told the user to what to input
4.	Edit event on organizer's home page	press the (pen) icon on an existing event's card enter new name of event enter new descriptio n of event enter new location of event	The existing event information has been updated	Pass	the (pen) icon on the event card on the organizer home page makes it easy to understand how to edit the event and all of the event's previous details are shown in the input fields of the edit event window and

		select new event start date select new event end date select new type of event if (eventTyp e == paid) enter amount select new event category press the (edit event) button confirm the confirmation edit event message			once erased the (hint text) is present to aid the user to understand what to input
5.	Delete event on organizer's home page	press the (bin) icon on an existing event's card confirm the delete event confirmation message	The existing event should be deleted	Pass	The (bin) icon is very straightforward it is a universal symbol for (removing) or (deleting) files and items, the confirmation message prevents accidental deletion of the events making the operation more safe and reliable
6.	Participate in an event on the participant	press on an event in the participant home page view the details of the event press the (participate in event) button	A confirmation message has been shown that the user has successfully participated in the event	Pass	the process is very straightforward as the user only needs to click on an event in order to view the entire details of the event and the (participate in event) button is present.

		press (ok) on the (successf ul participati on) alert box			allowing the user to register with ease, after the registration for an event a pop up message is shown to the user that tells them you have registered for the event.
7.	Search for events by name	press the (magnifyi ng glass) icon on the participant's home page enter the name of event in the input field and press search find the event and view it's details	the event that is being searched for should come up in the results if matches the name of the event in the database **Alternate route**: event does not show up due to event name incorrectness or simply is not in the database	Pass	the magnifying glass icon which acts as a universal visual (metaphor) for finding items contributes to the ease of use of the application in a way that the participant will understand immediately that the button is for searching for events, the user shall be taken to the search window in which they will be shown an input field that asks for the name of the event to be inputted and the (search) button is present to be clicked

8.	Search for event by category	press the (magnifying glass) icon on the participant's home page press a category press any event of that category that matches your interest to view it's details	those events that have been uploaded to the system under the name of the chosen category shall be shown to the user	Pass	same as the previous action the magnifying glass icon is pressed in order to be taken to the search window in which the user despite having an option to search by event name , they can choose from the various categories shown to them in the form of buttons and once they are clicked it takes them to a window in which all of the events under that category name shall be shown to them.
9.	view registered events	press the drawer icon on the top left of the participant home page press (registered events) list tile view the events that your have registered for	the registered events shall be shown to the user to either view or cancel their participation in it	Pass	on the participant's home page the drawer icon above the welcoming message is shown to them and once clicked it will open a drawer in which there will be a list tile with (registered events) title, once clicked they are shown

5.10 Chapter Summary

In chapter 5 we have discussed the importance of testing in the software development life cycle as it handles and finds defects and bugs that may lead to software crashes in turn leading to loss of lives.

Chapter 6

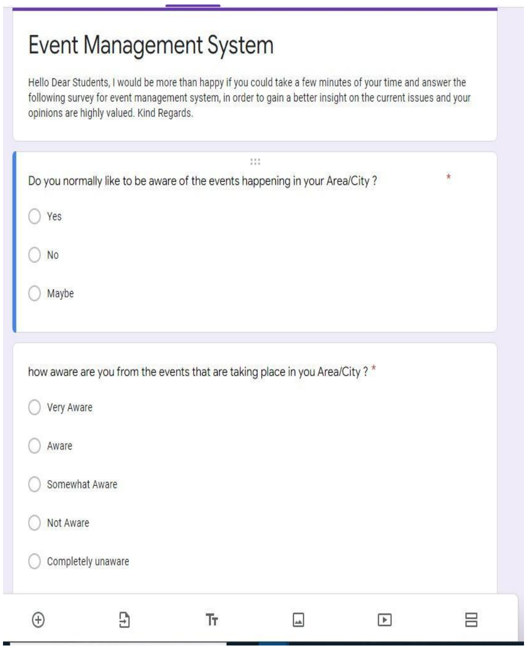
CONCLUSION

6.1 Introduction

in this chapter the significance of the (Design and Implementation of Online Event Management System) shall be discussed and put forth with the support of the findings that have been found by survey responses from the respondents of the survey to identify the problems being faced by the community in general in terms of finding events that they desire , the method they use to find such events and to what extent have those methods been efficient and to answer the question of (is there a need for such a system to exist ?) all of those above shall be discussed in this chapter.

6.2 Achievement of Project Objectives

a survey had been created in which (10) questions were asked to be answered by the survey respondents as shown in (Figure 6.1) - (Figure 6.5)



Event Management System

Hello Dear Students, I would be more than happy if you could take a few minutes of your time and answer the following survey for event management system, in order to gain a better insight on the current issues and your opinions are highly valued. Kind Regards.

Do you normally like to be aware of the events happening in your Area/City ? *

○ Yes

○ No

○ Maybe

how aware are you from the events that are taking place in you Area/City ? *

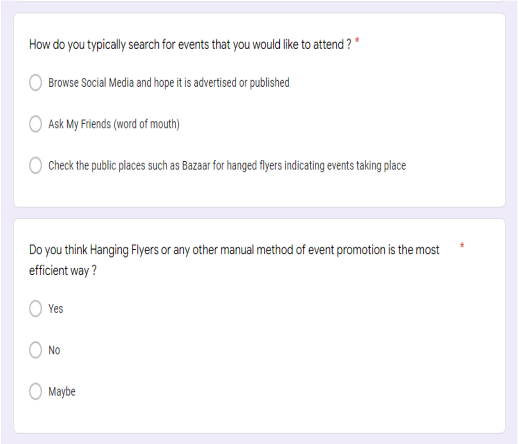
○ Very Aware

○ Aware

○ Somewhat Aware

○ Not Aware

○ Completely unaware



How do you typically search for events that you would like to attend ? *

○ Browse Social Media and hope it is advertised or published

○ Ask My Friends (word of mouth)

○ Check the public places such as Bazaar for hanged flyers indicating events taking place

Do you think Hanging Flyers or any other manual method of event promotion is the most efficient way ? *

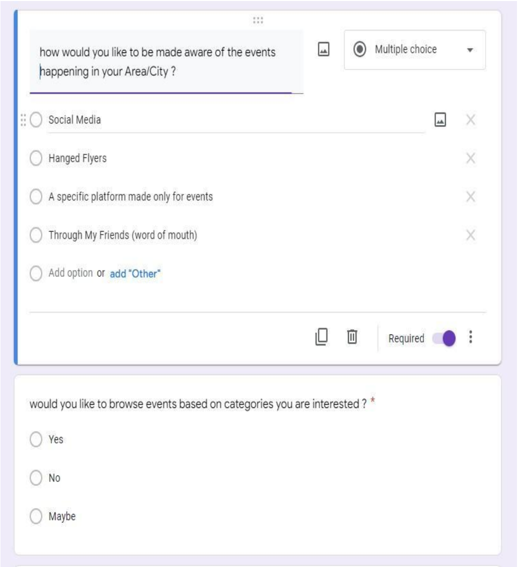
○ Yes

○ No

○ Maybe

Figure 6.1 : survey question 1-2

Figure 6.2 : survey question 3-4



how would you like to be made aware of the events happening in your Area/City ?

○ Social Media

○ Hanged Flyers

○ A specific platform made only for events

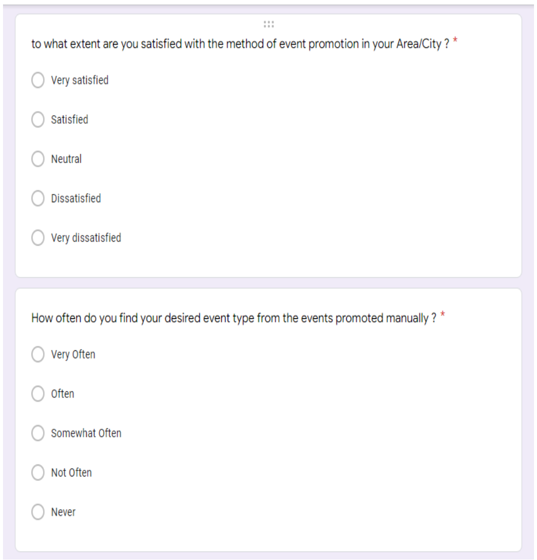
○ Through My Friends (word of mouth)

would you like to browse events based on categories you are interested ? *

○ Yes

○ No

○ Maybe



to what extent are you satisfied with the method of event promotion in your Area/City ? *

○ Very satisfied

○ Satisfied

○ Neutral

○ Dissatisfied

○ Very dissatisfied

How often do you find your desired event type from the events promoted manually ? *

○ Very Often

○ Often

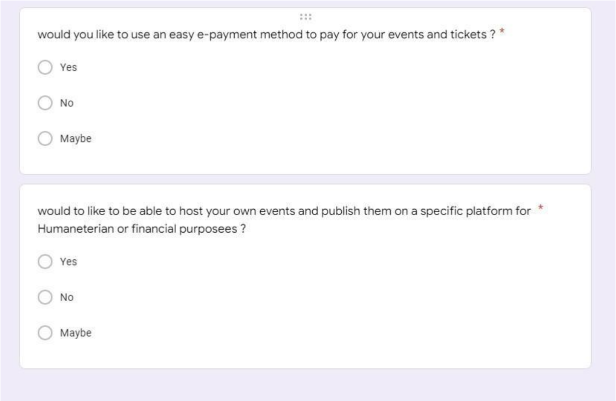
○ Somewhat Often

○ Not Often

○ Never

Figure 6.3 : survey question 5-6

Figure 6.4 : survey question 7-8



would you like to use an easy e-payment method to pay for your events and tickets ? *

○ Yes

○ No

○ Maybe

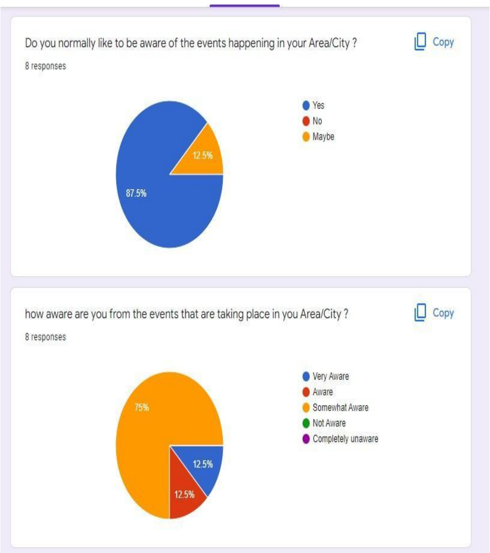
would to like to be able to host your own events and publish them on a specific platform for Humaneterian or financial purposees ? *

○ Yes

○ No

○ Maybe

Figure 6.5 : survey question 9-10



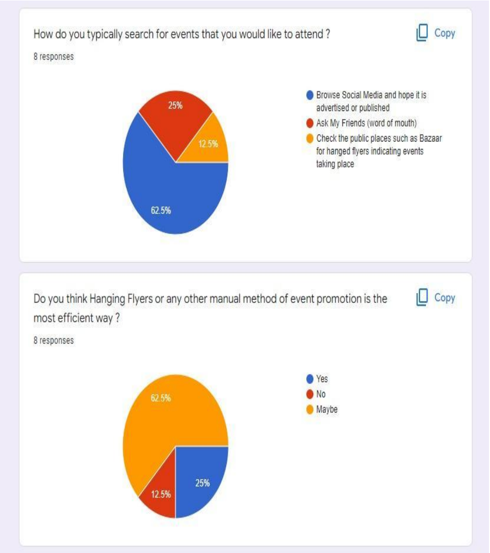
Do you normally like to be aware of the events happening in your Area/City ?

8 responses

how aware are you from the events that are taking place in you Area/City ?

8 responses

Figure 6.6 : responses for question 1-2



How do you typically search for events that you would like to attend ?

8 responses

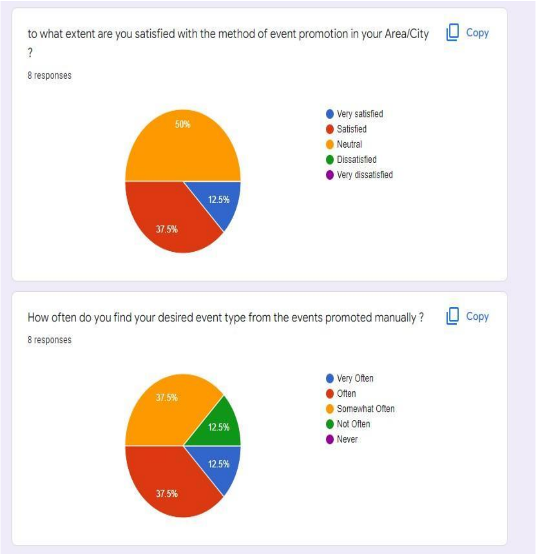
Copy
● Browse Social Media and hope it is advertised or published
● Ask My Friends (word of mouth)
● Check the public places such as Bazaar for hanged flyers indicating events taking place
25%
12.5%
62.5%

Do you think Hanging Flyers or any other manual method of event promotion is the most efficient way ?

8 responses

Copy
● Yes
● No
● Maybe
62.5%
25%
12.5%

Figure 6.7 : responses for question 3-4

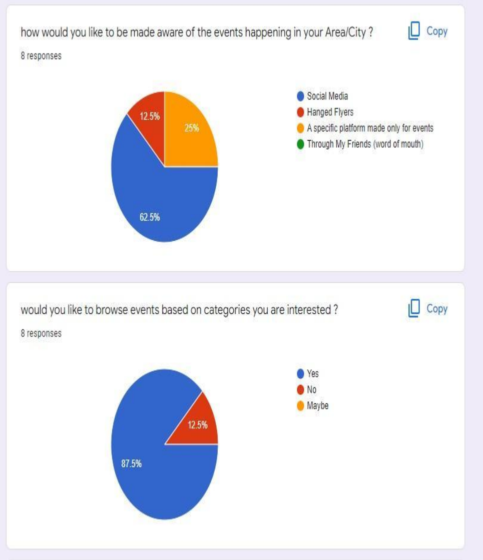


to what extent are you satisfied with the method of event promotion in your Area/City ?

8 responses

How often do you find your desired event type from the events promoted manually ?

8 responses



how would you like to be made aware of the events happening in your Area/City ?

8 responses

Copy
● Social Media
● Hanged Flyers
● A specific platform made only for events
● Through My Friends (word of mouth)
12.5%
25%
62.5%

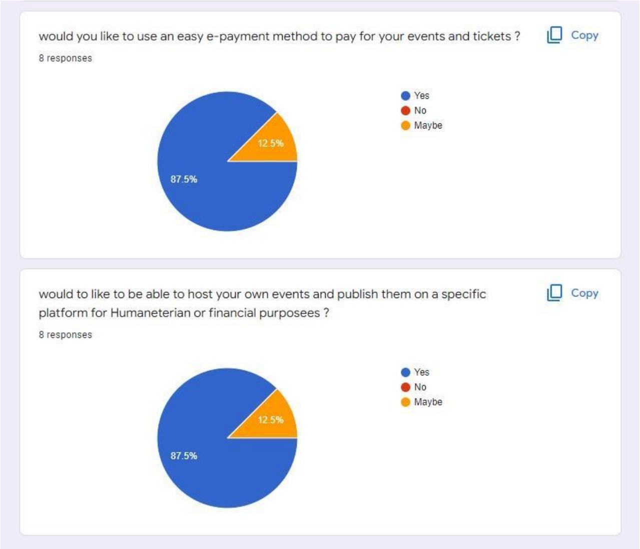
would you like to browse events based on categories you are interested ?

8 responses

Copy
● Yes
● No
● Maybe
12.5%
87.5%

Figure 6.8 : responses for question 5-6

Figure 6.9 : responses for question 7-8



would you like to use an easy e-payment method to pay for your events and tickets ?

8 responses

would to like to be able to host your own events and publish them on a specific platform for Humaneterian or financial purposees ?

8 responses

Figure 6.10 : response for question 9-10

based on the above results we can see that many students want to be aware of the events happening around them but the majority of them are somewhat aware of the events that are happening around them in their areas and their cities.The Majority of the students they either go on the social media platforms hoping that the events that they would like to attend have been published online or advertised or they ask their friends to know about the events that are taking place.Majority of the student think that the hanging flyers in the public places is somewhat efficient way of promotion but 12.5% of the students think that it is not the most effective way to promote for an event.Half of the students who took the survey are unsure about the method of event promotions that are taking place in their cities and areas although there are a small minority of students who are satisfied with how the events are promoted in their areas and cities.As for finding the desired event type from the events that have been promoted manually the result is 50-50 as some students have said to found their desired events that they were looking for , however the other half have said that they did not often find their desired event from the hanging flyers or from asking their friends.Most of the students would like to be able to browse events based on the categories they are interested in.

6.3 Suggestions for Future Improvement

this system would be quite new in the region and in Sulaymaniyah City and the absence of other similar systems would mean that this system needs to be used by the users and to provide feedback on what they think needs to be improved and to be added and the process of maintenance and modification is always iterative in which the new version of the system is released to the public to test and then the developers receive feedback on what to improve and what to add , The same approach will be taken for the (Design and Implementation of Online Event Management System).

REFERENCES

Oden, C. (n.d.). *Design and Implementation of an Online Event Management System, projectTopics , Design and Implementation of an Online Event Management System - Project Topics*.

Rafalson, B. (2018). *Why You Need Event Management Software, bizzabo, Why You Need Event Management Software - Bizzabo | Bizzabo*.

Kanabar, G. (2018). How does Eventbrite work? Everything You Need to Know About Eventbrite and How it Works , Alphanso Tech , How does Eventbrite work? Everything You need to Know About Eventbrite and How it Works.

Ikram, R. (2022). What's Up With FeverUp's Travel Entertainment Service?

Graham, E. (n.d.). Is Fever Up Legit or Scam? (Fever Up Reviews), Find Legit ,Is Fever Up Legit or Scam? (Fever Up Reviews) - Find Legit.

Page, M. (2012). Using Facebook Events - an example , smartinsights , Using Facebook Events - an example | Smart Insights.

Larman, C., & Vodde, B. (2016). Large-Scale Scrum More with LeSS , craig larman.

Layton, M. (2015). Scrum for Dummies , Google Books , Scrum For Dummies - Mark C. Layton - Google Books.

Owens, T. (2014). An Introduction to Scrum: Understanding and Applying the Software . google books , An Introduction to Scrum: Understanding and Applying the Software - Ted Owens - Google Books.

Dave Raggett, Arnaud Le Hors and Ian Jacobs. (1998, April 24th), HTML 4.0 Specification , immagic , HTML 4.0 Specification (immagic.com) (Technical Specification Document)

Drumond, C. (2018). What is Scrum? , Atlassian , Scrum - what it is, how it works, and why it.

Westland, J. (2022). Scrum Methodology: Roles, Events & Artifacts,Projectmanager.com , Scrum Methodology: Roles, Events & Artifacts (projectmanager.com).

Davey, L. (2019). The Best Event Management Software: 10 Apps to Plan and Manage Your Business Events , zapier, The Best Event Management Software: 10 Apps to Plan and Manage Your Business Events.

Abrahamsson, P., Salo, O., Ronkainen, J., & Warsta, J. (2002). Agile Software Development Methods:Review and Analysis.

Faust, S. (2020). Using Google´s Flutter Framework for the Development of a Large-Scale Reference Application, Using Google´s Flutter Framework for the Development of a Large-Scale.

Mirghani, A. (2020). JAVA and DART programming languages: Conceptual comparison. In Sebuah Kajian Pustaka: (semanticscholar.org).

Sinha, S. (2020). Quick start guide to dart programming language,Google Books , Quick Start Guide to Dart Programming: Create High-Performance Applications. In Sanjib Sinha - Google Books.

Beizer, B., & Wiley, J. (1996). Black box testing: Techniques for functional testing of software and systems. IEEE Software, 13(5), 98. doi:10.1109/ms.1996.536464

Badgett, T., Thomas, T. M., Sandler, C., & Myers, G. J. (2013). The art of software testing: Second edition. Barking, England: Lulu.com.

Badgett, T., Thomas, T. M., Sandler, C., & Myers, G. J. (2013). The art of software testing: Second edition. Barking, England: Lulu.com.

Kaner, C., Bach, J., Johnson, R., Lawrence, B., Falk, J., Nguyen, H. Q., & Moore Literary Agency. (2013). Testing Computer Software (3rd ed.). Nashville, TN: John Wiley & Sons.

Beizer, B. (1990). Software Testing Techniques. Van Nostrand Reinhold.

Badgett, T., Thomas, T. M., Sandler, C., & Myers, G. J. (2013). The art of software testing: Second edition. Barking, England: Lulu.com.

Pressman, R., & Maxim, B. (2019). Loose leaf for software engineering: A practitioner's approach (9th ed.). McGraw-Hill Education.

Sommerville, I. (2016). Software Engineering (10th ed.). Pearson.

Chen, T. Y., Kuo, F. C., & Wang, K. H. (2012). Automated black-box testing of software applications using machine learning techniques. Expert Systems with Applications, 39(1), 247–257.

Li, Z., Xie, M., & Huang, G. (2016). An empirical study of factors affecting user acceptance of mobile applications. Applied Soft Computing, 38, 809–817.

Bertolino, A., & Muccini, H. (2018). User acceptance testing: A systematic literature review. Information and Software Technology, 100, 137–159.

Mackay, J.(2019). Software Development Process: How to Pick The Process That's Right For You , Planio , Last ,available at : Software Development Process: How to Pick The Process That's Right For You | Planio.

APPENDIX

Software Testing Documentation

Design and Implementation of Online Event Management System

Version 1.0

14th - July -2022

Faculty of Engineering and Science - Software Engineering

Prepared by: Diyako Kaso

Revision Page

a. **Overview**

This version of the system describes the general overview and introduction of the system and then dives to the use case diagrams and sequence and activity diagrams of each use case along with a detailed description of each use case using usecase descriptions as well as many types of system and design and additional constrains have been discussed

a. Target Audience

System Architects , Software Engineers , Software Quality Assurance

b. Project Team Members

Diyako Kaso Chalk

c. Version Control History

Version	Primary Author(s)	Description of Version	Date Completed
<1.0>	Diyako Kaso Chalak	This version of the system describes the general overview and introduction of the system and then dives to the use case diagrams and sequence and activity diagrams of each use case along with a detailed description of each use case using usecase descriptions as	

		well as many types of system and design and additional constrains have been discussed	

Note:
This template is an annotated outline for a software testing document. It is based on IEEE standards 829, 1008, 1012 and 1012a. This document covers: unit testing (the verification of individual sub-systems or components of the system against their specifications), integration testing (the testing of inter-operating sub-systems or components against their specifications) and system testing (both verification against the system specification, and validation against the user requirements).This template has been simplified and customized to meet the need of SCSJ2203 course at Faculty of Computing, UTM. Compiled by Ruhaidah Samsudin, PhD and checked by Masitah Ghazali, PhD and Shahida Sulaiman, PhD (revised on 14 May 2016).

Table of Contents

1. Introduction

Event management systems are nowadays being used extensively to manage events of all sorts and types. The most important factor that led to the mass public usage of such event systems is innovation in live event technology (Brandon,2018). Event management systems are defined as systems that are used to manage and organize activities related to a certain event or events (Chrisantus Oden , n.d.). An event management software helps event organizers to register their events and to allow the customers to book their attendance (check-in) and to promote their events on the software. Some event management software only specializes in a type of functionality such as participant booking (check-in) this leads to a limited amount of functionality. Typical operations of an event management System are but not limited to (Event Registration, Event Marketing) (Brandon, 2018).

The technical aspects of the development of our own proposed system shall be discussed in terms of the programming language that is going to be used to program the functionalities of the system, such as dart programming language and the flutter framework and such technology shall be discussed, labelling their performance and security and the features that they are possessing.

1.1 Purpose

- To Elicit the requirements of the community in terms of Event Management
- To design and implement the system to address those issues
- To test the system in terms of Functional and non-functional requirements and completeness

1.2 Scope

- The system will be developed only for the City of Sulaymaniyah in the initial phases

- The system development will use Mobile Application languages and frameworks such as dart programming language and the flutter framework for mobile application development.

- The system will be developed using Visual Studio Code code Editor.

1.3 Definitions, Acronyms and Abbreviation

None

1.4 References

Rafalson,B . (2018,January 3rd) , *Why You Need Event Management Software,* bizzabo, Why You Need Event Management Software - Bizzabo | Bizzabo

Chrisantus Oden , n.d. , *Design and Implementation of an Online Event Management System,* projectTopics , Design and Implementation of an Online Event Management System - Project Topics

1.5 Overview

The aim of this project is to re-ignite the cultural movement in the city by developing an efficient and easy to use (Design and implementation of online event Management system) system to drive more traffic to the events and to make people aware of the events in order to increase the networking opportunities and to expand the opportunity spaces beyond the stereotypical jobs and ways of making a living and the classical way of networking with the aim of paving way for an opportunity filled city that has opportunities on every corner and every person from every background and major can make contribution to the cultural and artistic and academic movement in our city.

Test Cases, Data and Expected Results

1.1 Test TC001 for Module <Event Handling>: <Create Event(UC003)>

This test contains the following test cases:

UC003_01: e.g. Event Creation Form (Event)

Test Case ID	Input data	Expected result	Actual result	Pass / Fail
TC001_03_01	Art Exh3bt7n	Name of event must not contain numerical characters	alert message is shown to ask the user to re-enter the name of the event without numerical characters	Fail
TC001_03_02	Art Exhibition that is hosted in Erbil City Near the Belgium Embassy near Empire towers in the city center	Name of Event too long	Alert message is shown to ask the user to re-enter the name of the event briefly	Fail
TC001_03_03	Art Exhibition (Erbil)	Name of event acceptable	the name is accepted	Pass
TC001_03_04	$Donation Party$	Symbols are not allowed to be included in the name of the event	alert message is shown to ask the user to re-enter the name of the event without Symbols	Fail
TC001_03_05	1-4-2021	a Date that is referring to yesterday and the days before are not acceptable to use as event **start date**	Alert message is shown to ask the user to re-enter the Date of current date or future dates	Fail
TC001_03_06	7-7-2011	a Date that is referring to yesterday and the	Alert message is shown to ask the user to re-enter the	Fail

		days before are not acceptable to use as event **End date**	Date of current date or future dates	
TC001_03_07	999343599.00	high and unreasonable numbers are not acceptable in the payment amount specification of the event	Alert message is shown to ask the user to re-enter the amount of the payment for an event but in a reasonable range	Fail
TC001_03_08	-664.00	LOW and unreasonable numbers are not acceptable in the payment amount specification of the event	Alert message is shown to ask the user to re-enter the amount of the payment for an event but in a reasonable range	Fail
TC001_03_09	Cat3	Category must not contain any numbers	alert message is shown to ask the user to re-enter the Category	Fail

1.2 Test TC001 for Module <Event Handling>: <Modify Event(UC004)>

This test contains the following test cases:

UC004_1: e.g. Modify Form (Event)

Test Case ID	**Input data**	**Expected result**	**Actual result**	**Pass / Fail**
TC001_04_01	1-1-2022	a Date that is referring to yesterday and the days before are not acceptable to use as event **start date**	Alert message is shown to ask the user to re-enter the Date of current date or future dates	Fail

TC001_04_02	4-5-2021	a Date that is referring to yesterday and the days before are not	Alert message is shown to ask the user to re-enter the Date of current date or future dates	Fail

		acceptable to use as event **End date**		
TC001_04_03	9999999.00	high and unreasonable numbers are not acceptable in the payment amount specification of the event	Alert message is shown to ask the user to re-enter the amount of the payment for an event but in a reasonable range	Fail
TC001_04_04	-554.00	LOW and unreasonable numbers are not acceptable in the payment amount specification of the event	Alert message is shown to ask the user to re-enter the amount of the payment for an event but in a reasonable range	Fail
TC001_04_05	Mod4rn A3t Museum	Name of event must not contain numerical characters	alert message is shown to ask the user to re-enter the name of the event without numerical characters	Fail
TC001_04_06	Art Exhibition that is hosted in Erbil City Near the Belgium Embassy near Empire towers in the city center	Name of Event too long	Alert message is shown to ask the user to re-enter the name of the event briefly	Fail
TC001_04_07	Art Exhibition (Sulaimaniyah)	Name of event acceptable	the name is accepted	Pass
TC001_04_08	$Cancer $ Donation $	Symbols are not allowed to be included in the name of the event	alert message is shown to ask the user to re-enter the name of the event without Symbols	Fail
TC001_04_09	abc123	Category must not contain any numbers	alert message is shown to ask the user to re-enter the Category	Fail

1.3 Test TC001 for Module <Event Handling>: <Delete Event(UC005)>

This test contains the following test cases:

UC005_1: e.g. Delete event(Event)

Test Case ID	Input data	Expected result	Actual result	Pass / Fail
TC001_05_01	click the bin icon on the event	the delete confirmation message must pop up	Delete confirmation message does not pop up.	Fail
TC001_05_02	click the bin icon on the event	the delete confirmation message must pop up	the delete message confirmation message pops up	Pass
TC001_05_03	The delete confirmation messege is confirmed	the event is supposed to be deleted	the event is not deleted	Fail
TC001_05_04	The delete confirmation messege is confirmed	the event is supposed to be deleted	the event is deleted	Pass

1.4 Test TC001 for Module <Event Handling>: <Browse Event(UC006)>

This test contains the following test cases:

UC006_1: e.g. Browse event(Event)

Test Case ID	Input data	Expected result	Actual result	Pass / Fail
TC001_06_01	the events button is clicked	the events on the system must be shown	the events are not shown	Fail

TC001_06_02	the events button is clicked	the events on the system must be shown	the events are shown	Pass
TC001_06_03	Category of the events to be shown are chosen	the events related to that category must be shown	the events are not showing	Fail
TC001_06_04	Category of the events to be shown are chosen	the events related to that category must be shown	the events are showing	Pass

1.5 Test TC001 for Module <Event Handling>: <Give Feedback(UC007)>

This test contains the following test cases:

UC007_1: e.g. Give Feedback(Feedback)

Test Case ID	**Input data**	**Expected result**	**Actual result**	**Pass / Fail**
TC001_07_01	The event is clicked and a comment/feedback is written and submitted	the comment/feedback has been accepted and saved	the comment has not been saved	Fail
TC001_07_02	The event is clicked and a comment/feedback is written and submitted	the comment/feedback has been accepted and saved	the comment has been saved	Pass
TC001_07_03	This 7s a c0mm3nt	Comments/feedback must not contain symbols	Alert message is shown in order to ask the user to re-enter the comment/feedback properly	Fail

1.6 Test TC001 for Module <Event Handling>: <Give Rating(UC008)>

This test contains the following test cases:

UC008_1: e.g. Give Rating(Rating)

Test Case ID	Input data	Expected result	Actual result	Pass / Fail
TC001_08_01	The event is clicked and a rating has been chosen and submitted	the rating has been accepted and saved	the rating has not been saved	Fail
TC001_08_02	The event is clicked and a rating has been chosen and submitted	the rating has been accepted and saved	the rating has not been saved	Pass

1.7 Test TC002 for Module <Payment>: <Payment(UC001)>

This test contains the following test cases:

UC001_1: e.g.Payment(Payment)

Test Case ID	Input data	Expected result	Actual result	Pass / Fail
TC002_01_01	the choose payment method screen is shown and confirmed	the payment has been chosen and the user shall be registered for the event and a receipt generated	the user has not been registered and a receipt has not been generated	Fail
TC002_01_02	the choose payment method screen is	the payment has been chosen and the user shall be registered for the	the user has been registered and a receipt has been generated	Pass

	shown and confirmed	event and a receipt generated		
TC002_01_03	the choose payment method screen is shown and confirmed	the payment has been chosen and the user shall be registered for the event and a receipt generated	the user has been registered BUT receipt has not been generated	Fail
TC002_01_04	the choose payment method screen is shown and confirmed	the payment has been chosen and the user shall be registered for the event and a receipt generated	the user has been registered and receipt has been generated	Pass

1.8 Test TC002 for Module <Payment>: <Generate reciept(UC002)>

This test contains the following test cases:

UC002_1: e.g.Generate Reciept(Reciept)

Test Case ID	Input data	Expected result	Actual result	Pass / Fail
TC002_02_01	the choose payment method screen is shown and confirmed	the payment has been chosen and the user shall be registered for the event and a receipt generated	the user has not been registered and a receipt has not been generated	Fail
TC002_02_02	the choose payment method screen is shown and confirmed	the payment has been chosen and the user shall be registered for the event and a receipt generated	the user has been registered and a receipt has been generated	Pass
TC002_02_03	the choose payment method screen is shown and confirmed	the payment has been chosen and the user shall be registered for the event and a receipt generated	the user has been registered BUT receipt has not been generated	Fail
TC002_02_04	the choose payment	the payment has been chosen and	the user has been registered and	Pass

	method screen is shown and confirmed	the user shall be registered for the event and a receipt generated	receipt has been generated	

1.9 Test TC003 for Module <Event Registration>: <Register for an Event(UC009)>

This test contains the following test cases:

UC009_1: e.g.Generate Receipt(Receipt)

Test Case ID	Input data	Expected result	Actual result	Pass / Fail
TC003_09_01	the register for an event has been clicked	the choose payment option is opened if the event is paid or registered automatically if the payment is free	no action	Fail
TC003_09_02	the register for an event has been clicked	the choose payment option is opened if the event is paid or registered automatically if the payment is free	event type is free but no alert message is indicating the user has been registered	Fail
TC003_09_03	the register for an event has been clicked	the choose payment option is opened if the event is paid or registered automatically if the payment is free	event type is free and alert message is indicating the user has been registered	Pass

TC003_09_04	the register for an event has been clicked	the choose payment option is opened if the event is paid or registered automatically if the payment is free	event type is paid and the payment method screen is shown but no action when confirmed	Fail

TC003_09_05	the register for an event has been clicked	the choose payment option is opened if the event is paid or registered automatically if the payment is free	event type is paid and the payment method screen is shown and alerts the user they have been registered when confirmed	Pass

1.10 Test TC003 for Module <Payment>: <Payment(UC001)>

This test contains the following test cases:

UC001_1: e.g.Payment(Payment)

Test Case ID	**Input data**	**Expected result**	**Actual result**	**Pass / Fail**
TC003_01_01	the choose payment method screen is shown and confirmed	the payment has been chosen and the user shall be registered for the event and a receipt generated	the user has not been registered and a receipt has not been generated	Fail
TC003_01_02	the choose payment method screen is shown and confirmed	the payment has been chosen and the user shall be registered for the event and a receipt generated	the user has been registered and a receipt has been generated	Pass
TC003_01_03	the choose payment method screen is shown and confirmed	the payment has been chosen and the user shall be registered for the event and a receipt generated	the user has been registered BUT receipt has not been generated	Fail
TC003_01_04	the choose payment method screen is shown and confirmed	the payment has been chosen and the user shall be registered for the event and a receipt generated	the user has been registered and receipt has been generated	Pass

1.11 Test TC003 for Module <Payment>: <Generate reciept(UC002)>

This test contains the following test cases:

UC002_1: e.g.Generate Reciept(Reciept)

Test Case ID	Input data	Expected result	Actual result	Pass / Fail
TC003_02_01	the choose payment method screen is shown and confirmed	the payment has been chosen and the user shall be registered for the event and a receipt generated	the user has not been registered and a receipt has not been generated	Fail
TC003_02_02	the choose payment method screen is shown and confirmed	the payment has been chosen and the user shall be registered for the event and a receipt generated	the user has been registered and a receipt has been generated	Pass
TC003_02_03	the choose payment method screen is shown and confirmed	the payment has been chosen and the user shall be registered for the event and a receipt generated	the user has been registered BUT receipt has not been generated	Fail
TC003_02_04	the choose payment method screen is shown and confirmed	the payment has been chosen and the user shall be registered for the event and a receipt generated	the user has been registered and receipt has been generated	Pass

Test Approach Analysis

The test approach that I have used is a mixture of black box and white box testing in whcih the black box has been used in usecases that emphasize and focus on the corectness af the steps of the operations.White box testing has been used in test cases that focus on and emphasize particular values in which the internal implementation of the module and piece of code is examined in order to make sure the data is being set and the variables and classes contain the right value.

UC003: Create Event

- Name of event must have at least 6 characters and at maximum 25
- Name of Event Must Not contain symbols and numbers
- Description of the event must be maximum 150 characters
- Event Start Date must be more or equal to the current date
- Event End Date must be more or equal to the current date
- Paid Event amount must be between 1.00$ and 1000$
- Event Category Must not contain Numbers
- Event Category must have atleast 7 character and at maximum 20

Event Name

EP class 1 (valid): 5 < Event Name < 26

EP class 2 (invalid): Event Name< 6

EP class 3 (invalid): Event Name> 25

EP class 4 (invalid): Event Name other than character

BVA values for username: 5, 6, 25, 26

Event Description

EP class 1 (valid): Event Description <

151 EP class 2 (invalid): Event Name >

150 **BVA values for username: 150, 151**

Event Start Date

EP class 1 (valid): Event Start Date >= Current Date

EP class 2 (invalid): Event Start Date < Current Date

BVA values for username: 13th-July-2022, 14th-July-2022

Event End Date

EP class 1 (valid): Event EndDate >= Current

Date EP class 2 (invalid): Event End Date <

Current Date

BVA values for username: 13th-July-2022, 14th-July-2022

Event Payment Amount (Paid Event)

EP class 1 (valid): 1.00$ < Event Payment Amount < 1000.00$

EP class 2 (invalid): Event Payment Amount > 1000.00$

EP Class 3 (Invalid): Event Payment Amount < 1.00$

BVA values for username: 0.99 , 1.00 , 1000.00 , 1001.00

Event Category

EP class 1 (valid): 7 < Event Category <

20 EP class 2 (invalid): Event Category >

20 EP Class 3 (Invalid): Event Category

< 7

EP Class 4 (invalid} Event Category other than {characters}

BVA values for username: 6 , 7 ,20 , 21

UC004: Modify Event

- Name of event must have at least 6 characters and at maximum 25

- Name of Event Must Not contain symbols and numbers

- Description of the event must be maximum 150 characters
- Event Start Date must be more or equal to the current date
- Event End Date must be more or equal to the current date
- Paid Event amount must be between 1.00$ and 1000$
- Event Category Must not contain Numbers
- Event Category must have atleast 7 character and at maximum 20

Event Name

EP class 1 (valid): 5 < Event Name < 26

EP class 2 (invalid): Event Name< 6

EP class 3 (invalid): Event Name> 25

EP class 4 (invalid): Event Name other than character

BVA values for username: 5, 6, 25, 26

Event Description

EP class 1 (valid): Event Description <

151 EP class 2 (invalid): Event Name >

150 **BVA values for username: 150, 151**

Event Start Date

EP class 1 (valid): Event Start Date >= Current Date

EP class 2 (invalid): Event Start Date < Current Date

BVA values for username: 13th-July-2022, 14th-July-2022

Event End Date

EP class 1 (valid): Event EndDate >= Current

Date EP class 2 (invalid): Event End Date <

Current Date

BVA values for username: 13th-July-2022, 14th-July-2022

Event Payment Amount (Paid Event)

EP class 1 (valid): 1.00$ < Event Payment Amount < 1000.00$

EP class 2 (invalid): Event Payment Amount > 1000.00$

EP Class 3 (Invalid): Event Payment Amount < 1.00$

BVA values for username: 0.99 , 1.00 , 1000.00 , 1001.00

Event Category

EP class 1 (valid): 7 < Event Category <

20 EP class 2 (invalid): Event Category >

20 EP Class 3 (Invalid): Event Category

< 7

EP Class 4 (invalid} Event Category other than {characters}

BVA values for username: 6 , 7 ,20 , 21

Additional Material

APPENDIX A. TRACEABILITY MATRIX

Test Case ID	**Use Case ID/** **Sequence Diagram ID**	**Package ID**
TC001 for <Event Handling> ● TC001_03 ● TC001_04 ● TC001_05	UC003 ● SD001 UC004 ● SD001 UC005	P002

● TC001_06 ● TC001_07 ● TC001_08	● SD001 UC006 ● SD001 UC007 ● SD001 UC008 ● SD001	
TC002 for <Payment> ● TC002_01 ● TC002_02	UC001 ● SD001 UC002 ● SD001	P001
TC003 for <Event Registration> ● TC003_01 ● TC003_02 ● TC003_03	UC003 ● SD001 UC001 ● SD001 UC002 ● SD001	P003 P001

KERANA TUHAN UNTUK MANUSIA • UNIVERSITI TEKNOLOGI MALAYSIA •

UTM

UNIVERSITI TEKNOLOGI MALAYSIA

Software Requirements Specification

Design and Implementation Of Online Event
Management System

Version 1.0

13th - July - 2022

Faculty of Engineering and Science-Software Engineering

Revision Page

a. Overview

This version of the system describes the general overview and introduction of the system and then dives to the use case diagrams and sequence and activity diagrams of each use case along with a detailed description of each use case using usecase descriptions as well as many types of system and design and additional constrains have been discussed

b. Target Audience

System Architects , Software Engineers , Software Quality Assurance

c. Project Team Members

Diyako Kaso Chalk

d. Version Control History

Version	Primary Author(s)	Description of Version	Date Completed
<1.0>	Diyako Kaso Chalak	This version of the system describes the general overview and introduction of the system and then dives to the use case diagrams and sequence and activity diagrams of each use case along with a detailed description of each use case using usecase descriptions as	

		well as many types of system and design and additional constrains have been discussed	

Note:
This Software Requirements Specification (SRS) template is based on IEEE Std 830-1998, organized by modules according to system features (Appendix A.5 of the IEEE Std, 830-1998, Section 5) and customized to meet the need of SCSJ2203 course at Faculty of Computing, UTM. Compiled and checked by Shahida Sulaiman, PhD on 20 March 2016. Examples of models are from Satzinger (2011).

Table of Contents

3.1.2 Hardware Interfaces

3.1.3 Software Interfaces

3.1.4 Communication Interfaces

3.2 System Features

3.2.1 Module <Name of Module1>

3.2.1.1 UC001: Use Case <Name of Use Case1>

3.2.1.2 UC002: Use Case <Name of Use Case2>

3.2.1.3 UC003: Use Case <Name of Use Case3>

3.2.2 Module <Name of Module2>

3.2.1.1 UC004: Use Case <Name of Use Case4>

3.2.1.2 UC005: Use Case <Name of Use Case5>

3.2.n Module <Name of the *n* Module>

3.3 Performance Requirements

3.4 Design Constraints

3.5 Software System Attributes

3.6 Other Requirements

Appendices (if any)

1. Introduction

Event management systems are nowadays being used extensively to manage events of all sorts and types. The most important factor that led to the mass public usage of such event systems is innovation in live event technology (Brandon,2018). Event management systems are defined as systems that are used to manage and organize activities related to a certain event or events (Chrisantus Oden , n.d.). An event management software helps event organizers to register their events and to allow the customers to book their attendance (check-in) and to promote their events on the software. Some event management software only specializes in a type of functionality such as participant booking (check-in) this leads to a limited amount of functionality. Typical operations of an event management System are but not limited to (Event Registration, Event Marketing) (Brandon, 2018).

The technical aspects of the development of our own proposed system shall be discussed in terms of the programming language that is going to be used to program the functionalities of the system, such as dart programming language and the flutter framework and such technology shall be discussed, labelling their performance and security and the features that they are possessing.

1.1 Purpose

- To Elicit the requirements of the community in terms of Event Management
- To design and implement the system to address those issues
- To test the system in terms of Functional and non-functional requirements and completeness

1.2 Scope

- The system will be developed only for the City of Sulaymaniyah in the initial phases

- The system development will use Mobile Application languages and frameworks such as dart programming language and the flutter framework for mobile application development.

- The system will be developed using Visual Studio Code code Editor.

1.3 Definitions, Acronyms and Abbreviation

None

1.4 References

Rafalson,B . (2018,January 3rd) , *Why You Need Event Management Software,* bizzabo, Why You Need Event Management Software - Bizzabo | Bizzabo

Chrisantus Oden , n.d. , *Design and Implementation of an Online Event Management System,* projectTopics , Design and Implementation of an Online Event Management System - Project Topics

1.5 Overview

The aim of this project is to re-ignite the cultural movement in the city by developing an efficient and easy to use (Design and implementation of online event Management system) system to drive more traffic to the events and to make people aware of the events in order to increase the networking opportunities and to expand the opportunity spaces beyond the stereotypical jobs and ways of making a living and the classical way of networking with the aim of paving way for an opportunity filled city that has opportunities on every corner and every person from every background and major can make contribution to the cultural and artistic and academic movement in our city.

2. Overall Description

One of the most important signs of the livelihood of a society and the advancement of their cultures and mindsets is the number of activities that are done in that specific area or place that target multiple academic, arts, philosophical and musical tastes with the aim of nurturing the minds of that society and setting forth an environment where those tastes and abilities can grow and transform for the better. Events are an important part of the social life as it gathers multiple people from different academic and ethical and knowledge backgrounds together to interact on a common subject or theme in which the exchange of ideas might lead to brighter ideas and the networking among those people might lead to better life opportunities regardless of the event that has its own importance to open up the minds of the society and to drive them to a better destination to guide their train of thoughts.

Kurdistan Region in Iraq is a culturally rich region as it stands in the middle of the Iran , Turkey and Iraq those regions had interactions with our region throughout history and our cultures to some extent have borrowed beautiful cultures from each other whether they are in music , arts , Islamic arts , literature and Theatres and Clothing those culture borrowing is reflected in our culture and the activities that we do. Approximately all of Kurdistan is constantly full of Activities and events held to drive the society forwards and more specifically Sulaymaniyah city as it is the Capital of Culture and is the spearhead of cultural and artistic activities and events as most of the well-known and respected poets and literature writers come from this region and most of the activities are to be conducted in the city of Sulaymaniyah.

Unfortunately till now Kurdistan region and more specifically Sulaymaniyah City has not yet utilized the technological advancements to better improve the way the events are held and people to be notified. Tens of events are held weekly but only a little portion of the society knows about it due to the fact that events are published in the Bazaar by posting a big flyer that states the activity’s name and the venue and the date which is not really efficient and the problem is in the knowledge background of the community members as they have not yet utilized the event publishing functions on the social media platform due to very poor understanding to those event publishing functions to be viewed by the majority of people of all kinds of interests and backgrounds which leads to a limited amount of participants who knew about the event and in turn leads to a very weak cultural movement. Hence the (Design and implementation of online event Management system) is Proposed.

Fortunately, technological advancements have led to many wonderful things that can be utilized in the favour of Human Beings such as Mobile Applications, Web Applications and Desktop Applications which all can make human lives easier and to allow us to better utilize the opportunities that we have in front of us. The (Design and implementation of online event Management system) will be a system that will address the issues mentioned above and will allow users of all sorts of backgrounds and technical knowledge to create their own accounts on the system and gives them the capability to create events by specifying the name of the events, the venue of the event and the date of the event as well as the description of the event to let the users know what this event is all about. The system will categorize the events based on specific titles and topics such as (art, theatre, conferences, fashion, festivals, sports, etc…) to make it easier for the person browsing through the system to better find what they are looking for. The system shall also allow the users to book their attendance for a specific event and to have their names on the list of the event. Our system will have an e-wallet payment method to buy tickets to certain theatre shows and movies or pay a fee for an event in general. Our system shall be the rise of the cultural movement by letting the majority of people know about how alive this city is in terms of the events it holds and to attract most of the population to attend these activities and events and in the future this system can be utilized in the touristic movement by offering multiple language support for the tourists to increase the tourism traffic to our city.

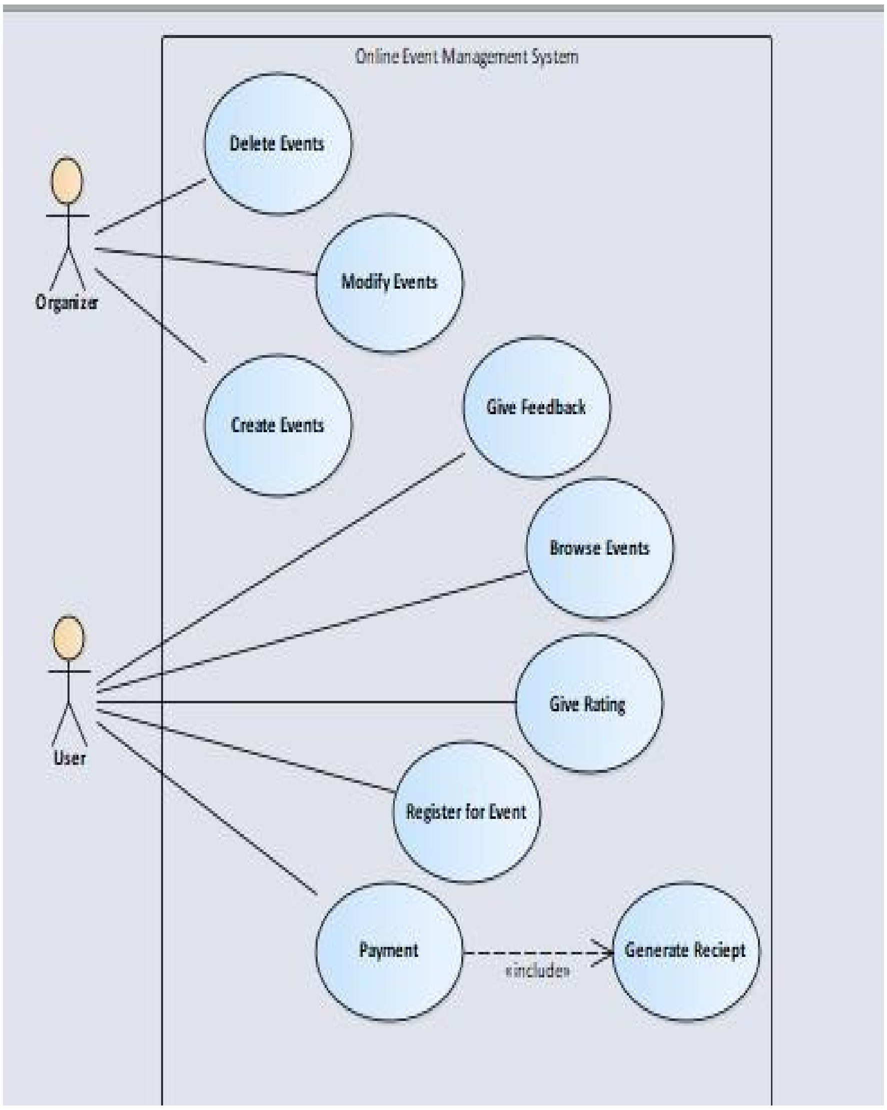


Figure 2.1: Use Case Diagram of <Design and Implementation of Online Event Management System>

2.1 Product Perspective

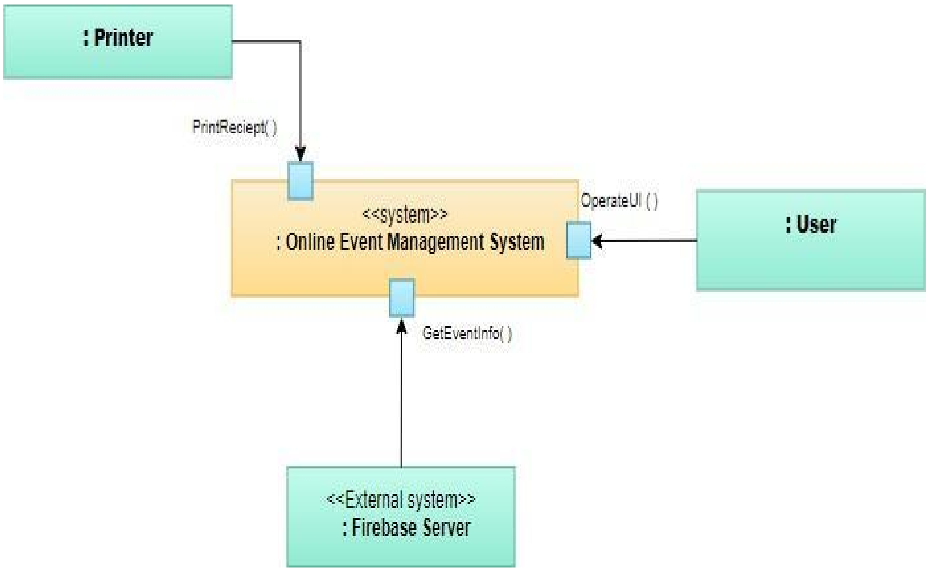


Figure 2.2: Block Diagram of <Design and Implementation of Online Event Management System>

2.1.1 System Interfaces

None

2.1.2 User Interfaces

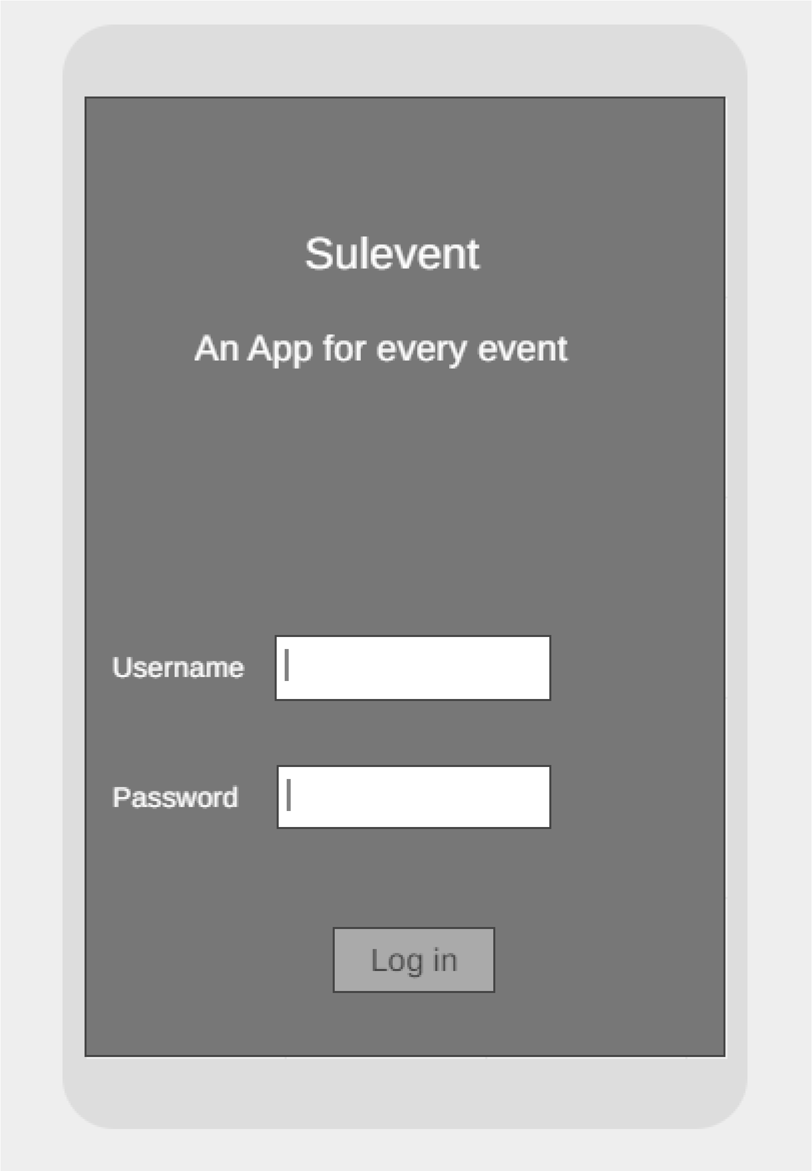


figure 2.3 : login structure of the system

This is the login page of the system in which the User logs in to the system by inserting username and password.

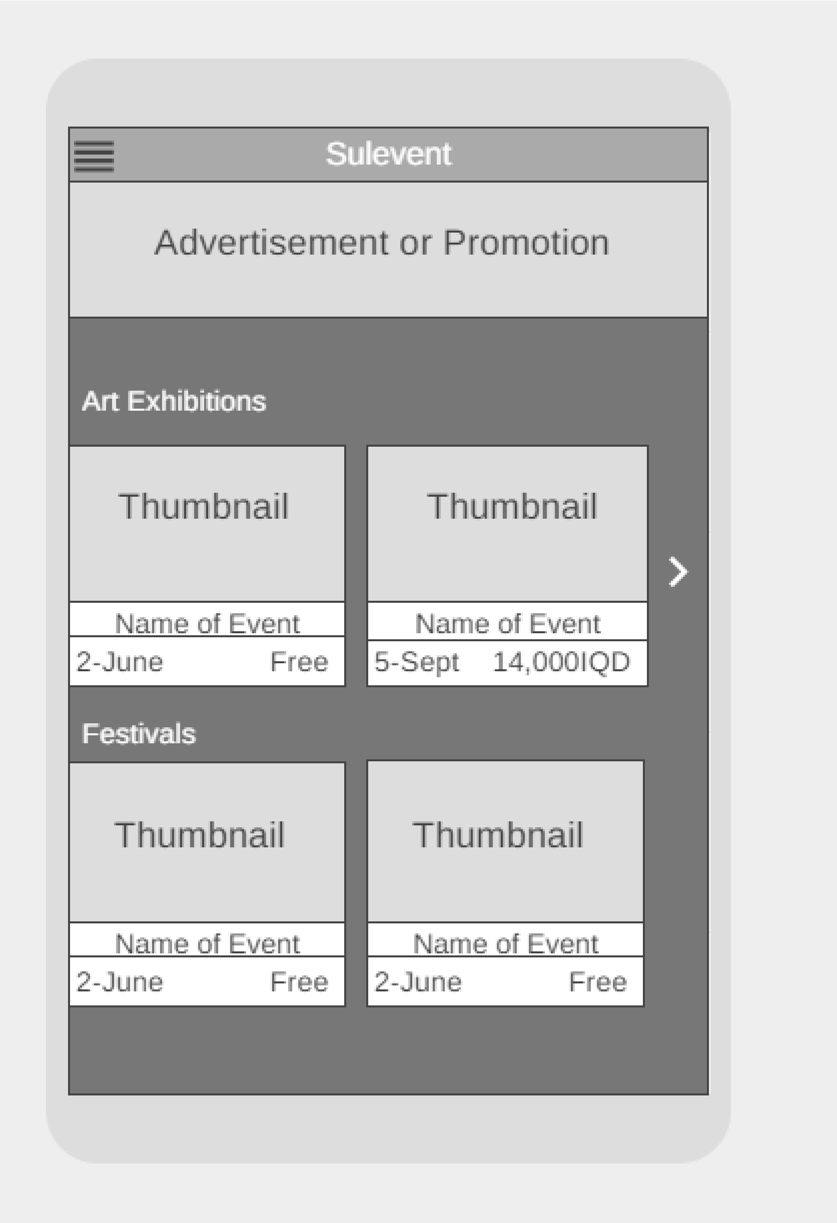


figure 2.4 : the main screen after login

This is the main screen of the system after the login by the user in which all the events of the system are shown in each category in which the user can browse; they can click on a specific category on this screen or the category tab can be used to view all the categories of the system.

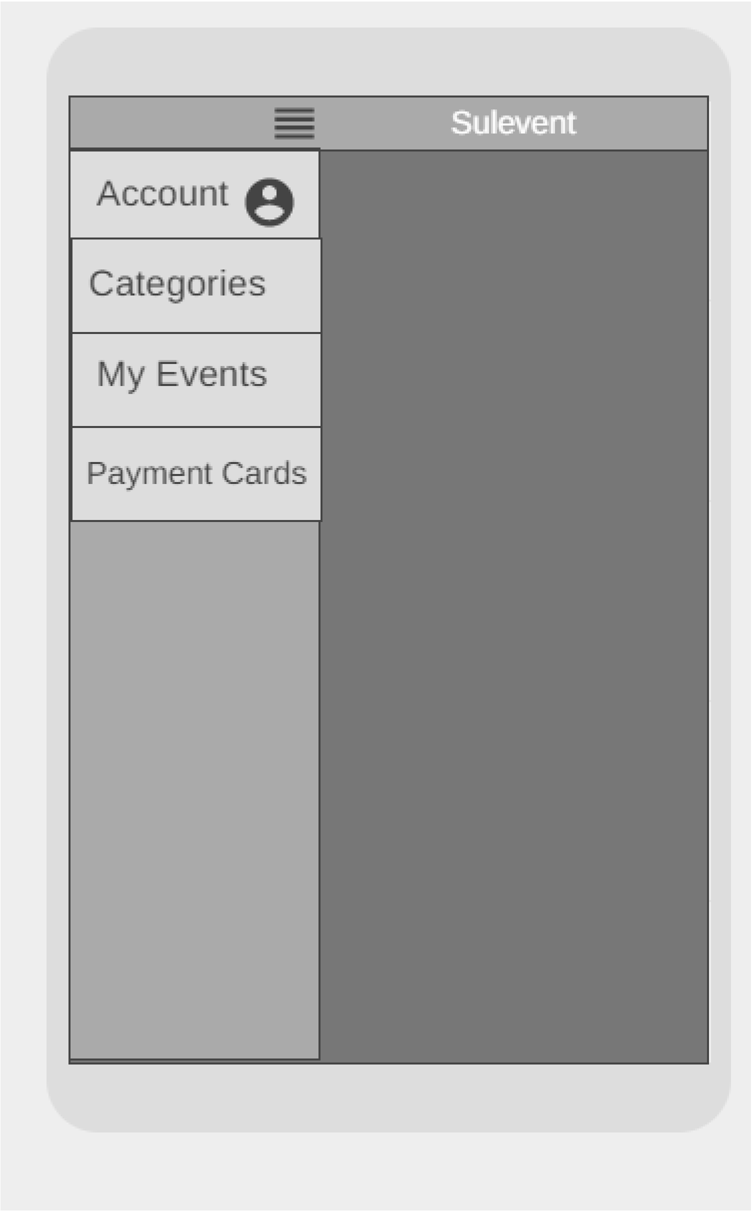
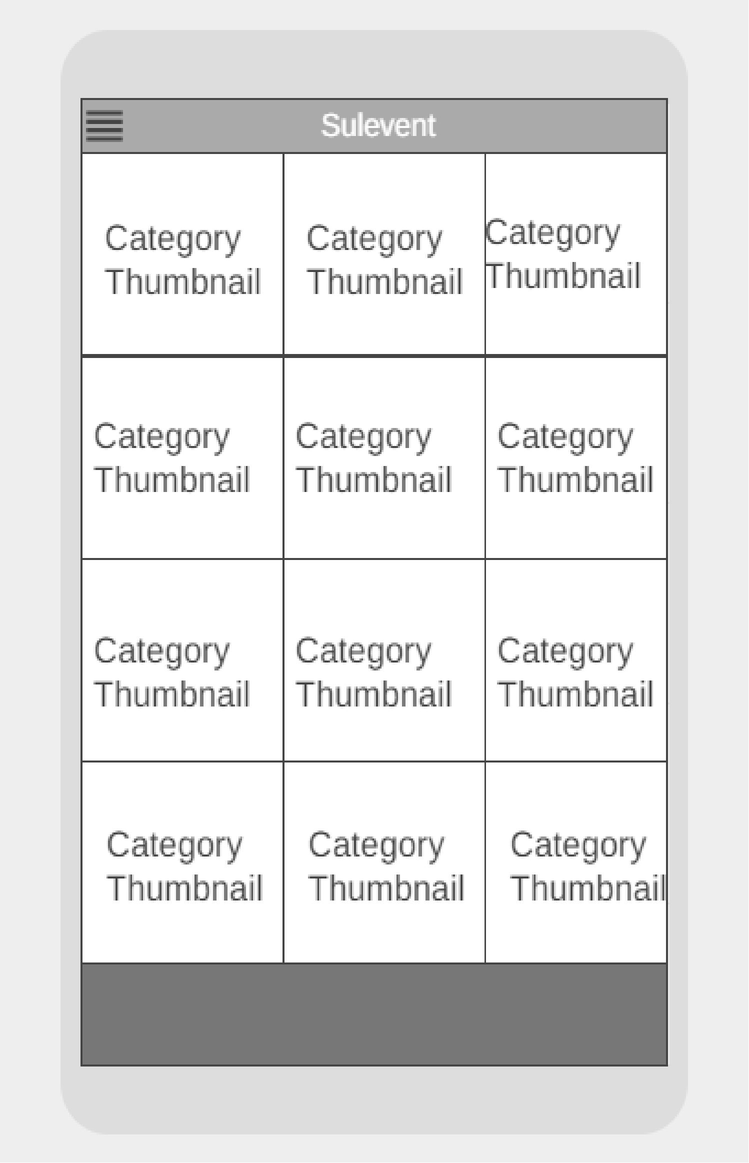


figure 2.5 : the slider that contains the options

this is the slider menu that contains the options in the system such as viewing the account of the user or viewing the categories by clicking on the category tab or show the events of the user by clicking on the my events tab , the user can also choose the payment cards and insert their information by clicking on the payment cards tab.

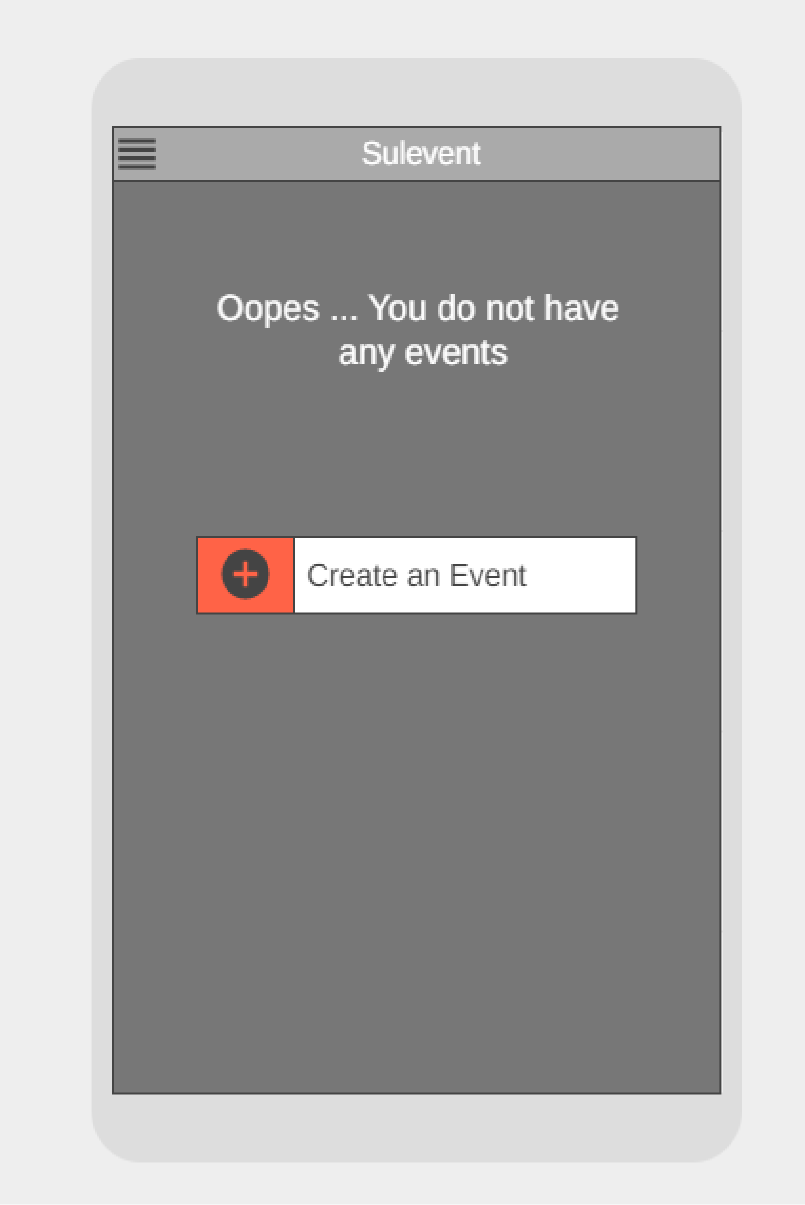


Sulevent

Category Thumbnail	Category Thumbnail	Category Thumbnail
Category Thumbnail	Category Thumbnail	Category Thumbnail
Category Thumbnail	Category Thumbnail	Category Thumbnail
Category Thumbnail	Category Thumbnail	Category Thumbnail

figure 2.6 : the categories

The user after clicking on the categories in the slider menu they are shown these tiles that contain a thumbnail of each category and once clicked it shows the events related to that certain category.



Oopes ... You do not have any events

figure 2.7 : create event

this menu option is shown after the my events tab has been clicked by the user of the system in which the events that the user has created are shown here but in this case there are no events for that user hence the message “Oops…You do not have any events” is shown along with the choice of creating an event in which the user is taken to the input form for the creation of the event once they click on the create an event button.

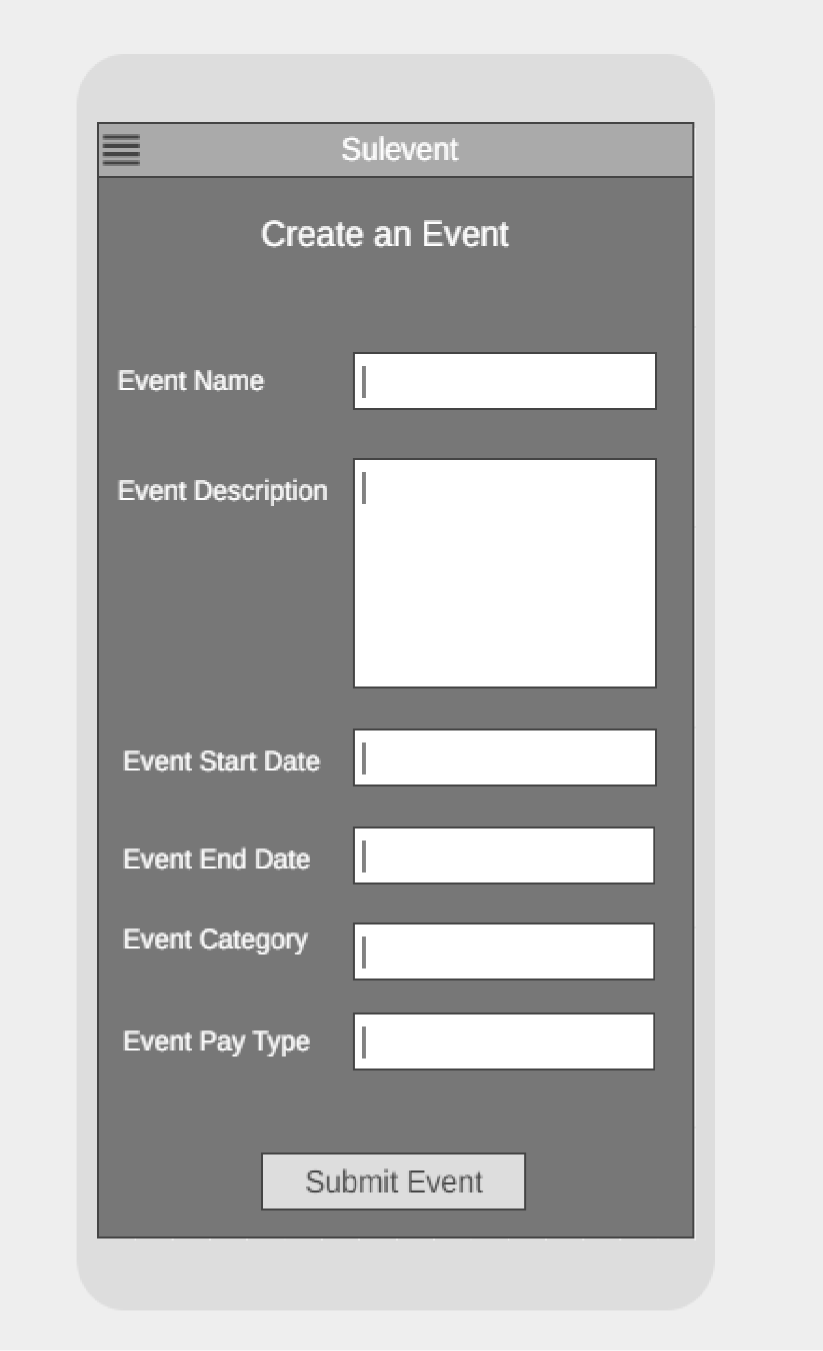


figure 2.8 : create event form

this is the input form for creating an event once the create an event button is clicked by the user it allows the user to input the name of the event , the description for the event , the event's start date , the event's end date , the event's category and the event pay type whether free or paid event. Then the submit event button is clicked in order to create the event.

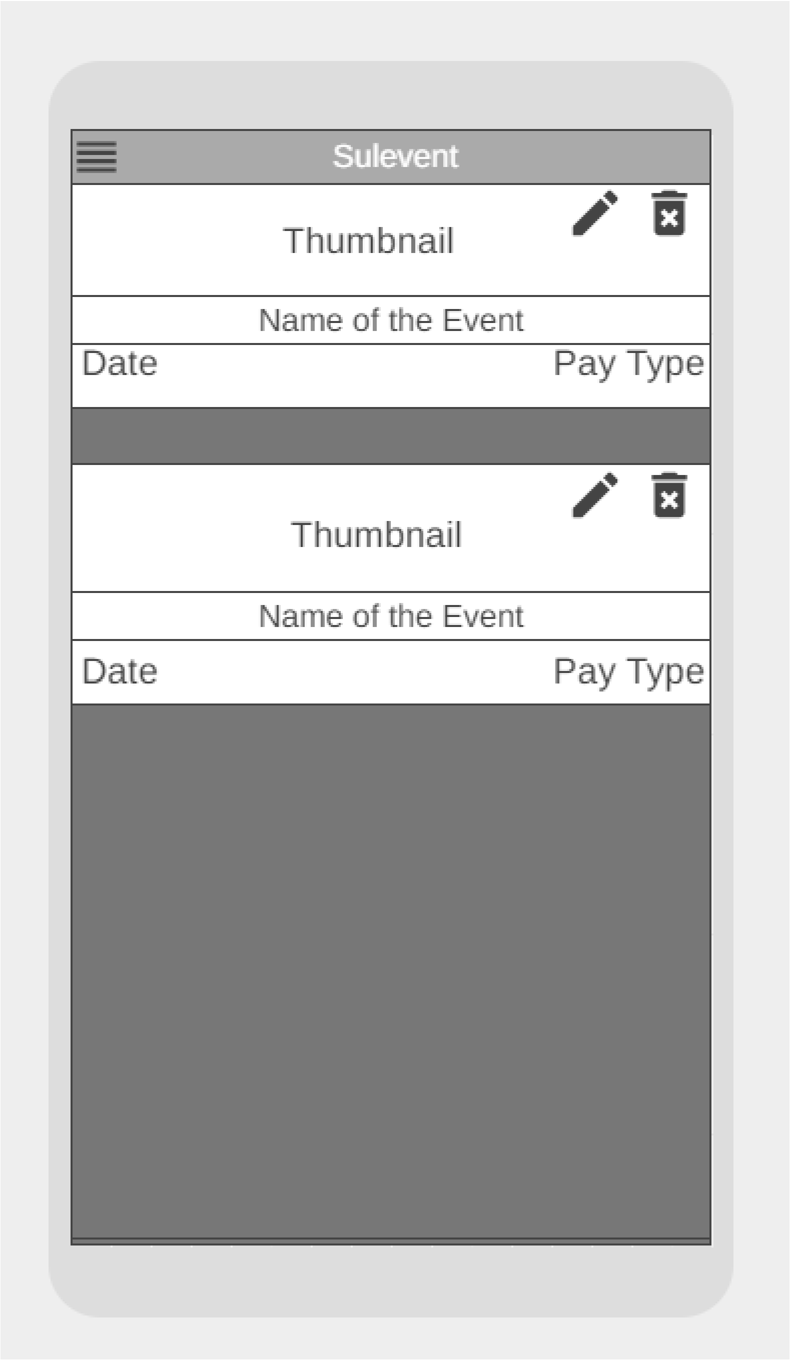
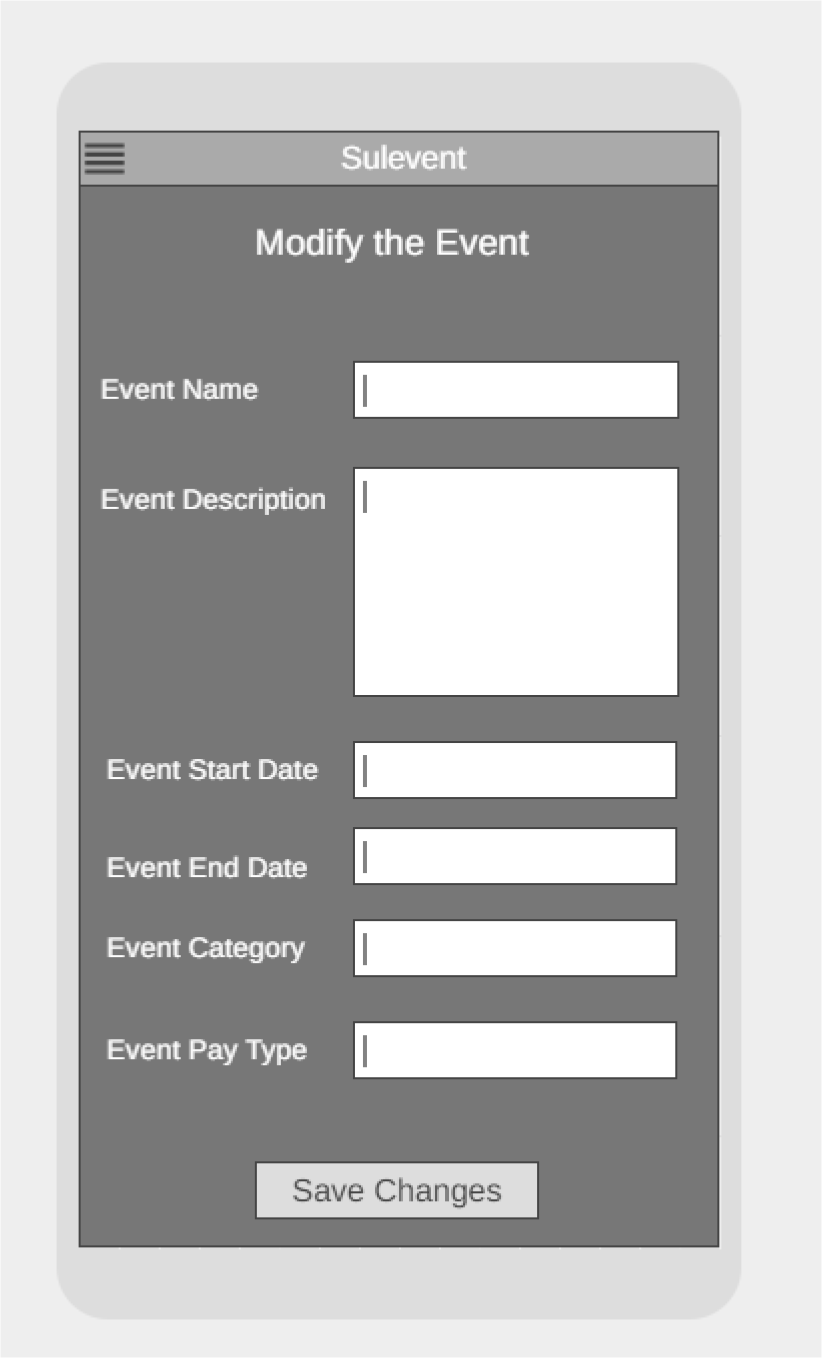


figure 2.9 : my events

This screen is shown once the user clicks on the my events tab in the slider menu in which the events of the user are shown and they are given the ability to either delete the event or to modify the existing information for that specific event.



Sulevent

Modify the Event

Event Name

Event Description

Event Start Date

Event End Date

Event Category

Event Pay Type

figure 2.10 : modify events

this screen is shown when the user clicks on the modification button in the my events tab to edit the information of the event that has been created, the user can then modify the event name , event description , the event start date , the event end date and the event category and the event pay type whether free or paid.

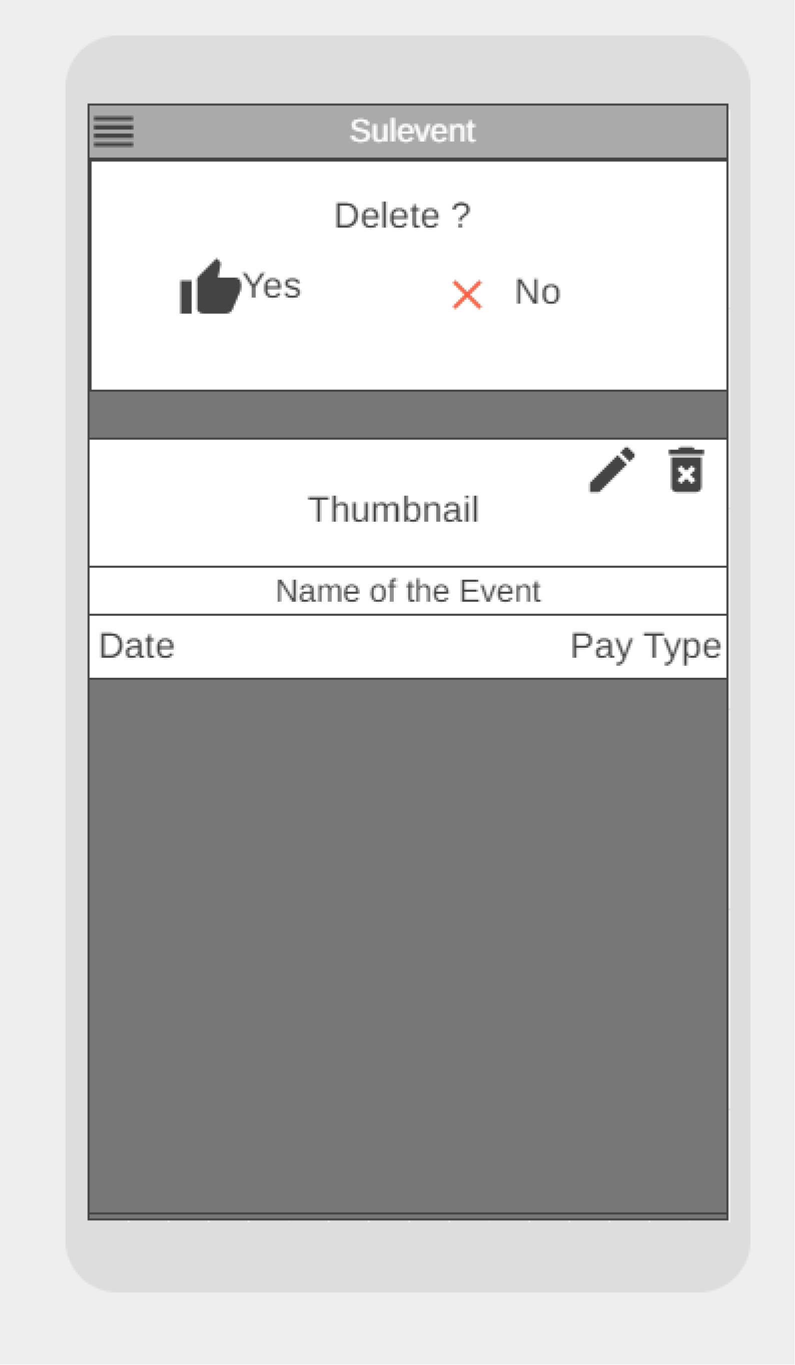


figure 2.11 : confirm delete event message

This message is shown when the delete icon is clicked and two choices are given to the user to either say no to refuse the deletion of the event or to say yes to confirm the deletion of the event.

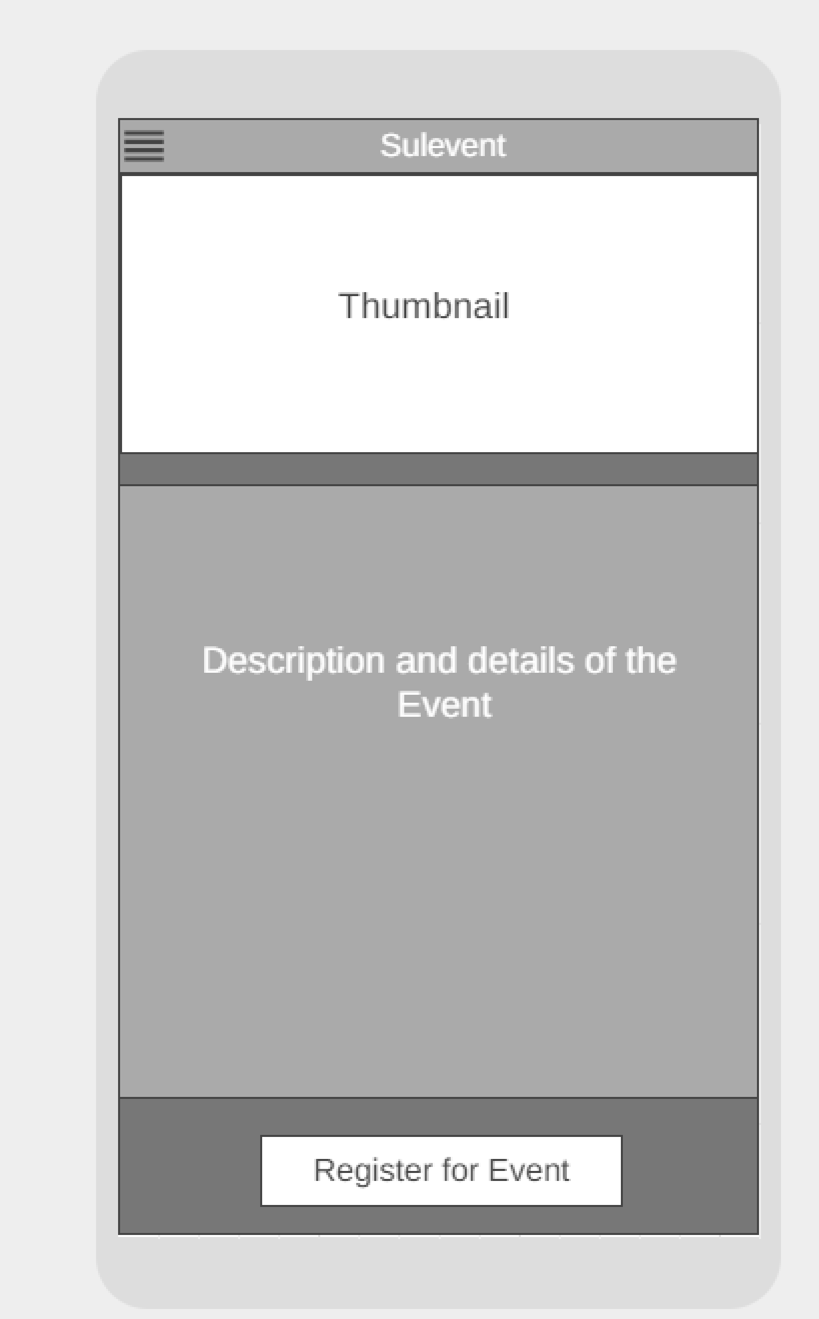


figure 2.12 : register for event

this screen is shown when the event is clicked to show the full details of the event as well as giving the user the ability to register for the event by clicking on the register for event button in which the user will be registered if the pay type of the event is free.

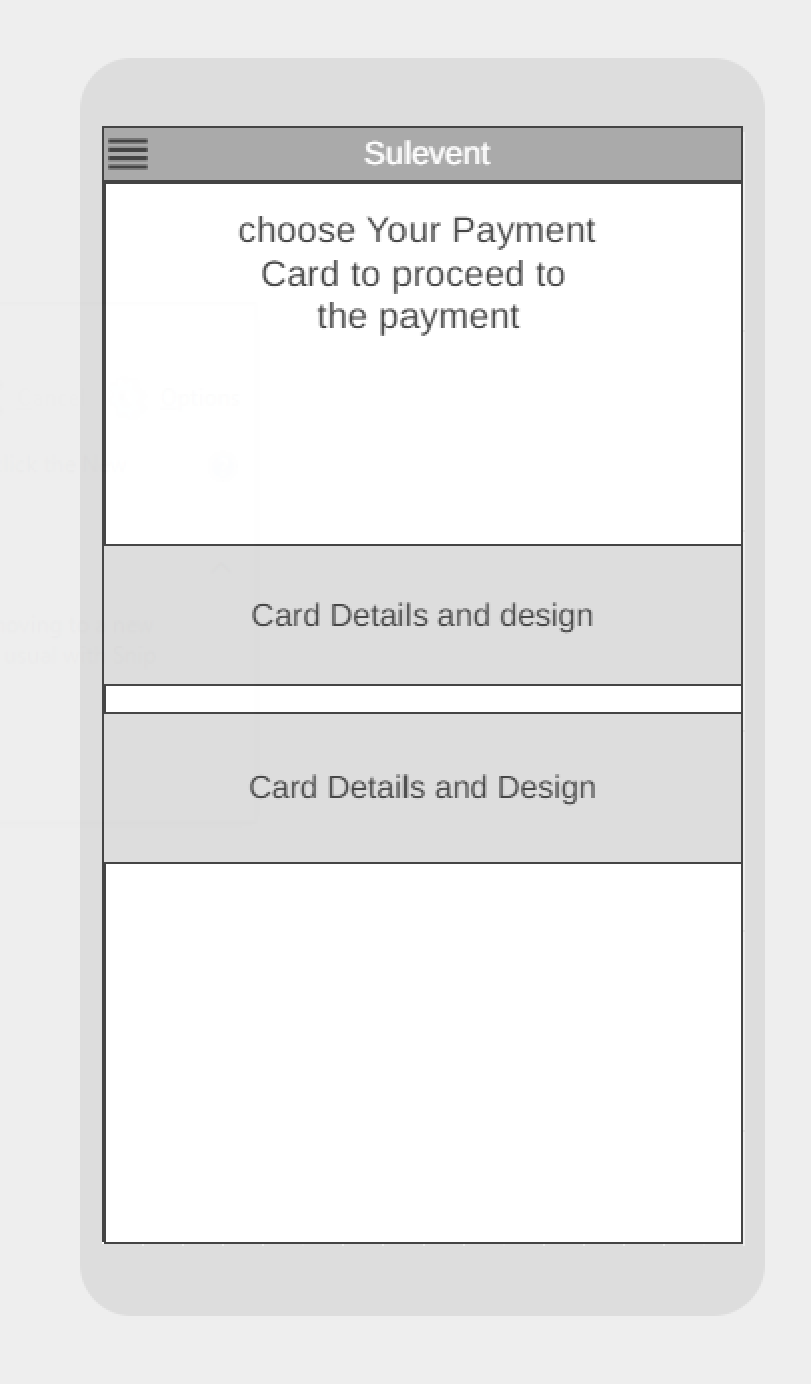


figure 2.13 : payment cards to choose from

This screen is shown to the user when the event pay type is not free and the user has clicked on register for the event , the choice is given to the user to choose their payment card to pay for the event.

2.1.3 Hardware Interfaces

None

2.1.4 Software Interfaces

Integration with social media platforms such as Facebook and Instagram as well as integration with e-payment applications through the use of API's

2.1.5 Communication Interfaces

None

2.1.6 Memory

2 GB of Random Access Memory (RAM) and 64 GB of Hard Disk Drive (HDD) memory are needed.

2.1.7 Operations

Organizers will be able to **(Create event)** in which the necessary details for a specific event such as (event name , event description , event start date , event end date , event category , event pay type) are inserted and click on submit to create an event.

Organizer will be able to **(modify event)** in which the necessary details for a specific event such as (event name , event description , event start date , event end date , event category , event pay type) are modified and click on modify event to modify the event.

Organizers will be able to **(Delete event)** by first fetching all the events that the Organizer has created and displaying a bin icon on top of each event in order for the Organizer to click on it and delete the event , a delete confirmation message is shown to the Organizer to click on before deleting the event permanently.

each user will be able to **(Register for an event)** by clicking on the display event information button and then clicking on register for an event button in the bottom of the event information if the (event type == free) then the registration wil automatically happen and the user shall be registered however if not then a choose payment method screen is shown to the user and a payment method is chosen and the event is registered.

each user will be able to **(browse event)** based on categories that the user wishes to search for such as (cinema , festivals , food , etc..) and view all the events in general that are currently present in the system.

each user will be able to do **(Payment)** for a paid event by choosing a payment method from the choose payment screen that is shown to the users to make payment for that specific event.

each user will be get a receipt of purchase through **(Generate Reciept)** in which the users will receive a receipt of purchase through their email addresses for the event which they have purchased tickets for.

each user will be able to **(Give Rating)** for which they can leave ratings for the events which they have participated in based on how satisfied they were with the events.

each user will be able to **(Give Feedback)** in which they can leave comments and feedbacks for the event they have participated in and express their opinions.

2.1.8 Site Adaptation Requirements

None

2.2 Product Functions

refer to Figure 2.1 Use case diagram of the <Online Event Management System>

Organizers will be able to **(Create event)** in which the necessary details for a specific event such as (event name , event description , event start date , event end date , event category , event pay type) are inserted and click on submit to create an event.

Organizer will be able to **(modify event)** in which the necessary details for a specific event such as (event name , event description , event start date , event end date , event category , event pay type) are modified and click on modify event to modify the event.

Organizers will be able to **(Delete event)** by first fetching all the events that the Organizer has created and displaying a bin icon on top of each event in order for the Organizer to click on it and delete the event , a delete confirmation message is shown to the Organizer to click on before deleting the event permanently.

each user will be able to **(Register for an event)** by clicking on the display event information button and then clicking on register for an event button in the bottom of the event information if the (event type == free) then the registration wil automatically happen and the user shall be registered however if not then a choose payment method screen is shown to the user and a payment method is chosen and the event is registered.

each user will be able to **(browse event)** based on categories that the user wishes to search for such as (cinema , festivals , food , etc..) and view all the events in general that are currently present in the system.

each user will be able to do **(Payment)** for a paid event by choosing a payment method from the choose payment screen that is shown to the users to make payment for that specific event.

each user will be get a receipt of purchase through **(Generate Reciept)** in which the users will receive a receipt of purchase through their email addresses for the event which they have purchased tickets for.

each user will be able to **(Give Rating)** for which they can leave ratings for the events which they have participated in based on how satisfied they were with the events.

each user will be able to **(Give Feedback)** in which they can leave comments and feedbacks for the event they have participated in and express their opinions.

2.3 User Characteristics

The system is intended for Users from various technological backgrounds , to be used by the most basic user by providing an intuitive interface that is easy to understand and straightforward, that does not require long periods of time to train on in order to understand.The young population is the main focus of the system but this system is also intended for all types of users.

2.4 Constraints

Minimum requirements

a) Processor : Quad-core 2.4 Ghz
b) Chipset : Exynos 1280
c) Ram : 2 GB
d) Storage : 64 GB
e) Network : LTE

Recommended requirements

f) Processor : Octa-core 2.4 Ghz
g) Chipset : Qualcomm SM7325 Snapdragon 778G 5G
h) Ram : 4 GB
i) Storage : 128 GB
j) Network : 5G

2.5 Assumption and Dependencies

The system shall be developed for Mobile systems and it shall be cross-platform and will developed to be used on both (ios) and (Android) Mobile operating systems.If any change happens in the implementation of the system or by changing the intended device and platform the SRS shall be changed accordingly.

2.6 Apportioning of Requirements

The (Give Rating) and (Give Feedback) use cases shall be delayed until the future versions of the system and shall not be implemented in the current version of the system.

3. Specific Requirements

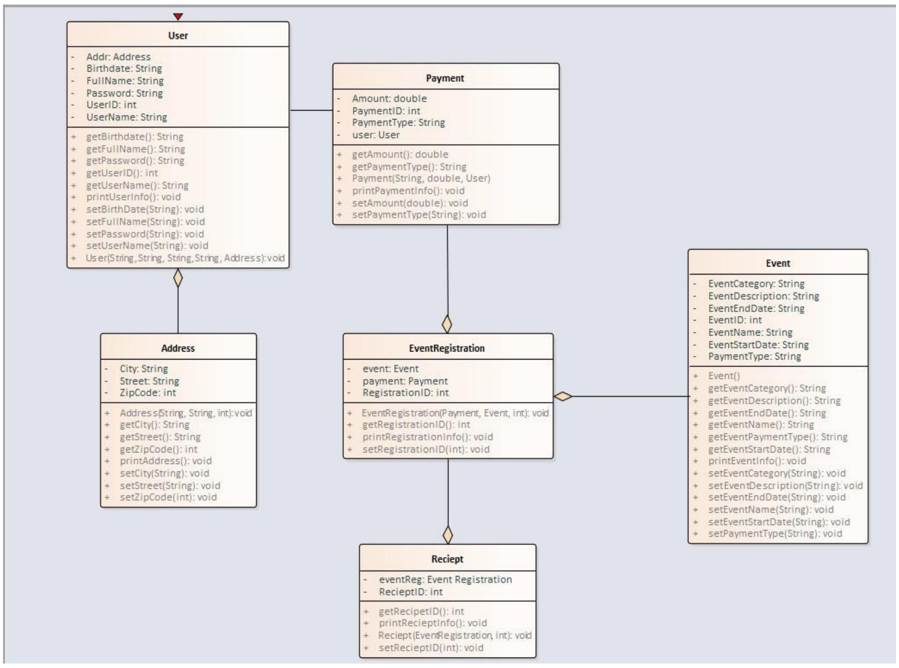

User
- Addr: Address
- Birthdate: String
- FullName: String
- Password: String
- UserID: int
- UserName: String
+ getBirthdate(): String
+ getFullName(): String
+ getPassword(): String
+ getUserID(): int
+ getUserName(): String
+ printUserInfo(): void
+ setBirthDate(String): void
+ setFullName(String): void
+ setPassword(String): void
+ setUserName(String): void
+ User(String,String, String,String, Address):void
Payment
- Amount: double
- PaymentID: int
- PaymentType: String
- user: User
+ getAmount(): double
+ getPaymentType(): String
+ Payment(String, double, User)
+ printPaymentInfo(): void
+ setAmount(double): void
+ setPaymentType(String): void
Event
- EventCategory: String
- EventDescription: String
- EventEndDate: String
- EventID: int
- EventName: String
- EventStartDate: String
- PaymentType: String
+ Event()
+ getEventCategory(): String
+ getEventDescription(): String
+ getEventEndDate(): String
+ getEventName(): String
+ getEventPaymentType(): String
+ getEventStartDate(): String
+ printEventInfo(): void
+ setEventCategory(String): void
+ setEventDescription(String): void
+ setEventEndDate(String): void
+ setEventName(String): void
+ setEventStartDate(String): void
+ setPaymentType(String): void
Address
- City: String
- Street: String
- ZipCode: int
+ Address(String, String, int):void
+ getCity(): String
+ getStreet(): String
+ getZipCode(): int
+ printAddress(): void
+ setCity(String): void
+ setStreet(String): void
+ setZipCode(int): void
EventRegistration
- event: Event
- payment: Payment
- RegistrationID: int
+ EventRegistration(Payment, Event, int): void
+ getRegistrationID(): int
+ printRegistrationInfo(): void
+ setRegistrationID(int): void
Reciept
- eventReg: Event Registration
- RecieptID: int
+ getRecipetID(): int
+ printRecieptInfo(): void
+ Reciept(EventRegistration, int): void
+ setRecieptID(int): void

Figure 3.1: Domain Model of <Online Event Management System>

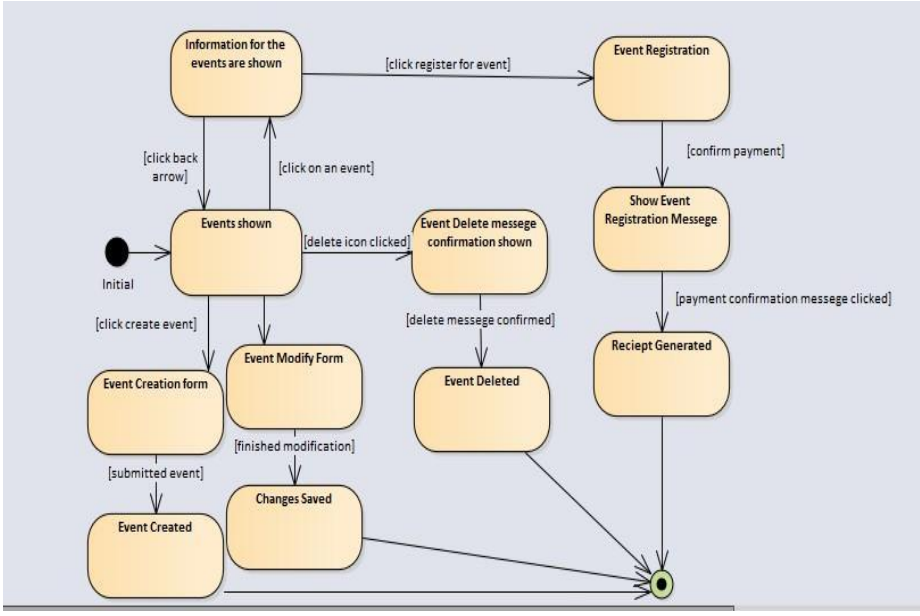


Figure 3.2: State Machine Diagram of <Online Event Management System>

3.1 External Interface Requirements

3.1.1 User Interfaces

Figure 2.3 This is the login page of the system in which the User logs in to the system by inserting username and password.

Figure 2.4 This is the main screen of the system after the login by the user in which all the events of the system are shown in each category in which the user can browse; they can click on a specific category on this screen or the category tab can be used to view all the categories of the system.

Figure 2.5 this is the slider menu that contains the options in the system such as viewing the account of the user or viewing the categories by clicking on the category tab or show the events of the user by clicking on the my events tab , the user can also choose the payment cards and insert their information by clicking on the payment cards tab.

Figure 2.6 The user after clicking on the categories in the slider menu they are shown these tiles that contain a thumbnail of each category and once clicked it shows the events related to that certain category.

Figure 2.7 this menu option is shown after the my events tab has been clicked by the Organizer in which the events that the Organizer has created are shown here but in this case there are no events for that user hence the message "Oops…You do not have any events" is shown along with the choice of creating an event in which the Organizer is taken to the input form for the creation of the event once they click on the create an event button.

Figure 2.8 this is the input form for creating an event once the create an event button is clicked by the Organizer it allows the Organizer to input the name of the event , the description for the event , the event's start date , the event's end date , the event's category and the event pay type whether free or paid event. Then the submit event button is clicked in order to create the event.

Figure 2.9 This screen is shown once the Organizer clicks on the my events tab in the slider menu in which the events of the Organizer are shown and they are given the ability to either delete the event or to modify the existing information for that specific event.

Figure 2.10 this screen is shown when the Organizer clicks on the modification button in the my events tab to edit the information of the event that has been created, the Organizer can then modify the event name , event description , the event start date , the event end date and the event category and the event pay type whether free or paid.

Figure 2.11 This message is shown when the delete icon is clicked and two choices are given to the Organizer to either say no to refuse the deletion of the event or to say yes to confirm the deletion of the event.

Figure 2.12 this screen is shown when the event is clicked to show the full details of the event as well as giving the user the ability to register for the event by clicking on the register for event button in which the user will be registered if the pay type of the event is free , however if not free then the user is given the choice to choose their payment car

3.1.2 Hardware Interfaces

None

3.1.3 Software Interfaces

None

3.1.4 Communication Interfaces

None

3.2 System Features

3.2.1 Module <Online Event Management System>

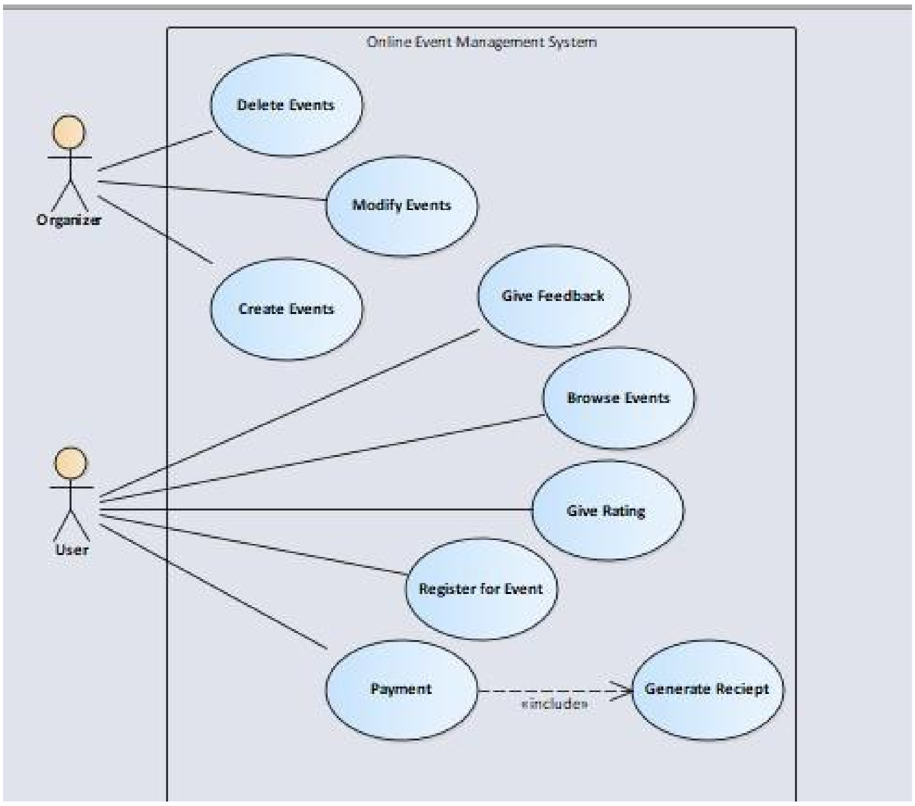
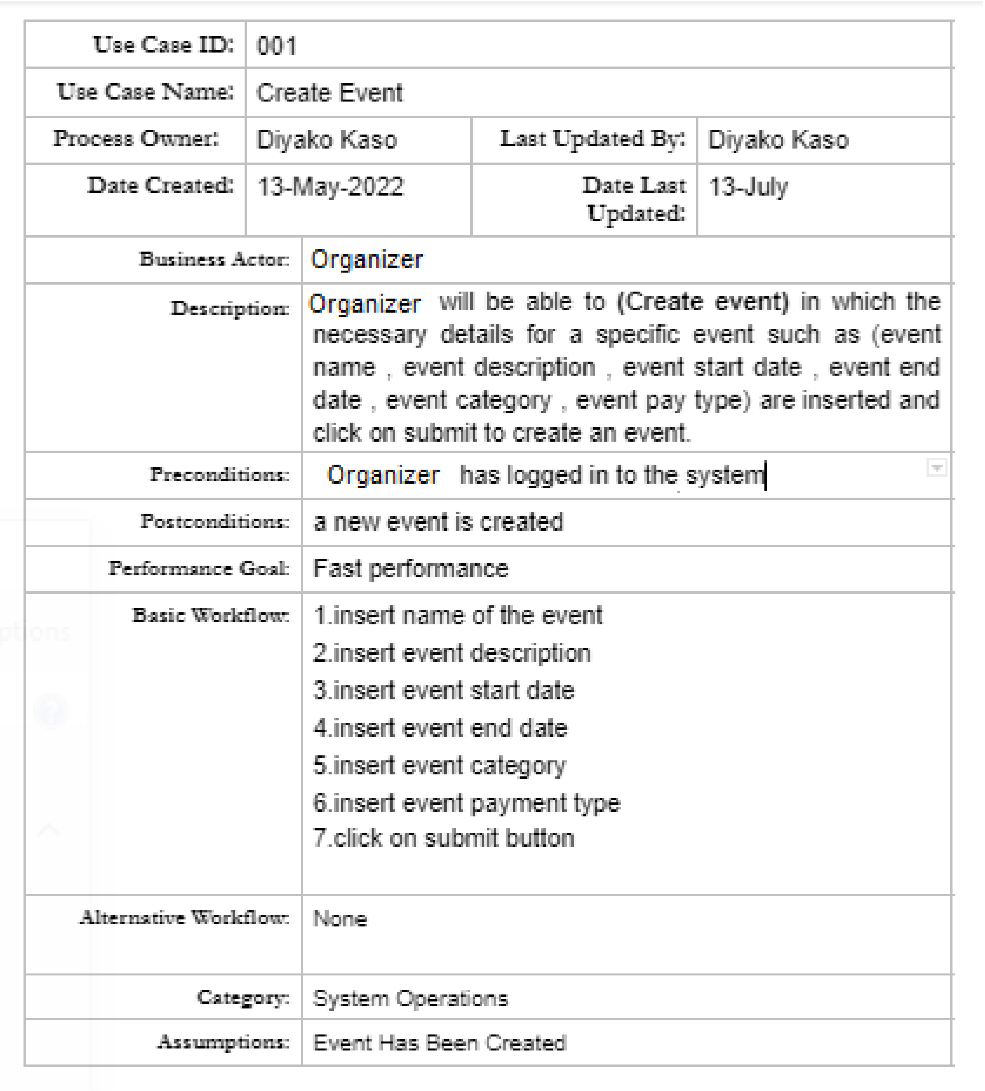


Figure 3.2: <Online Event Management System>

3.2.1.1 UC001: Use Case <Create Event>



Use Case ID:	001		
Use Case Name:	Create Event		
Process Owner:	Diyako Kaso	Last Updated By:	Diyako Kaso
Date Created:	13-May-2022	Date Last Updated:	13-July
Business Actor:	Organizer		
Description:	Organizer will be able to (Create event) in which the necessary details for a specific event such as (event name , event description , event start date , event end date , event category , event pay type) are inserted and click on submit to create an event.		
Preconditions:	Organizer has logged in to the system		
Postconditions:	a new event is created		
Performance Goal:	Fast performance		
Basic Workflow:	1.insert name of the event 2.insert event description 3.insert event start date 4.insert event end date 5.insert event category 6.insert event payment type 7.click on submit button		
Alternative Workflow:	None		
Category:	System Operations		
Assumptions:	Event Has Been Created		

Table 3.1: Use Case Description for <Create Event>

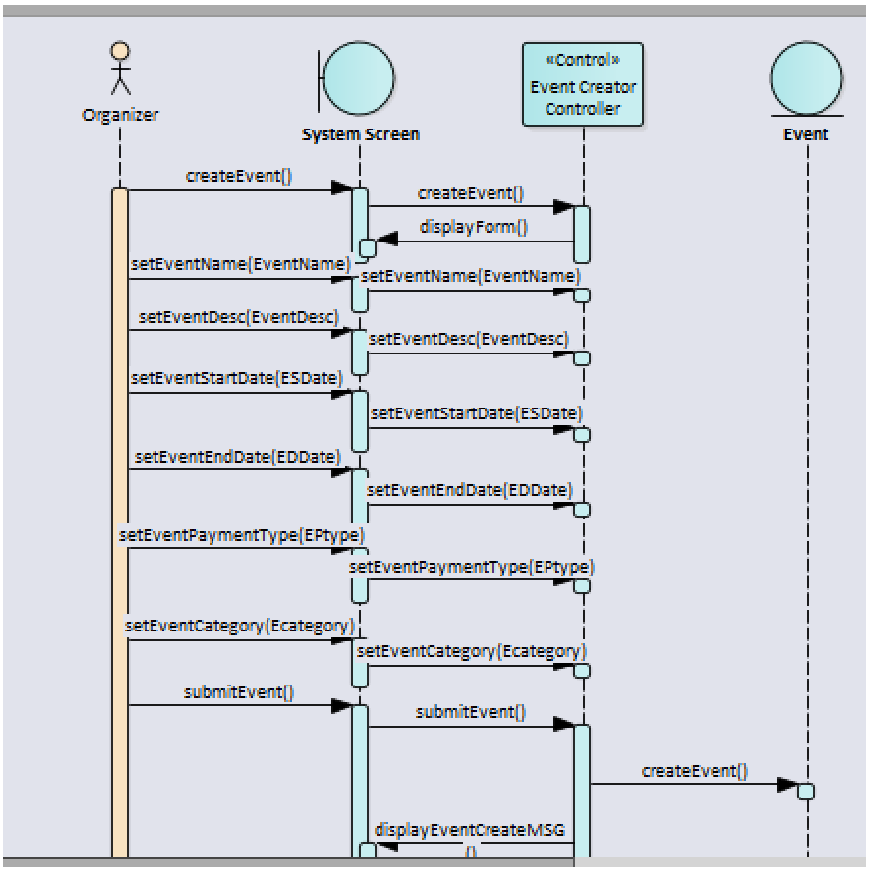


Figure 3.3: System Sequence Diagram of <Create Event>

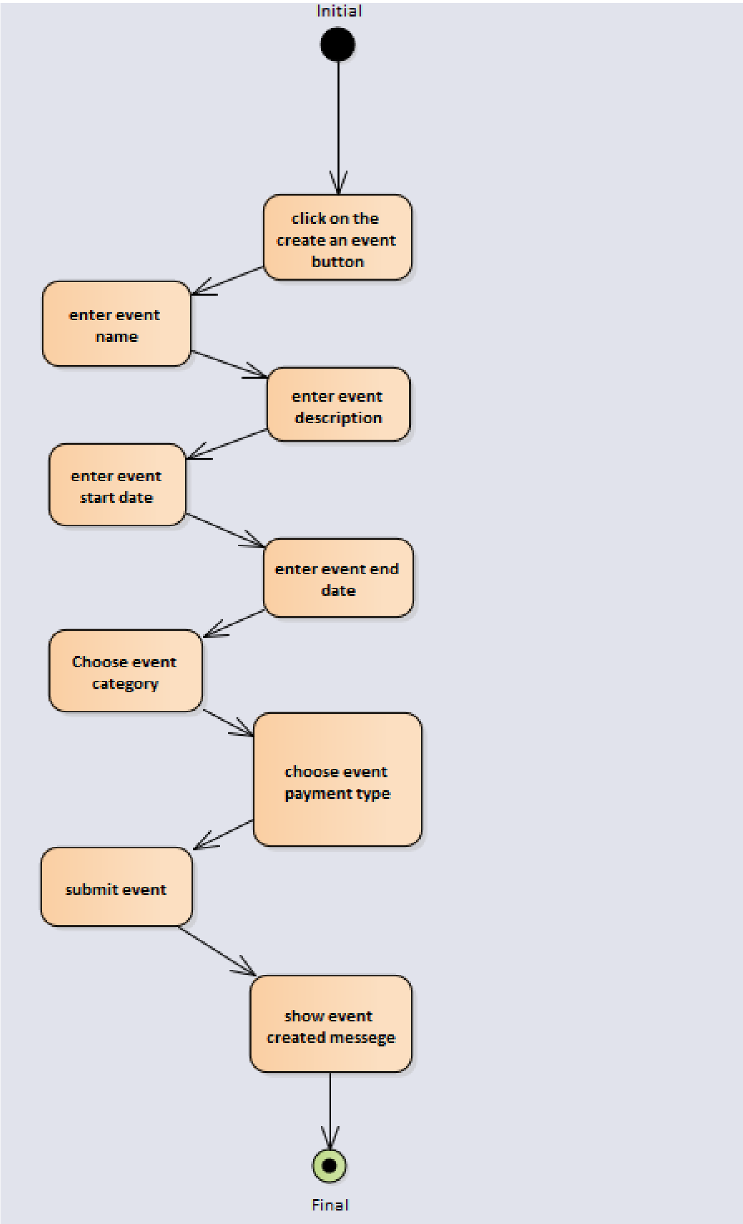
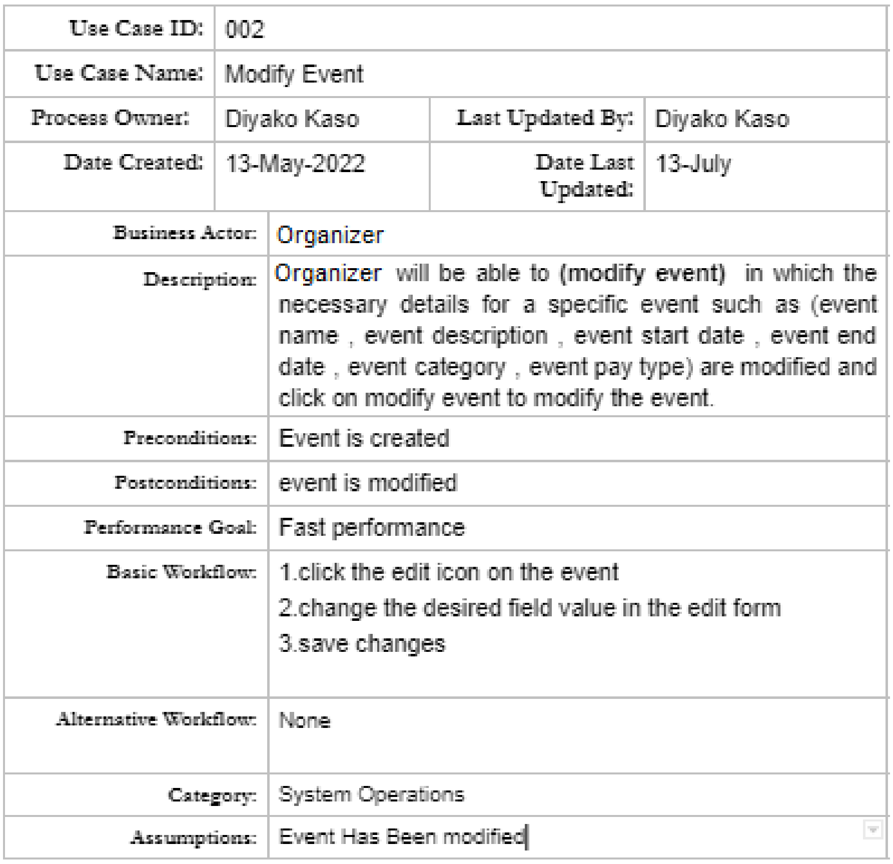


Figure 3.4: Activity Diagram of <Create Event>

3.2.1.2 UC002: Use Case <Modify Event>



Use Case ID:	002		
Use Case Name:	Modify Event		
Process Owner:	Diyako Kaso	Last Updated By:	Diyako Kaso
Date Created:	13-May-2022	Date Last Updated:	13-July
Business Actor:	Organizer		
Description:	Organizer will be able to **(modify event)** in which the necessary details for a specific event such as (event name , event description , event start date , event end date , event category , event pay type) are modified and click on modify event to modify the event.		
Preconditions:	Event is created		
Postconditions:	event is modified		
Performance Goal:	Fast performance		
Basic Workflow:	1.click the edit icon on the event 2.change the desired field value in the edit form 3.save changes		
Alternative Workflow:	None		
Category:	System Operations		
Assumptions:	Event Has Been modified		

Table 3.2: Use Case Description for <Modify Event>

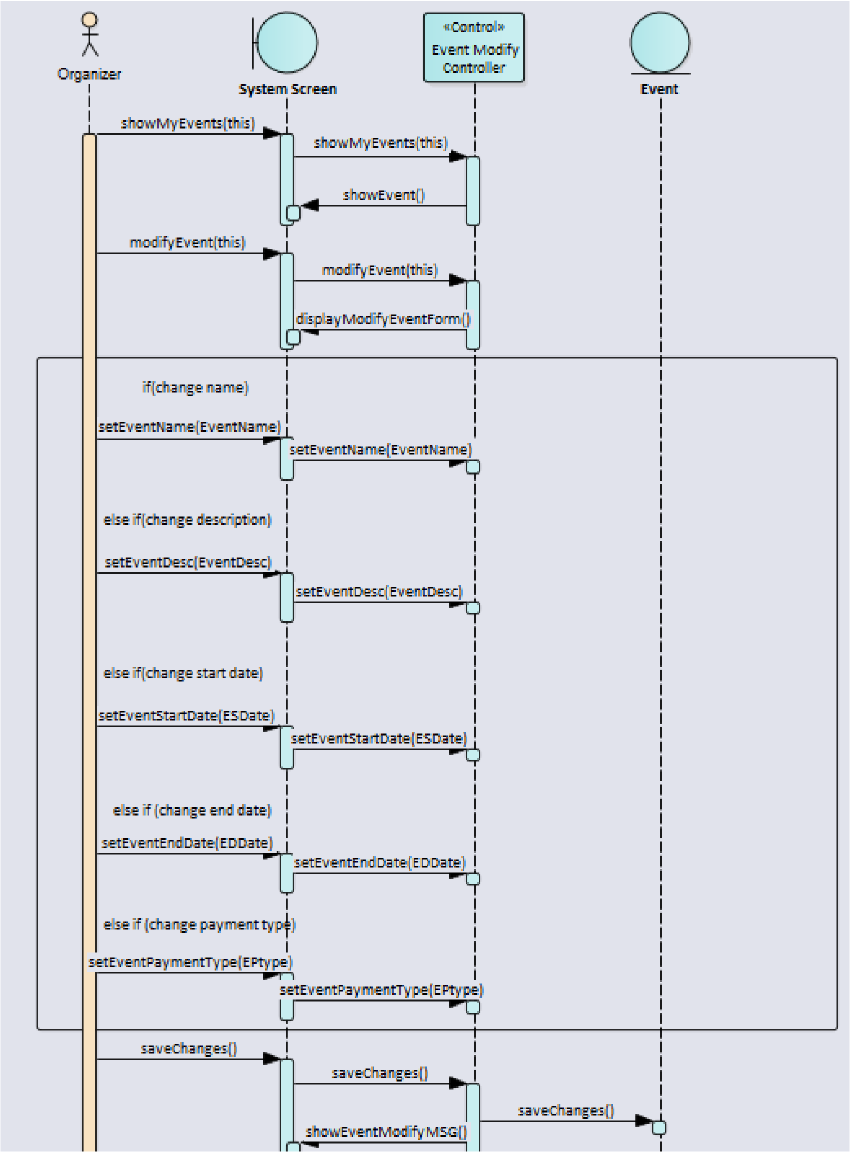


Figure 3.5: System Sequence Diagram of <Modify Event>

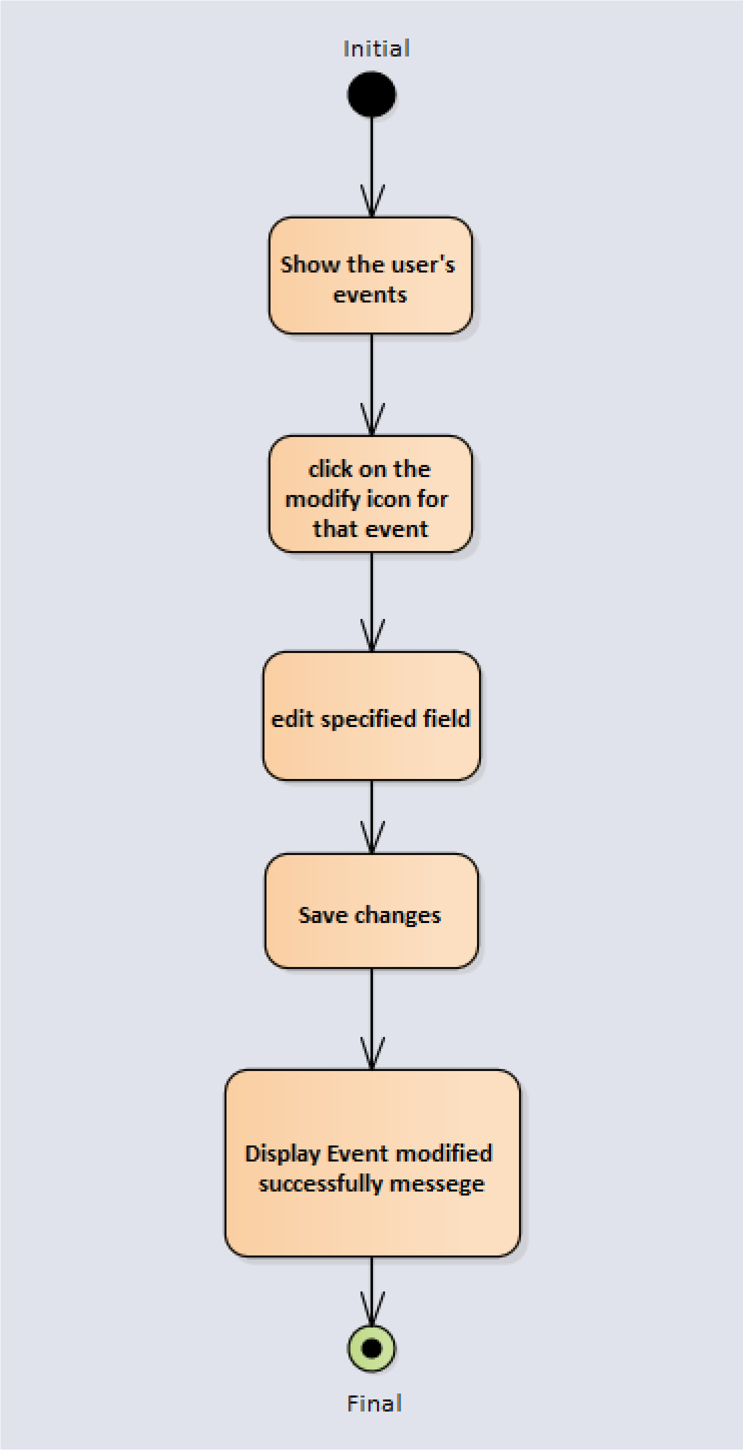
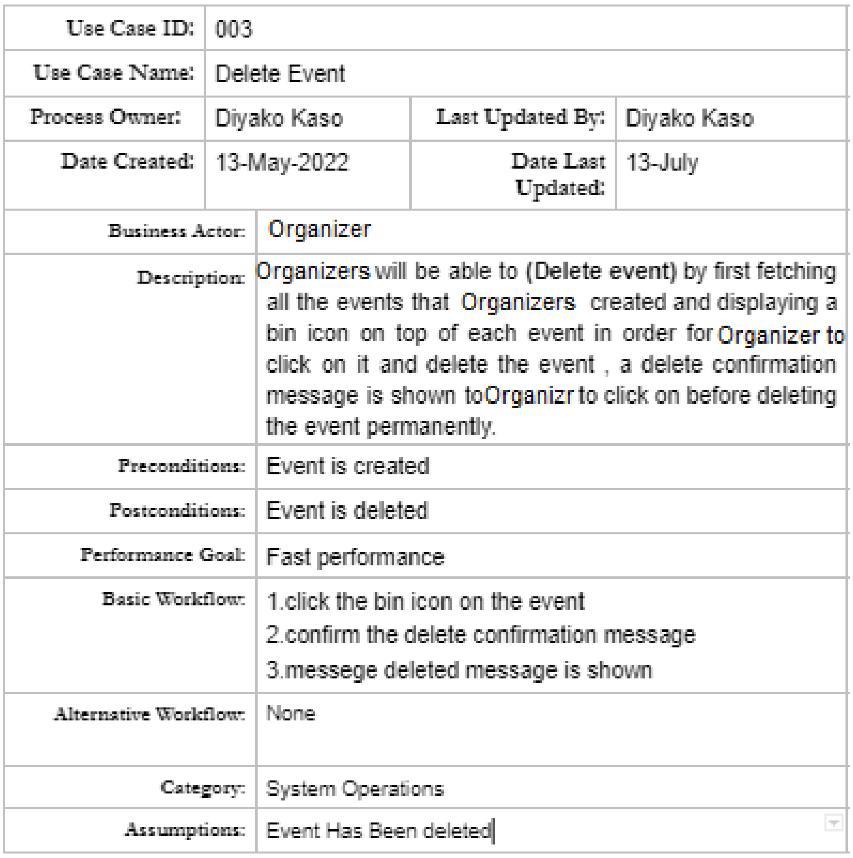

Initial
Show the user's events
click on the modify icon for that event
edit specified field
Save changes
Display Event modified successfully messege
Final

Figure 3.6: Activity Diagram of <Modify Event>

3.2.1.3 UC003: Use Case <Delete Event>



Use Case ID:	003		
Use Case Name:	Delete Event		
Process Owner:	Diyako Kaso	Last Updated By:	Diyako Kaso
Date Created:	13-May-2022	Date Last Updated:	13-July
Business Actor:	Organizer		
Description:	Organizers will be able to (Delete event) by first fetching all the events that Organizers created and displaying a bin icon on top of each event in order for Organizer to click on it and delete the event , a delete confirmation message is shown toOrganizr to click on before deleting the event permanently.		
Preconditions:	Event is created		
Postconditions:	Event is deleted		
Performance Goal:	Fast performance		
Basic Workflow:	1.click the bin icon on the event 2.confirm the delete confirmation message 3.messege deleted message is shown		
Alternative Workflow:	None		
Category:	System Operations		
Assumptions:	Event Has Been deleted		

Table 3.3: Use Case Description for <Delete Event>

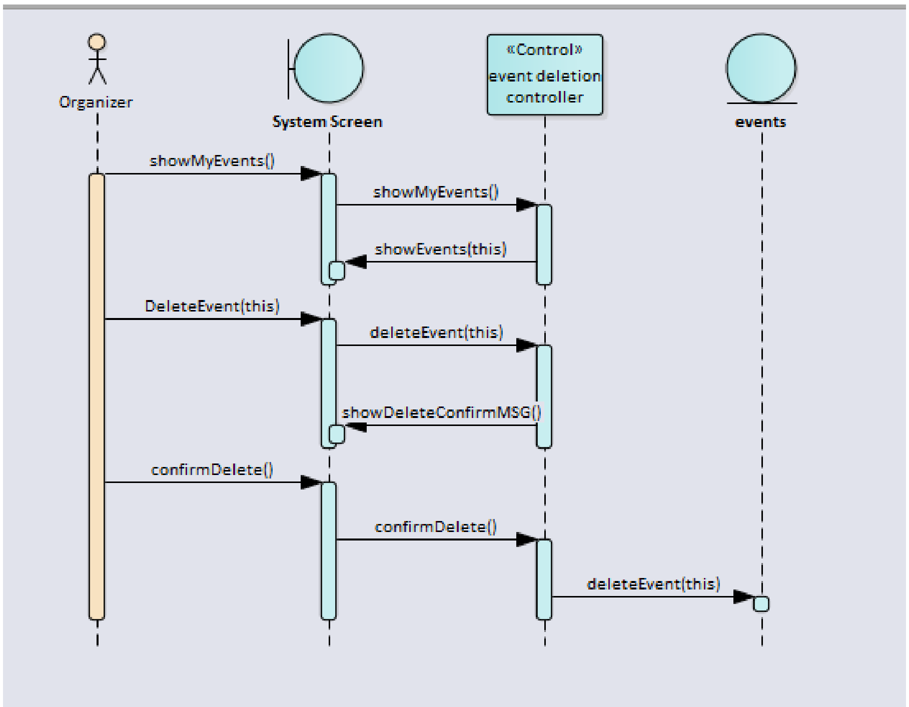


Figure 3.7: System Sequence Diagram of <Delete Event>

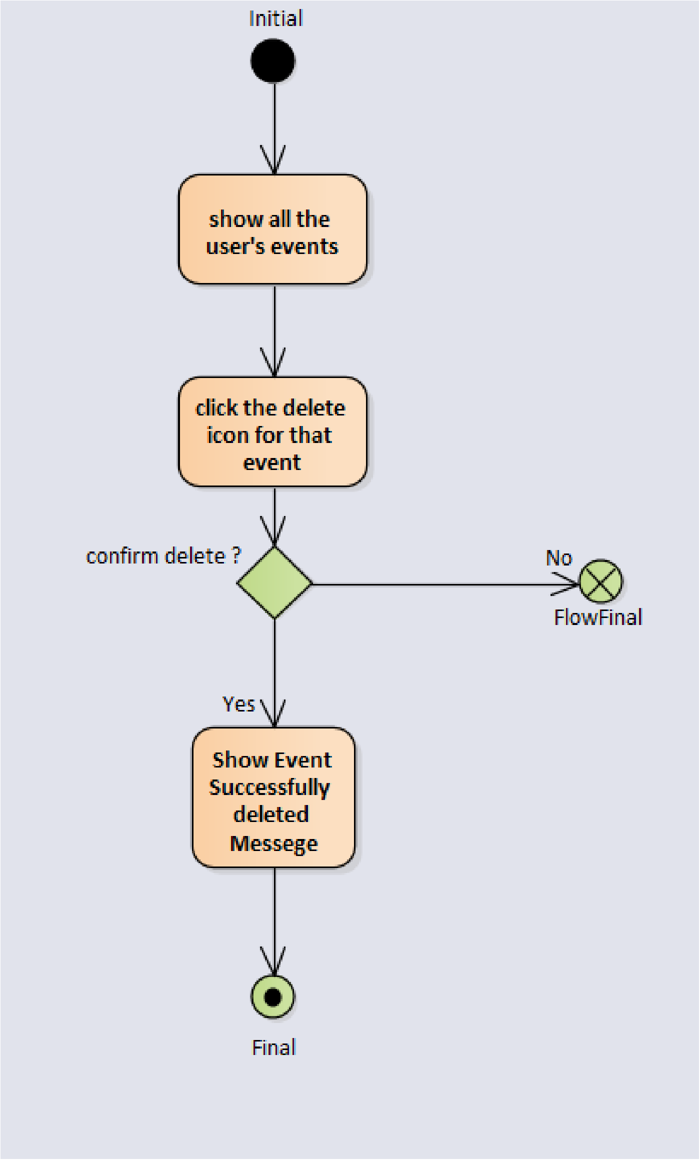
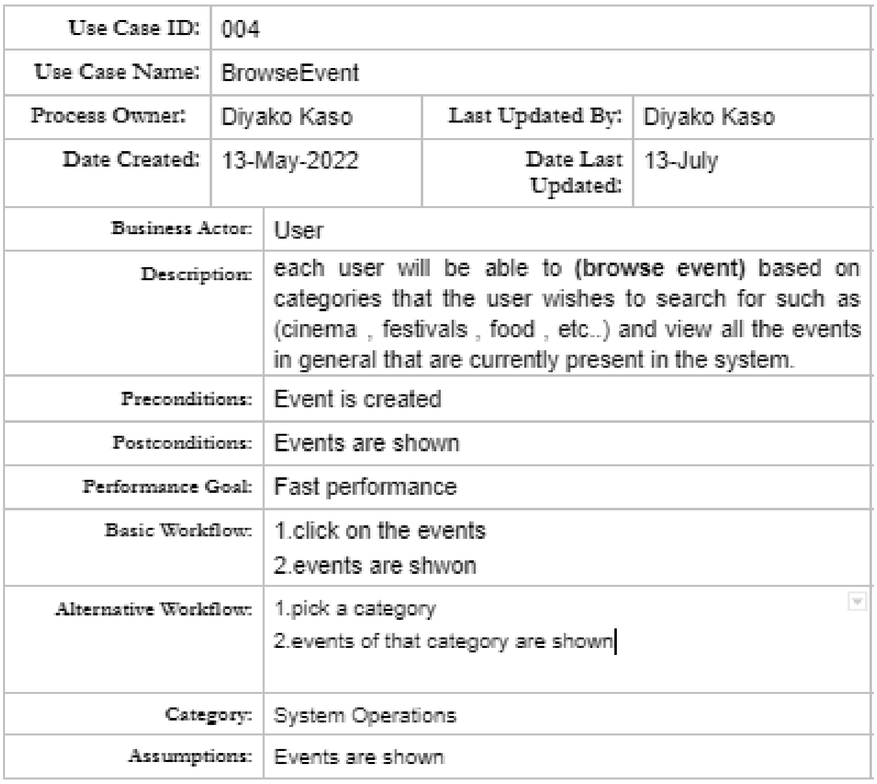

Initial
show all the user's events
click the delete icon for that event
confirm delete ?
No
FlowFinal
Yes
Show Event Successfully deleted Messege
Final

Figure 3.8: Activity Diagram of <Delete Event>

3.2.1.4 UC004 : Use Case <Browse Event>



Use Case ID:	004		
Use Case Name:	BrowseEvent		
Process Owner:	Diyako Kaso	Last Updated By:	Diyako Kaso
Date Created:	13-May-2022	Date Last Updated:	13-July
Business Actor:	User		
Description:	each user will be able to **(browse event)** based on categories that the user wishes to search for such as (cinema , festivals , food , etc..) and view all the events in general that are currently present in the system.		
Preconditions:	Event is created		
Postconditions:	Events are shown		
Performance Goal:	Fast performance		
Basic Workflow:	1.click on the events 2.events are shwon		
Alternative Workflow:	1.pick a category 2.events of that category are shown		
Category:	System Operations		
Assumptions:	Events are shown		

Table 3.4: Use Case Description for <Browse Event>

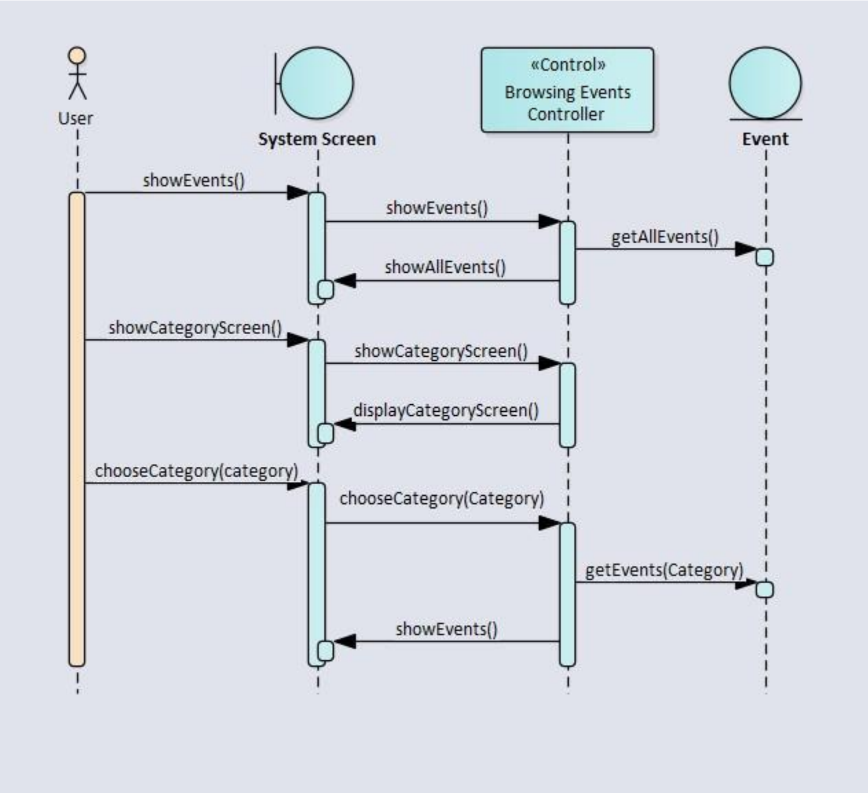


Figure 3.9: System Sequence Diagram of <Browse Event>

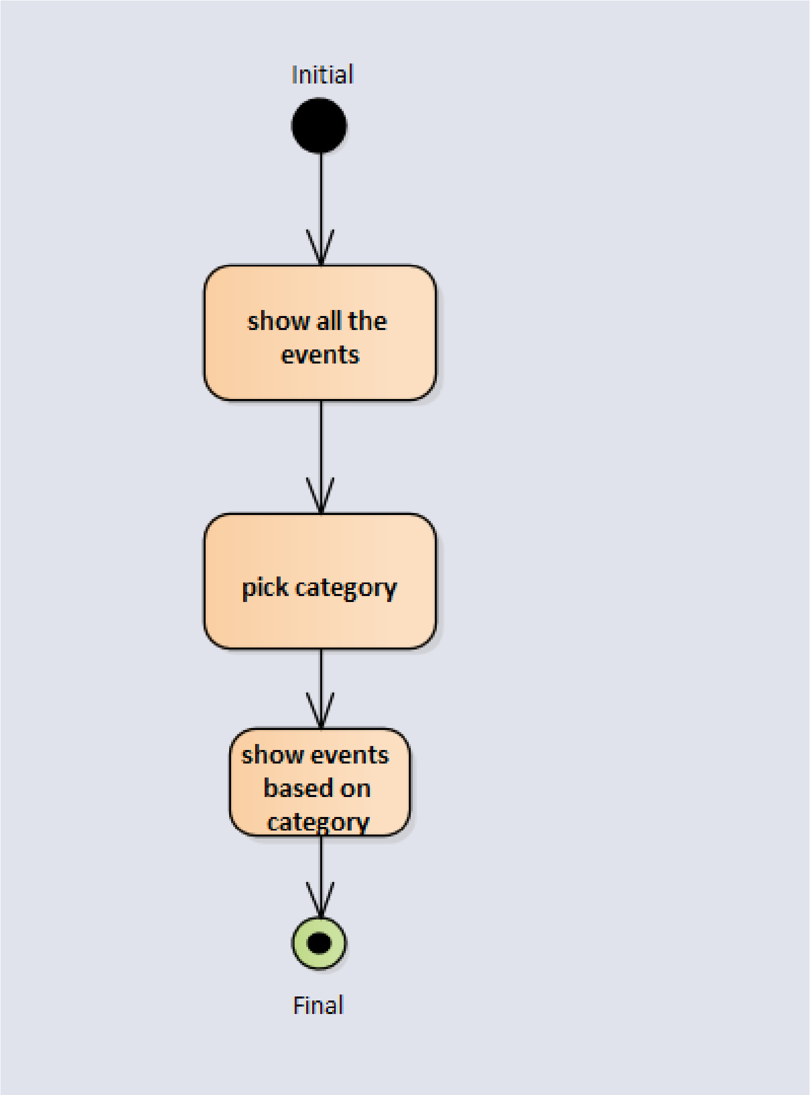
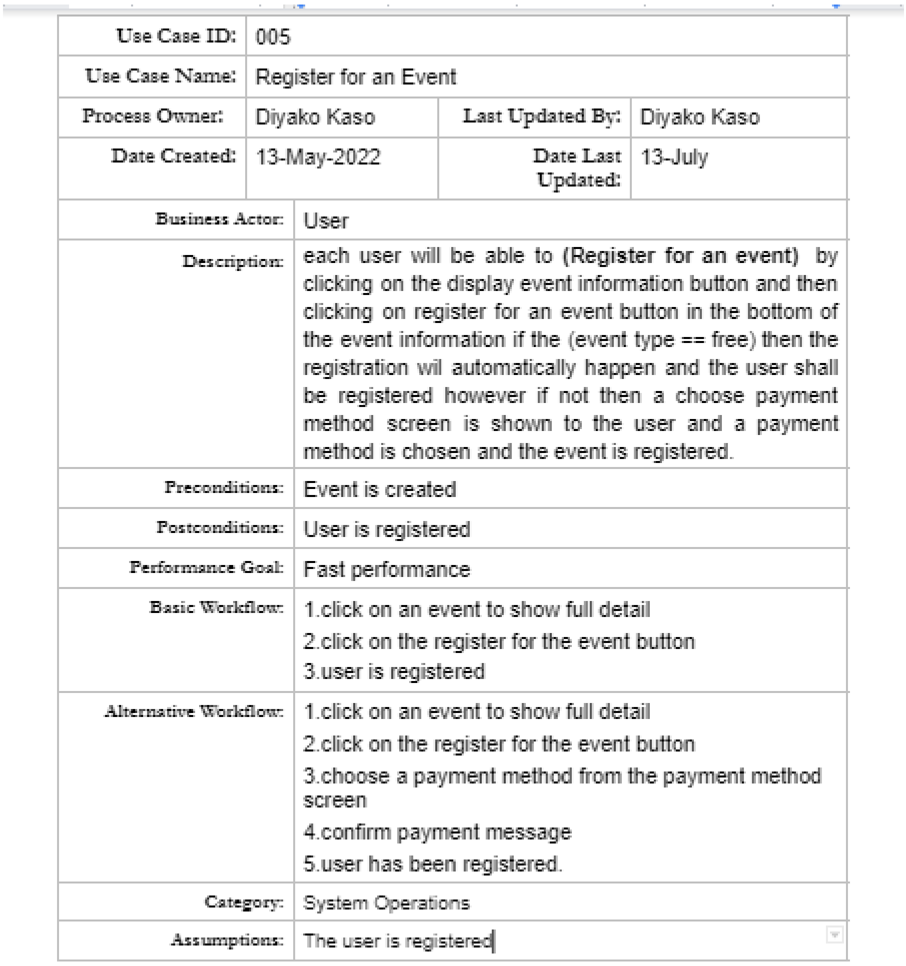


Figure 3.10: Activity Diagram of <Browse Event>

3.2.1.5 UC005 : Use Case <Register for an Event>



Use Case ID:	005		
Use Case Name:	Register for an Event		
Process Owner:	Diyako Kaso	Last Updated By:	Diyako Kaso
Date Created:	13-May-2022	Date Last Updated:	13-July
Business Actor:	User		
Description:	each user will be able to **(Register for an event)** by clicking on the display event information button and then clicking on register for an event button in the bottom of the event information if the (event type == free) then the registration wil automatically happen and the user shall be registered however if not then a choose payment method screen is shown to the user and a payment method is chosen and the event is registered.		
Preconditions:	Event is created		
Postconditions:	User is registered		
Performance Goal:	Fast performance		
Basic Workflow:	1.click on an event to show full detail 2.click on the register for the event button 3.user is registered		
Alternative Workflow:	1.click on an event to show full detail 2.click on the register for the event button 3.choose a payment method from the payment method screen 4.confirm payment message 5.user has been registered.		
Category:	System Operations		
Assumptions:	The user is registered		

Table 3.5: Use Case Description for <Register for an Event>

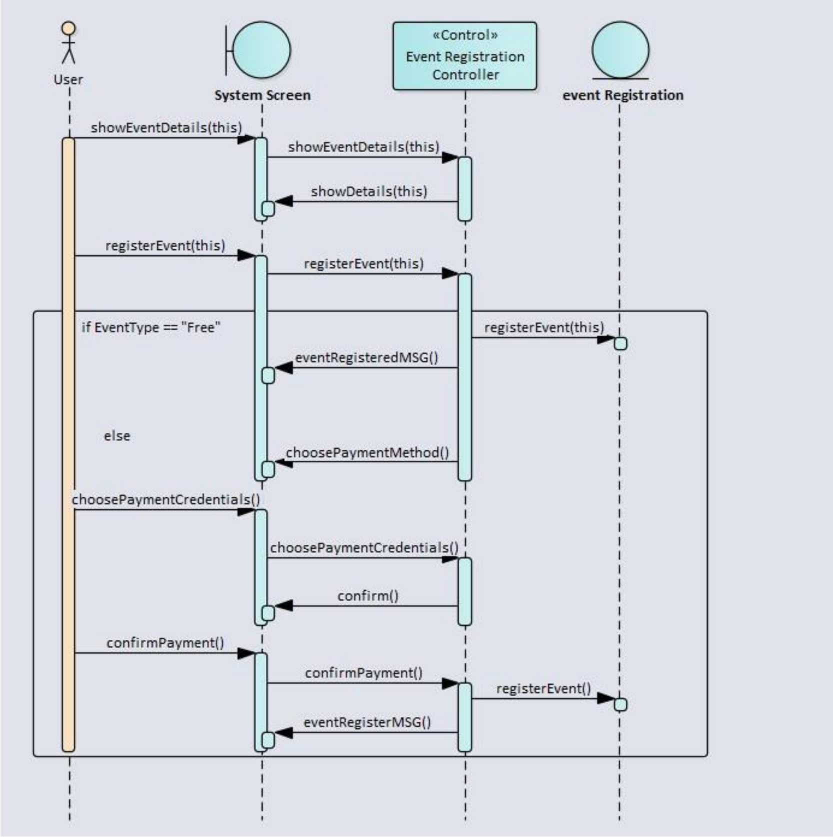


Figure 3.11: System Sequence Diagram of <Register for an Event>

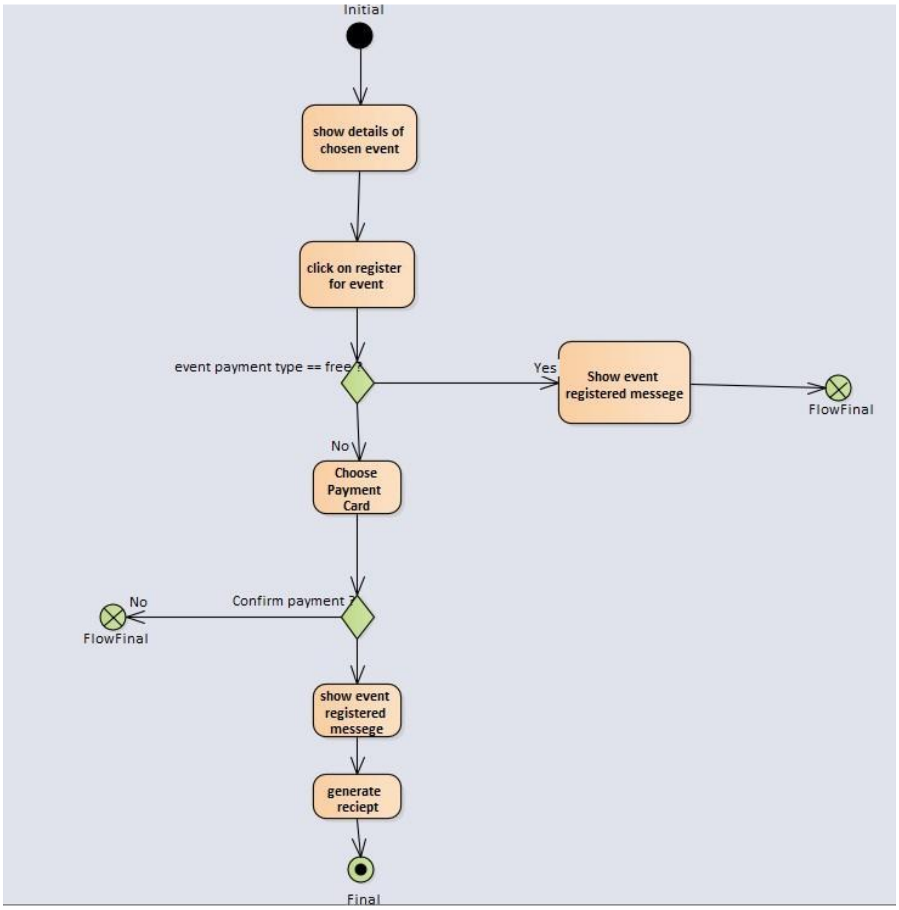
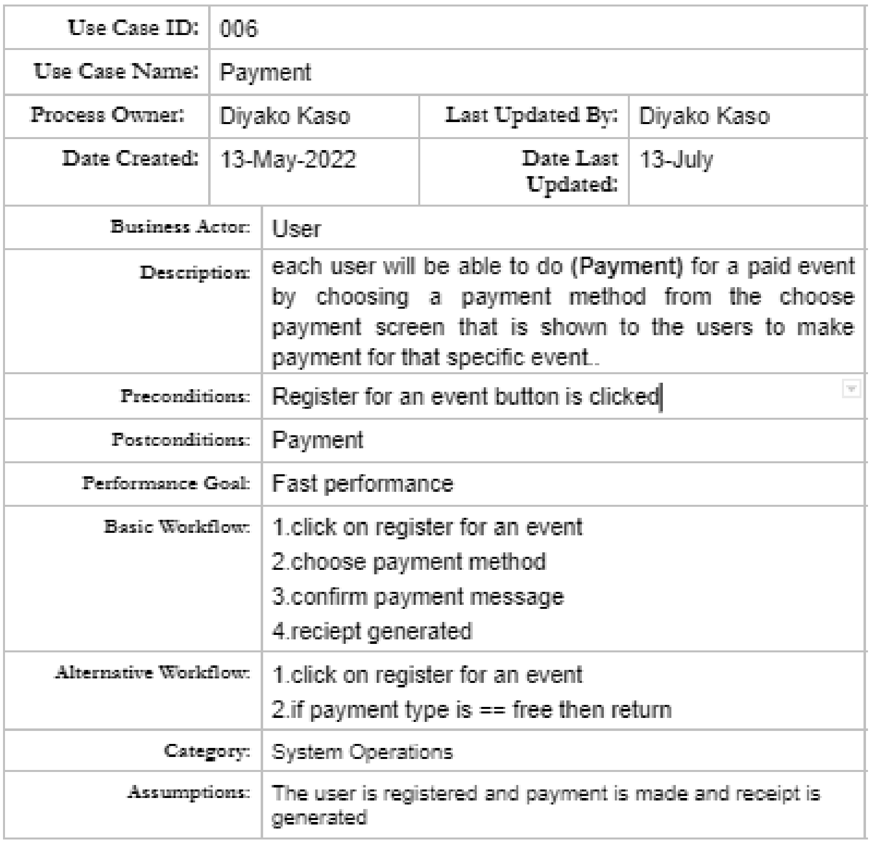

Initial
show details of chosen event
click on register for event
event payment type == free ?
Yes
Show event registered messege
FlowFinal
No
Choose Payment Card
No
Confirm payment ?
FlowFinal
show event registered messege
generate reciept
Final

Figure 3.12: Activity Diagram of <Register for an Event>

3.2.1.6 UC006 : Use Case <Payment>



Use Case ID:	006		
Use Case Name:	Payment		
Process Owner:	Diyako Kaso	Last Updated By:	Diyako Kaso
Date Created:	13-May-2022	Date Last Updated:	13-July
Business Actor:	User		
Description:	each user will be able to do **(Payment)** for a paid event by choosing a payment method from the choose payment screen that is shown to the users to make payment for that specific event..		
Preconditions:	Register for an event button is clicked		
Postconditions:	Payment		
Performance Goal:	Fast performance		
Basic Workflow:	1.click on register for an event 2.choose payment method 3.confirm payment message 4.reciept generated		
Alternative Workflow:	1.click on register for an event 2.if payment type is == free then return		
Category:	System Operations		
Assumptions:	The user is registered and payment is made and receipt is generated		

Table 3.6: Use Case Description for <Payment>

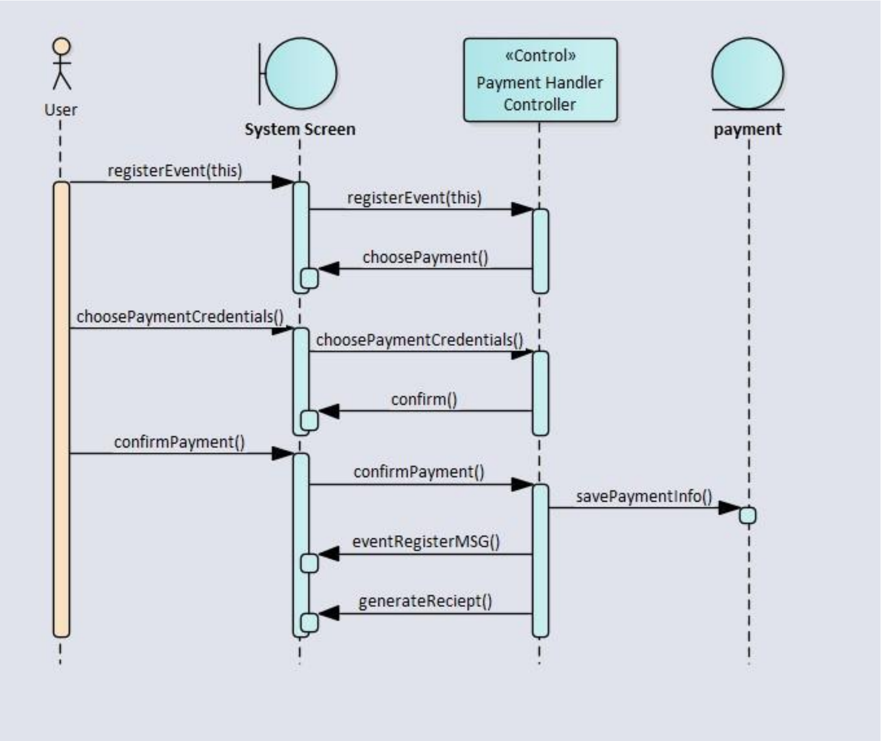


Figure 3.13: System Sequence Diagram of <Payment>

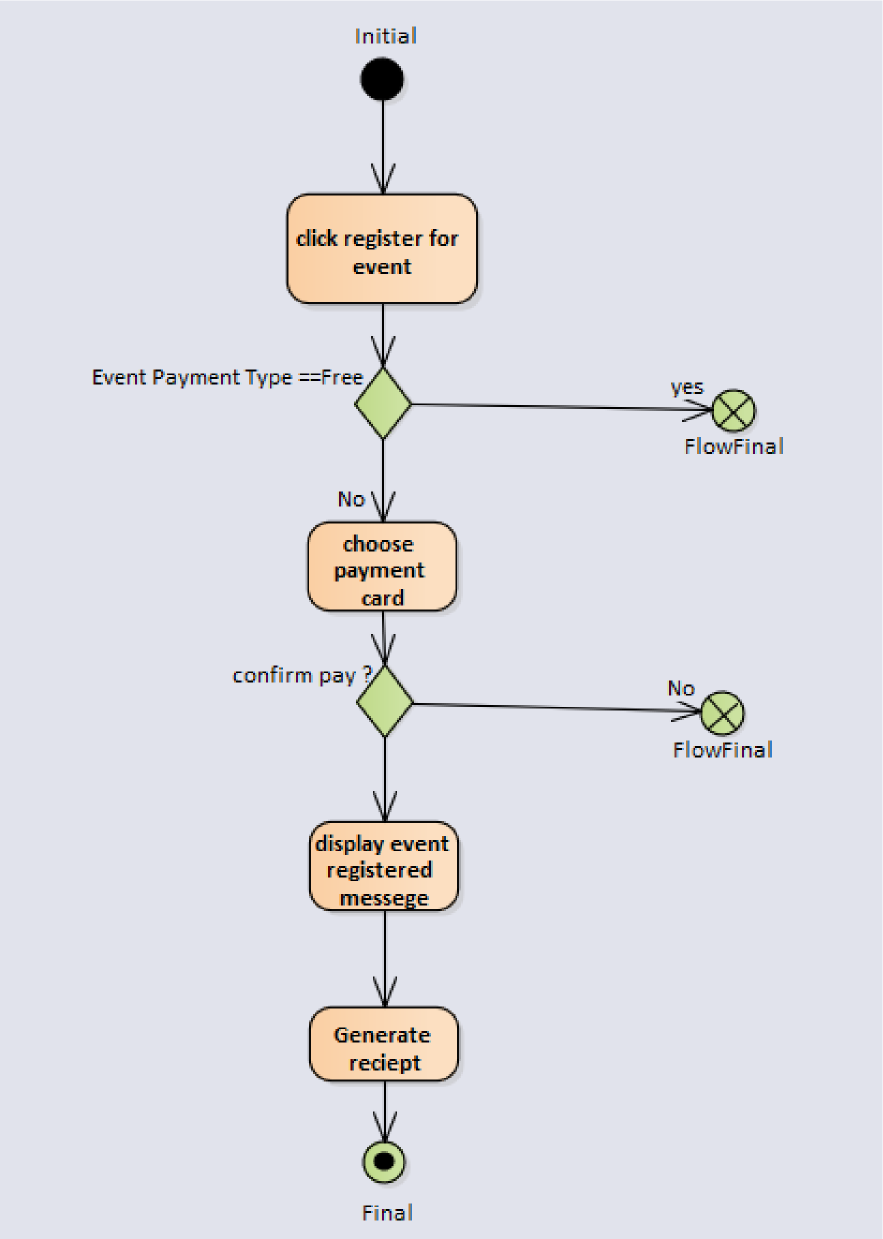
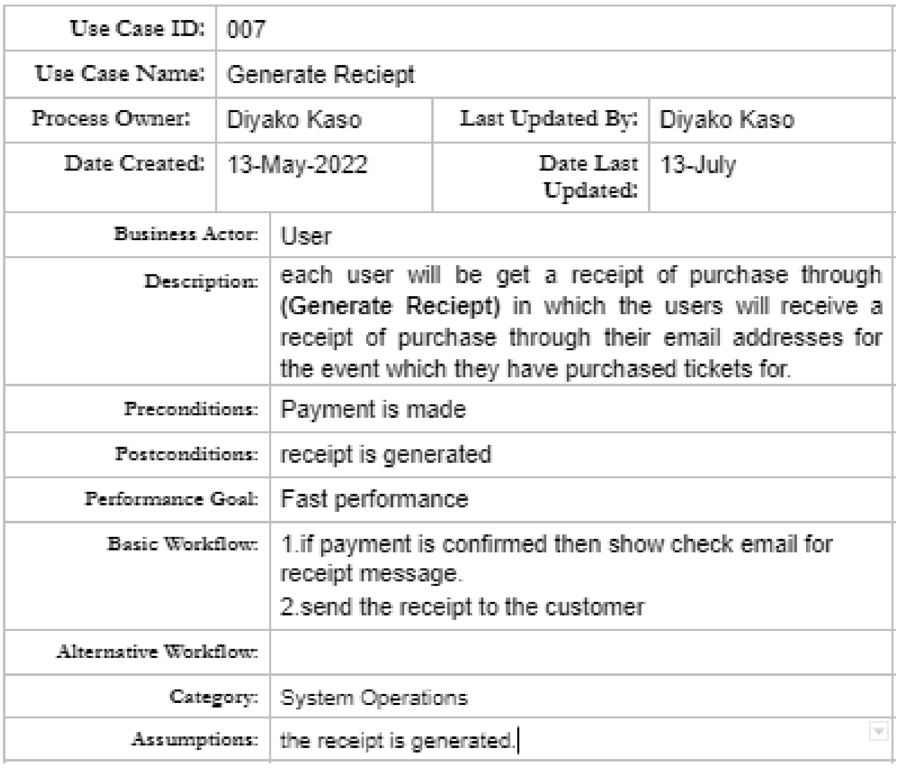


Figure 3.14: Activity Diagram of <Payment>

3.2.1.7 UC007 : Use Case <Generate Reciept>



Use Case ID:	007		
Use Case Name:	Generate Reciept		
Process Owner:	Diyako Kaso	Last Updated By:	Diyako Kaso
Date Created:	13-May-2022	Date Last Updated:	13-July

Business Actor:	User
Description:	each user will be get a receipt of purchase through **(Generate Reciept)** in which the users will receive a receipt of purchase through their email addresses for the event which they have purchased tickets for.
Preconditions:	Payment is made
Postconditions:	receipt is generated
Performance Goal:	Fast performance
Basic Workflow:	1.if payment is confirmed then show check email for receipt message. 2.send the receipt to the customer
Alternative Workflow:	
Category:	System Operations
Assumptions:	the receipt is generated.

Table 3.7: Use Case Description for <Generate Receipt>

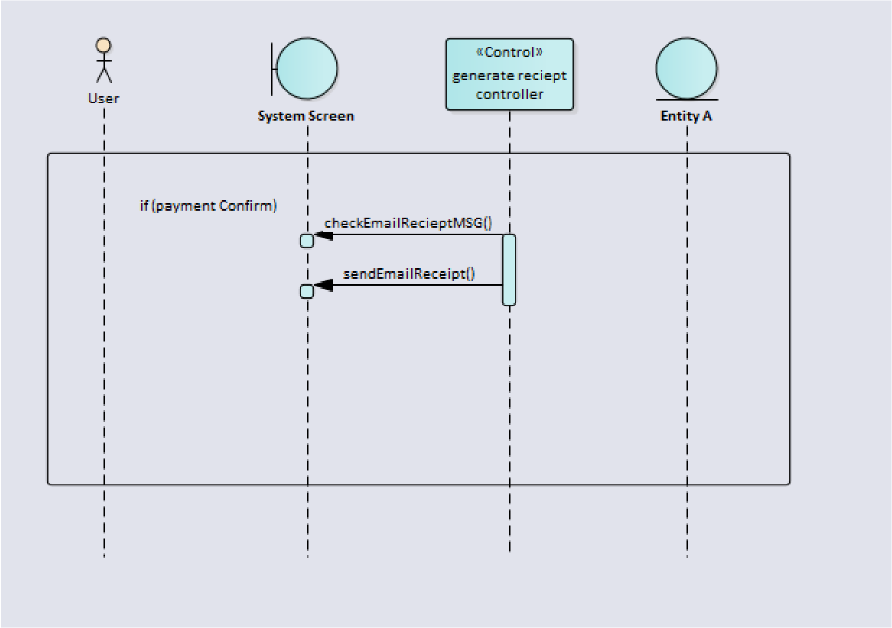


Figure 3.15: System Sequence Diagram of <Generate Receipt>

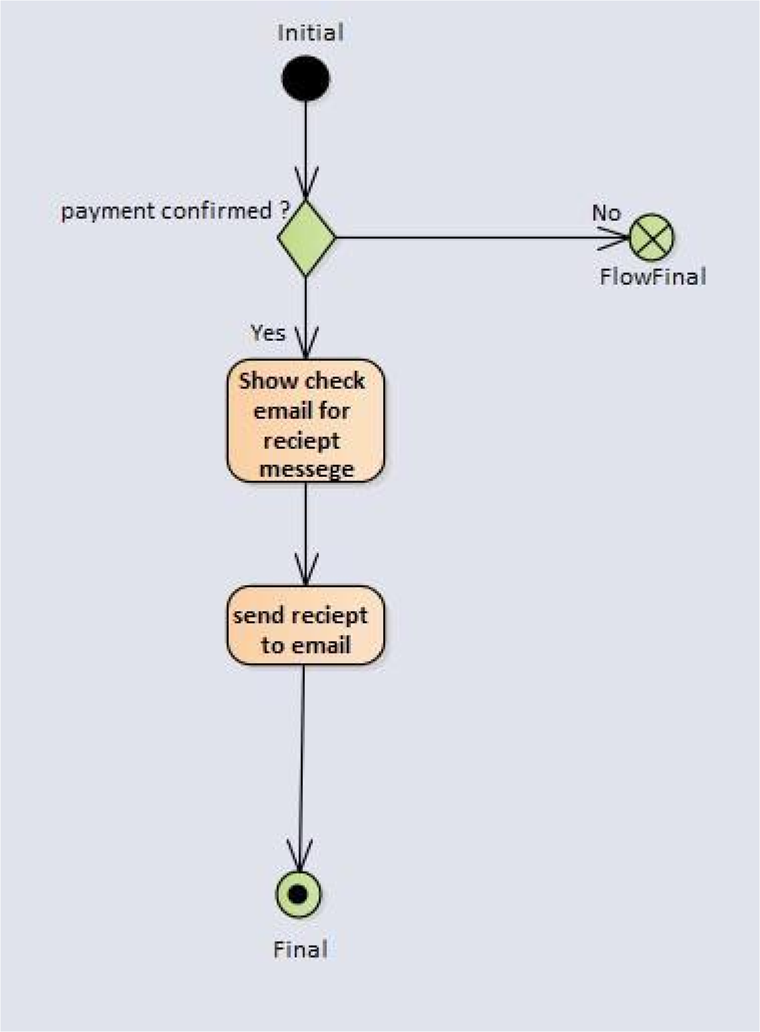
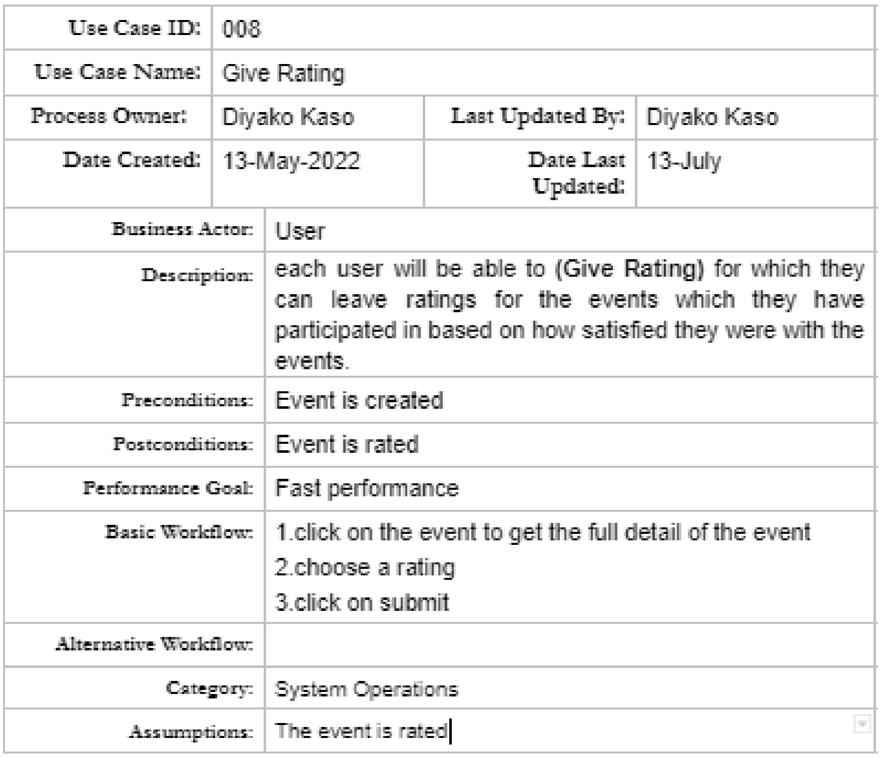


Figure 3.16: Activity Diagram of <Generate Receipt>

3.2.1.8 UC008 : Use Case <Give Rating>



Use Case ID:	008		
Use Case Name:	Give Rating		
Process Owner:	Diyako Kaso	Last Updated By:	Diyako Kaso
Date Created:	13-May-2022	Date Last Updated:	13-July
Business Actor:	User		
Description:	each user will be able to (Give Rating) for which they can leave ratings for the events which they have participated in based on how satisfied they were with the events.		
Preconditions:	Event is created		
Postconditions:	Event is rated		
Performance Goal:	Fast performance		
Basic Workflow:	1.click on the event to get the full detail of the event 2.choose a rating 3.click on submit		
Alternative Workflow:			
Category:	System Operations		
Assumptions:	The event is rated		

Table 3.8: Use Case Description for <Give Rating>

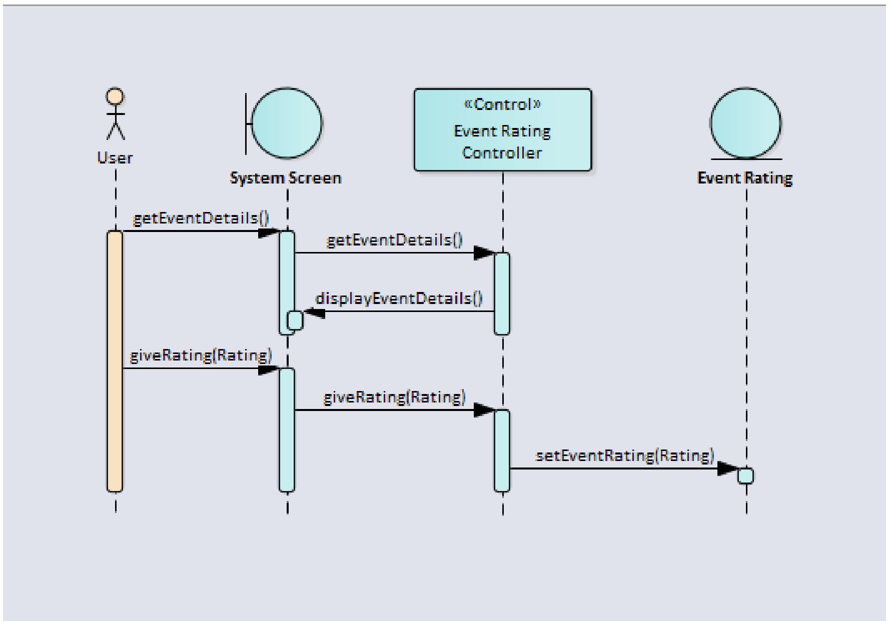


Figure 3.17 System Sequence Diagram of <Give Rating>

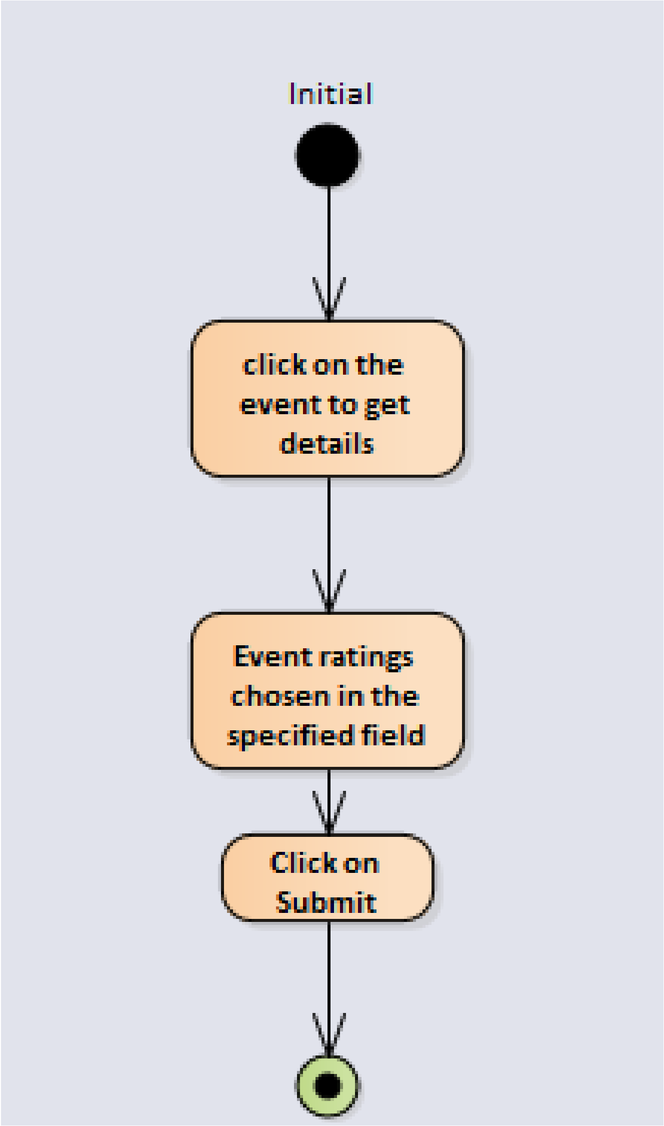
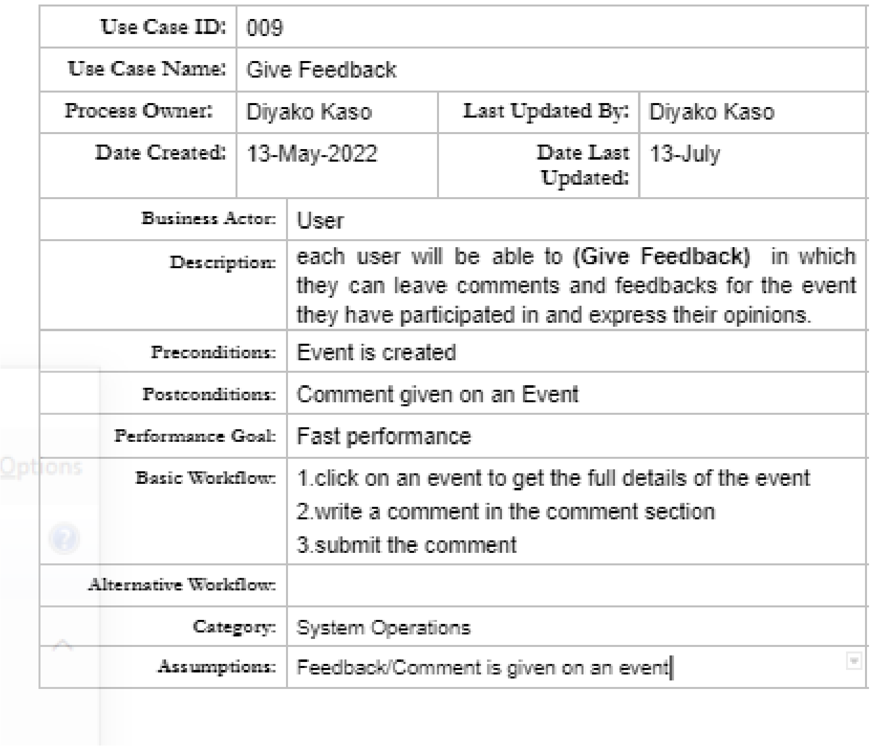

Initial
click on the event to get details
Event ratings chosen in the specified field
Click on Submit

Figure 3.18: System Activity Diagram of <Give Rating>

3.2.1.9 UC009 : Use Case <Give Feedback>



Use Case ID:	009		
Use Case Name:	Give Feedback		
Process Owner:	Diyako Kaso	Last Updated By:	Diyako Kaso
Date Created:	13-May-2022	Date Last Updated:	13-July
Business Actor:	User		
Description:	each user will be able to **(Give Feedback)** in which they can leave comments and feedbacks for the event they have participated in and express their opinions.		
Preconditions:	Event is created		
Postconditions:	Comment given on an Event		
Performance Goal:	Fast performance		
Basic Workflow:	1.click on an event to get the full details of the event 2.write a comment in the comment section 3.submit the comment		
Alternative Workflow:			
Category:	System Operations		
Assumptions:	Feedback/Comment is given on an event		

Table 3.9: Use Case Description for <Give Feedback>

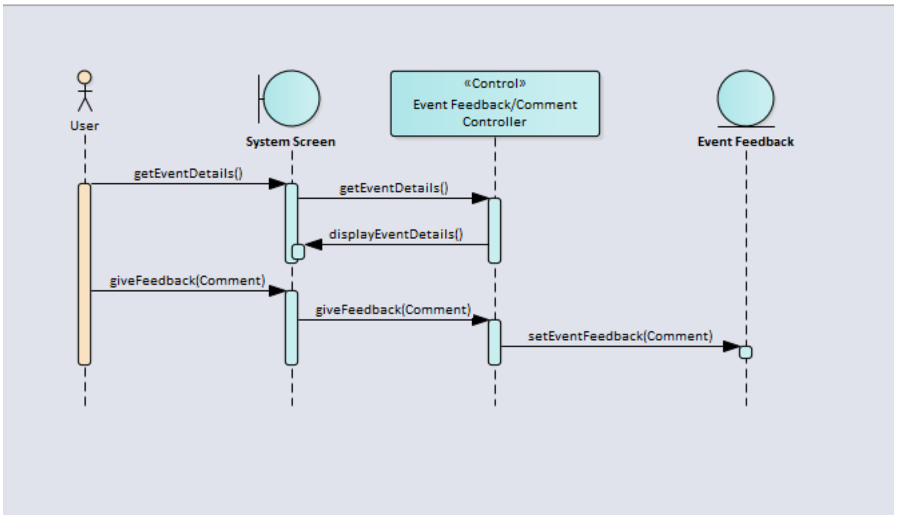


Figure 3.19: System Sequence Diagram of <Give Rating>

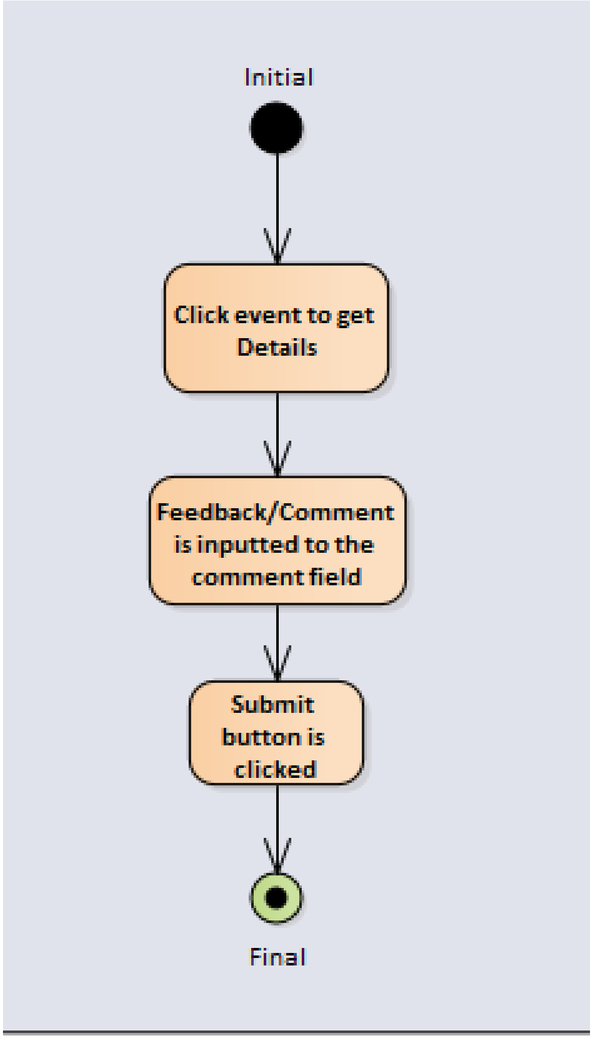

Initial
Click event to get Details
Feedback/Comment is inputted to the comment field
Submit button is clicked
Final

Figure 3.20: System Activity Diagram of <Give Feedback>

3.3 Performance Requirements

the <system> shall allow the <user> to create events the

<system> shall allow the <user> to modify events the

<system> shall allow the <user> to delete events

the <system> shall allow the <user> to browse events the

<system> shall allow the <user> to register for events the

<system> shall allow the <user> to pay for events

the <system> shall be able to generate receipts

3.4 Design Constraints

None

3.5 Software System Attributes

The system must be **attractive** with **beautiful** and **colorful** user interface and must be **easy to use** by the most basic user and not spend too much time learning the system and the system must be **secure** as it deals with payment and critical data.

3.6 Other Requirements

Usability : our system should be user friendly with an easy to use user interface that is intuitive and can be picked up easily.

Reliability : our system should be reliable under a large amount of user requests and usage of the system at one time unit and not crash once the requests and the queries are too much for the system at a time.

Maintainability : our system should be developed with future modification and improvements in mind and should be maintainable and understood easily by future employees who work on the system.

SCSJ3323: Software Design and Architecture

Software Design Document

Design and Implementation of Online Event Management System

Version 1.0

14th - July - 2022

Faculty of Engineering and Science - Software Engineering

Prepared by: <Diyako Kaso>

Revision Page

a. Overview

This version of the system describes the general overview and introduction of the system and then dives to the use case diagrams and sequence and activity diagrams of each use case along with a detailed description of each use case using usecase descriptions as well as many types of system and design and additional constrains have been discussed

b. Target Audience

System Architects , Software Engineers , Software Quality Assurance

c. Project Team Members

Diyako Kaso Chalk

d. Version Control History

Version	Primary Author(s)	Description of Version	Date Completed
<1.0>	Diyako Kaso Chalak	This version of the system describes the general overview and introduction of the system and then dives to the use case diagrams and sequence and activity diagrams of each use case along with a detailed description of each use case using usecase descriptions as	

		well as many types of system and design and additional constrains have been discussed	

Note:

This template is an annotated outline for a software design document adapted from the IEEE Recommended Practice for Software Design Descriptions. The IEEE Recommended Practice for Software Design Descriptions have been reduced in order to simplify this assignment while still retaining the main components and providing a general idea of a project definition report. Please refer to IEEE Std 10161998 1 for the full IEEE Recommended Practice for Software Design Descriptions. Examples of models are from Satzinger (2011). Compiled by Shahliza Abdul Halim, PhD and checked by Shahida Sulaiman, PhD on 2 May 2016.

Table of Contents

4 Data Design

4.1 Data Description

4.2 Data Dictionary

5 User Interface Design

5.1 Overview of User Interface

5.2 Screen Images

6 Requirements Matrix

7 Appendices

Appendices (if any)

1. Introduction

Event management systems are nowadays being used extensively to manage events of all sorts and types. The most important factor that led to the mass public usage of such event systems is innovation in live event technology (Brandon,2018). Event management systems are defined as systems that are used to manage and organize activities related to a certain event or events (Chrisantus Oden , n.d.). An event management software helps event organizers to register their events and to allow the customers to book their attendance (check-in) and to promote their events on the software. Some event management software only specializes in a type of functionality such as participant booking (check-in) this leads to a limited amount of functionality. Typical operations of an event management System are but not limited to (Event Registration, Event Marketing) (Brandon, 2018).

The technical aspects of the development of our own proposed system shall be discussed in terms of the programming language that is going to be used to program the functionalities of the system, such as dart programming language and the flutter framework and such technology shall be discussed, labelling their performance and security and the features that they are possessing.

1.1 Purpose

- To Elicit the requirements of the community in terms of Event Management
- To design and implement the system to address those issues
- To test the system in terms of Functional and non-functional requirements and completeness

1.2 Scope

- The system will be developed only for the City of Sulaymaniyah in the initial phases

- The system development will use Mobile Application languages and frameworks such as dart programming language and the flutter framework for mobile application development.

- The system will be developed using Visual Studio Code code Editor.

1.3 Definitions, Acronyms and Abbreviation

None

1.4 References

Rafalson,B . (2018,January 3rd) , *Why You Need Event Management Software,* bizzabo, Why You Need Event Management Software - Bizzabo | Bizzabo

Chrisantus Oden , n.d. , *Design and Implementation of an Online Event Management System,* projectTopics , Design and Implementation of an Online Event Management System - Project Topics

1.5 Overview

The aim of this project is to re-ignite the cultural movement in the city by developing an efficient and easy to use (Design and implementation of online event Management system) system to drive more traffic to the events and to make people aware of the events in order to increase the networking opportunities and to expand the opportunity spaces beyond the stereotypical jobs and ways of making a living and the classical way of networking with the aim of paving way for an opportunity filled city that has opportunities on every corner and every person from every background and major can make contribution to the cultural and artistic and academic movement in our city.

2. System Architectural Design

2.1 Architecture Style and Rationale

The chosen architectural style is Model-View-Controller (MVC) due to it’s flexibility in development and adding more maintainability to the code base and separating the implementation from the data base.

2.2 Architecture Model

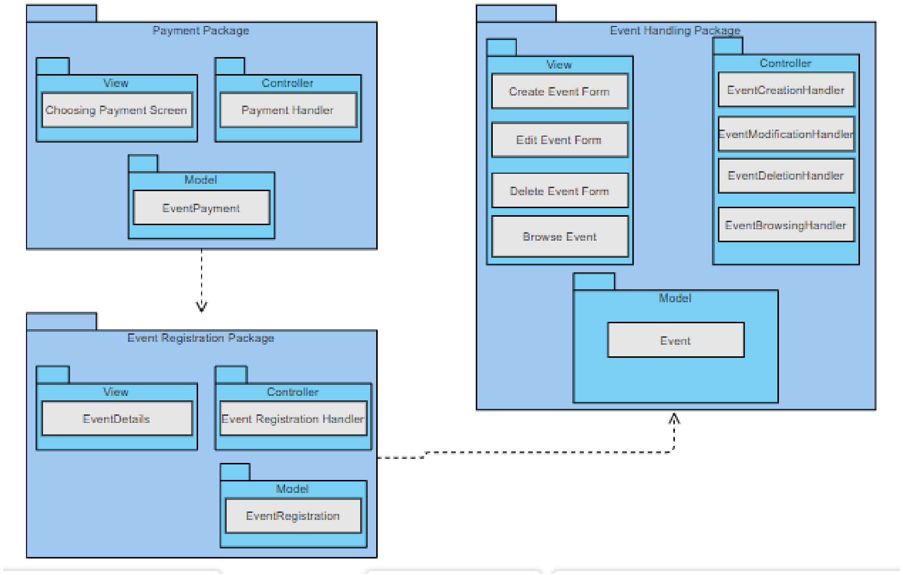


Figure 2.1: Component Model of <Online Event Management System>

2.3 Use Case Diagram

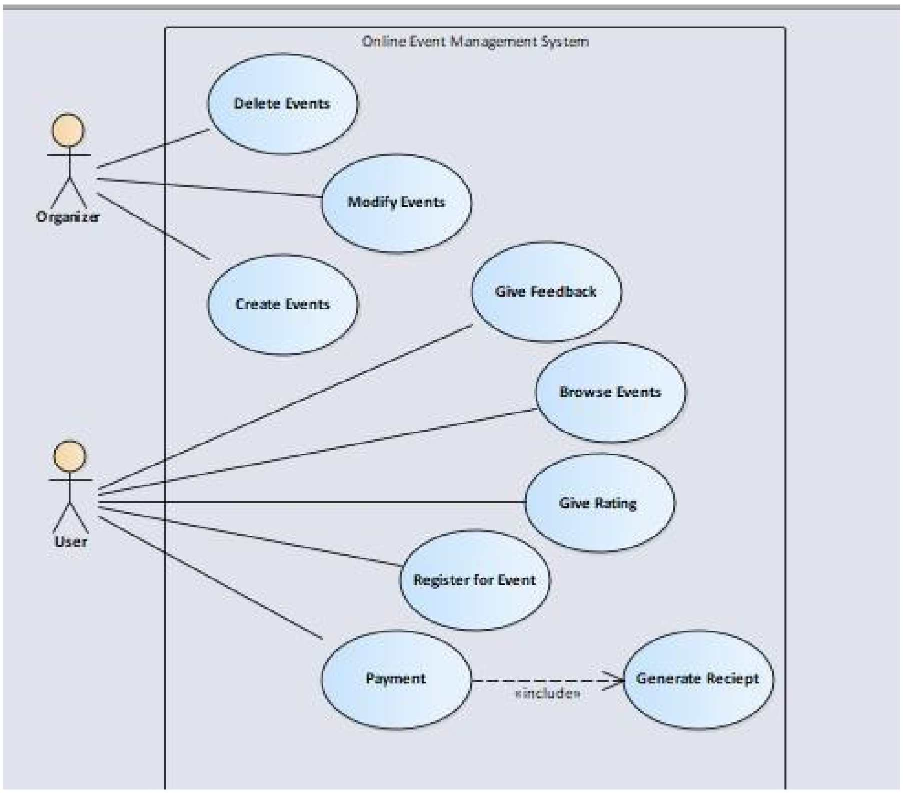


Figure 2.2: Use Case Diagram of <Online Event Management System>

3. Detailed Description of Components

3.1 Complete Package Diagram

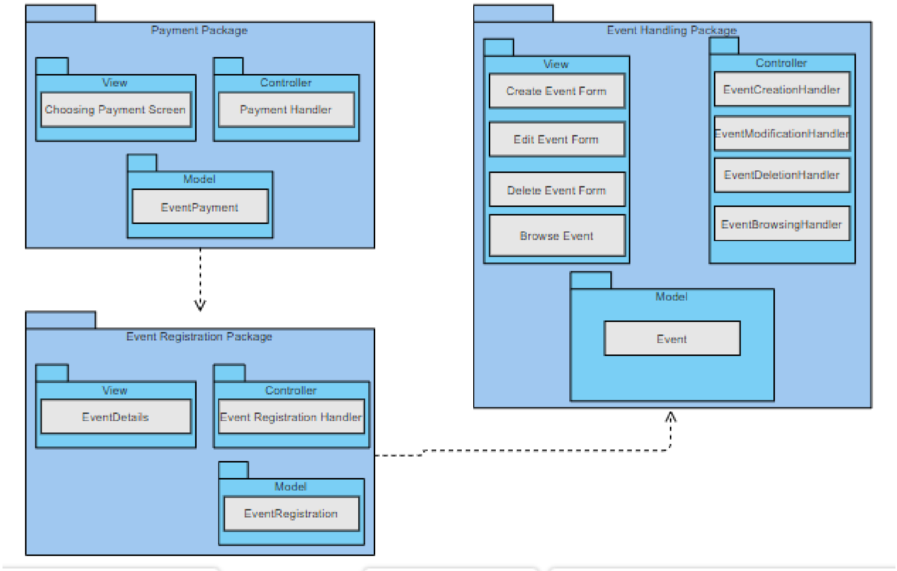


Figure 3.1: Package Diagram of <Online Event Management System>

3.2 Component Model

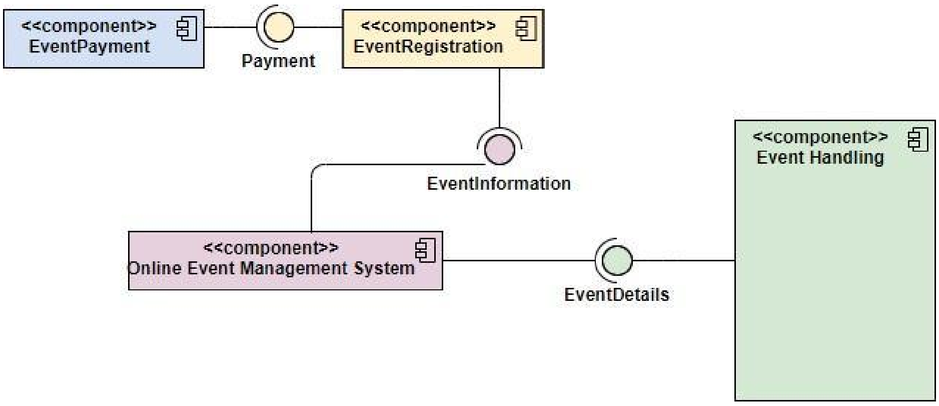


Figure 3.2: Component Diagram of <Online Event Management System>

3.3 Detailed Description

3.3.1 Subsystem <Payment Package>

3.3.1.1 P001: Package <Payment Package>

3.3.1.2 Class Diagram

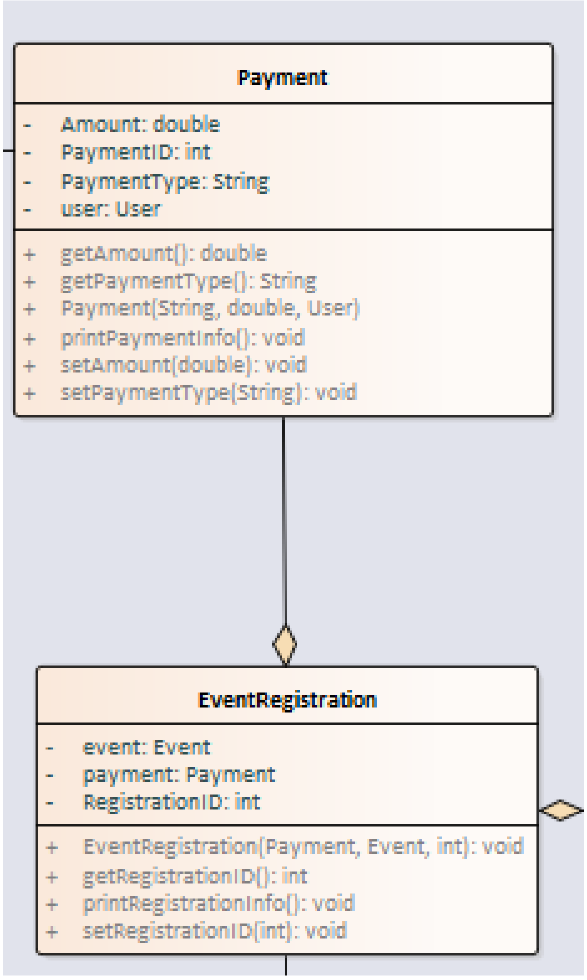

Payment
- Amount: double
- PaymentID: int
- PaymentType: String
- user: User
+ getAmount(): double
+ getPaymentType(): String
+ Payment(String, double, User)
+ printPaymentInfo(): void
+ setAmount(double): void
+ setPaymentType(String): void
EventRegistration
- event: Event
- payment: Payment
- RegistrationID: int
+ EventRegistration(Payment, Event, int): void
+ getRegistrationID(): int
+ printRegistrationInfo(): void
+ setRegistrationID(int): void

Figure 3.2: Class diagram for <Payment Package>

3.3.1.3 Sequence Diagrams

a) SD001: Sequence diagram for Payment

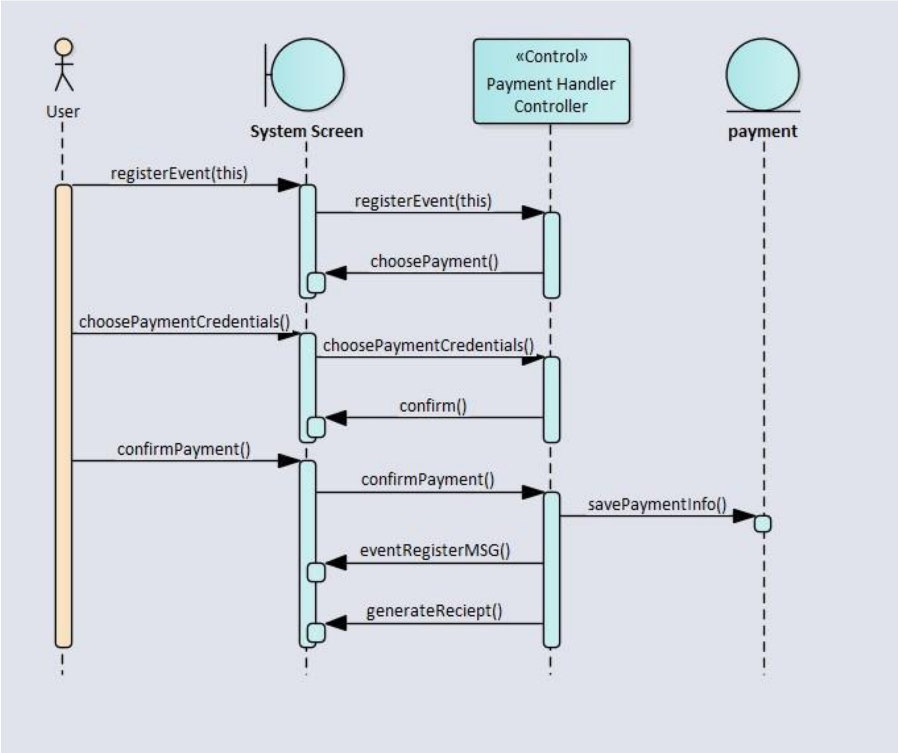


Figure 3.3: Sequence Diagram of <Payment>

b) SD002: Sequence diagram for Generate Reciept

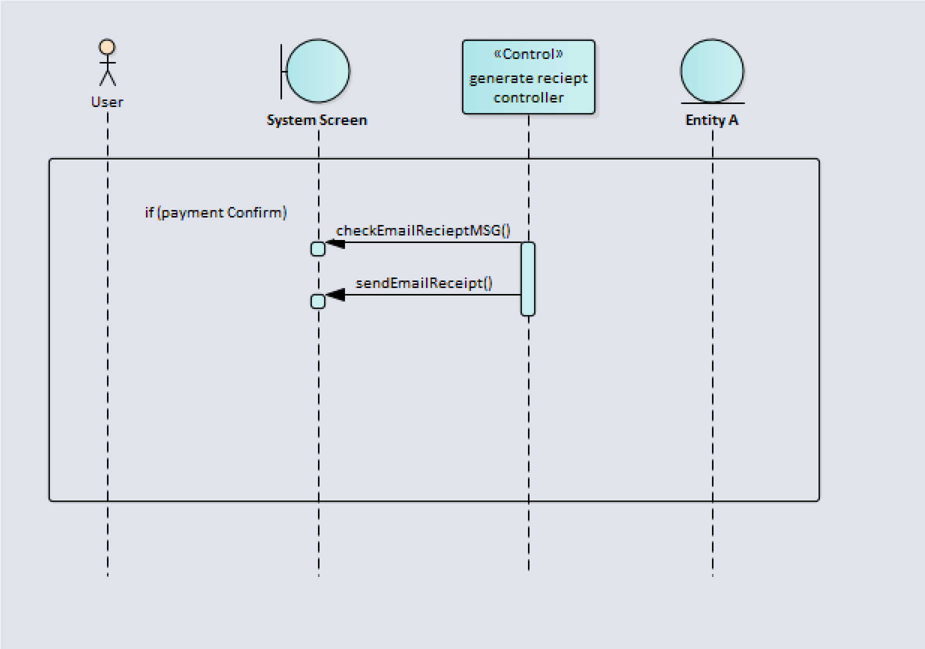


Figure 3.4: Sequence Diagram of <Generate Reciept>

3.3.2 Subsystem <Event Handling>

3.3.2.1 P002: Package <Event Handling>

3.3.2.2 Class Diagram

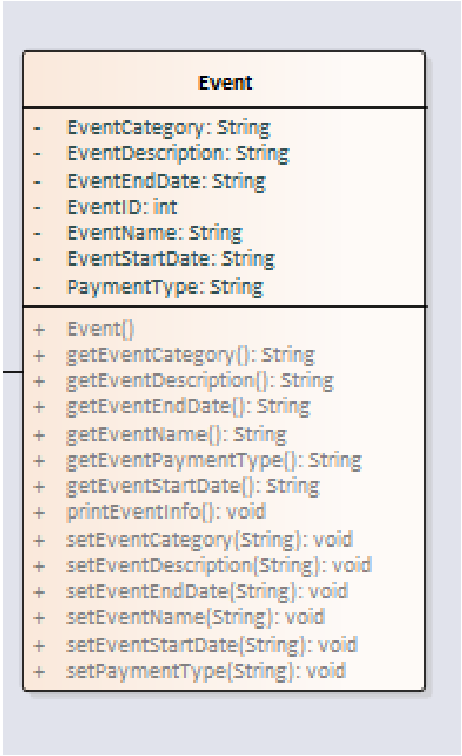

Event
- EventCategory: String
- EventDescription: String
- EventEndDate: String
- EventID: int
- EventName: String
- EventStartDate: String
- PaymentType: String
+ Event()
+ getEventCategory(): String
+ getEventDescription(): String
+ getEventEndDate(): String
+ getEventName(): String
+ getEventPaymentType(): String
+ getEventStartDate(): String
+ printEventInfo(): void
+ setEventCategory(String): void
+ setEventDescription(String): void
+ setEventEndDate(String): void
+ setEventName(String): void
+ setEventStartDate(String): void
+ setPaymentType(String): void

Figure 3.5: Class diagram for <Event Pacakge>

3.3.2.3 Sequence Diagrams

a) SD001: Sequence diagram for Create Event

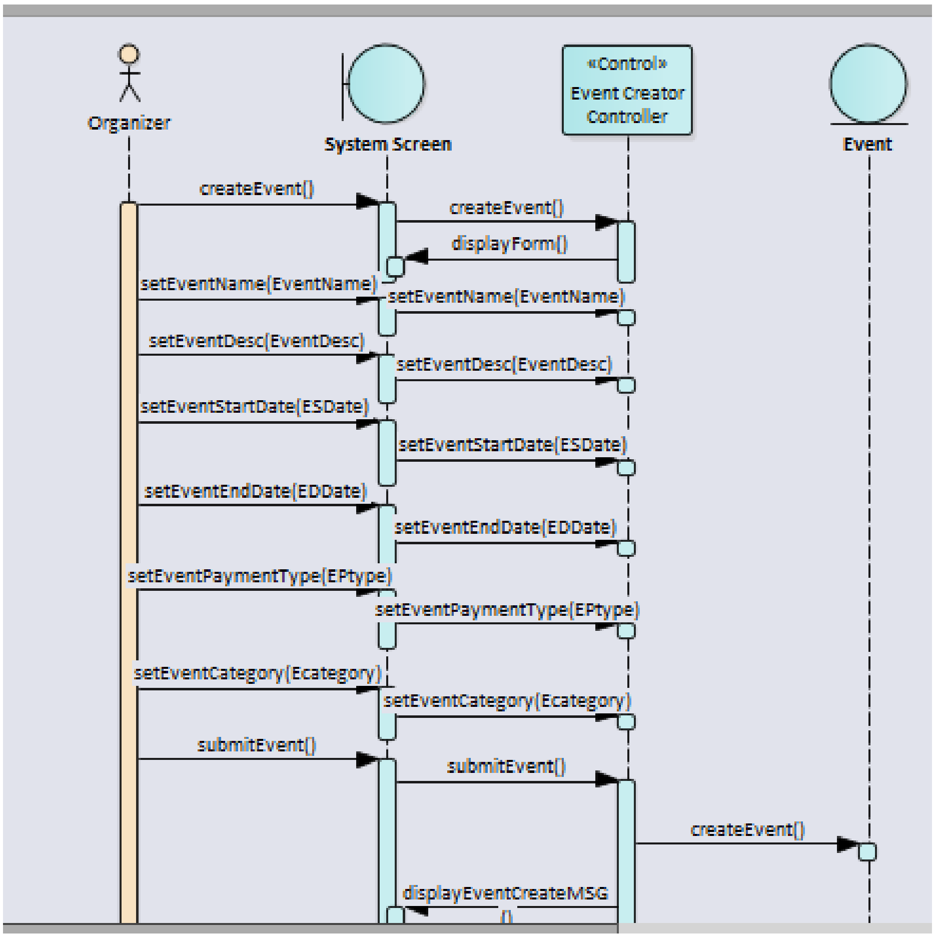


Figure 3.6: Sequence Diagram of <Create Event>

b) SD002: Sequence diagram for Modify Event

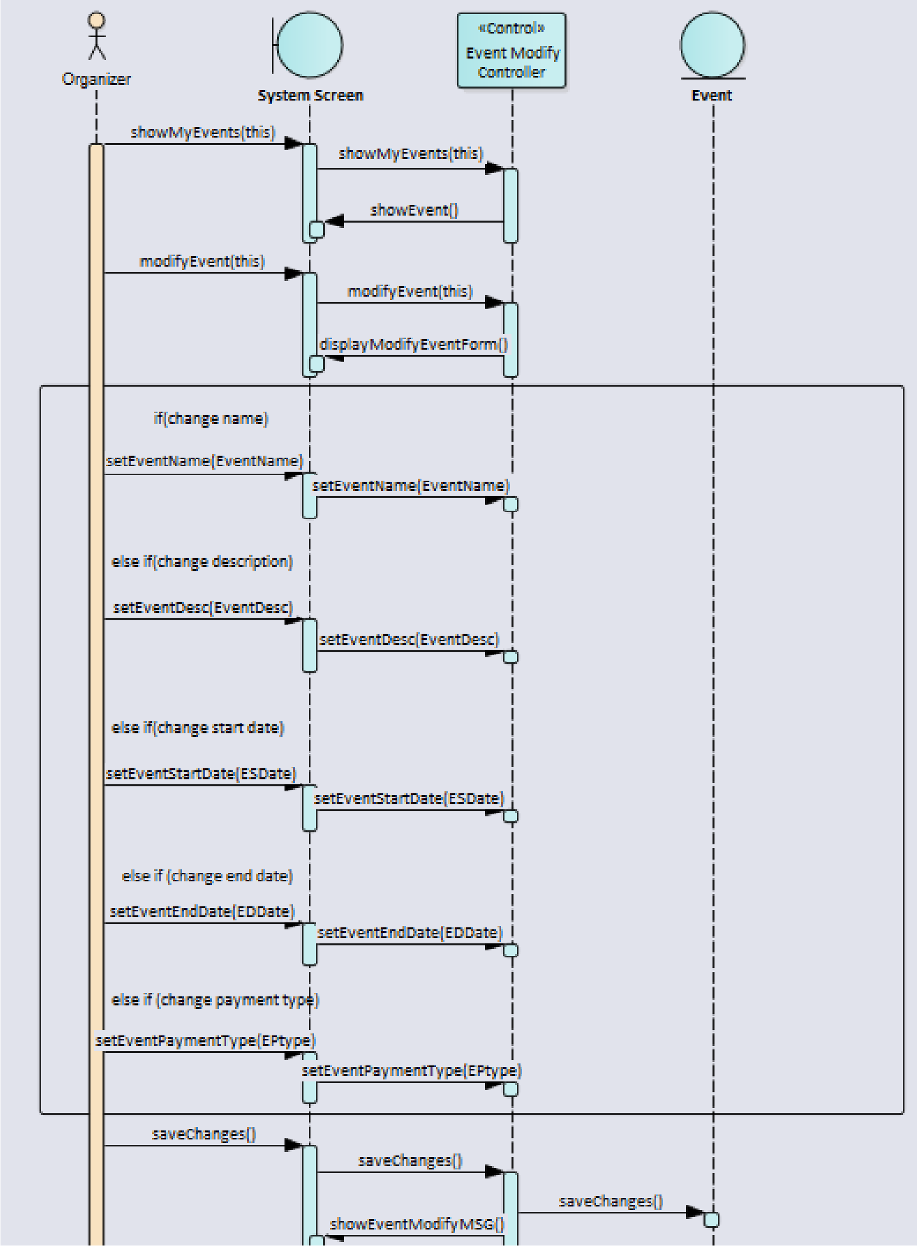


Figure 3.7: Sequence Diagram of <Modify Event>

c) SD003: Sequence diagram for Delete Event

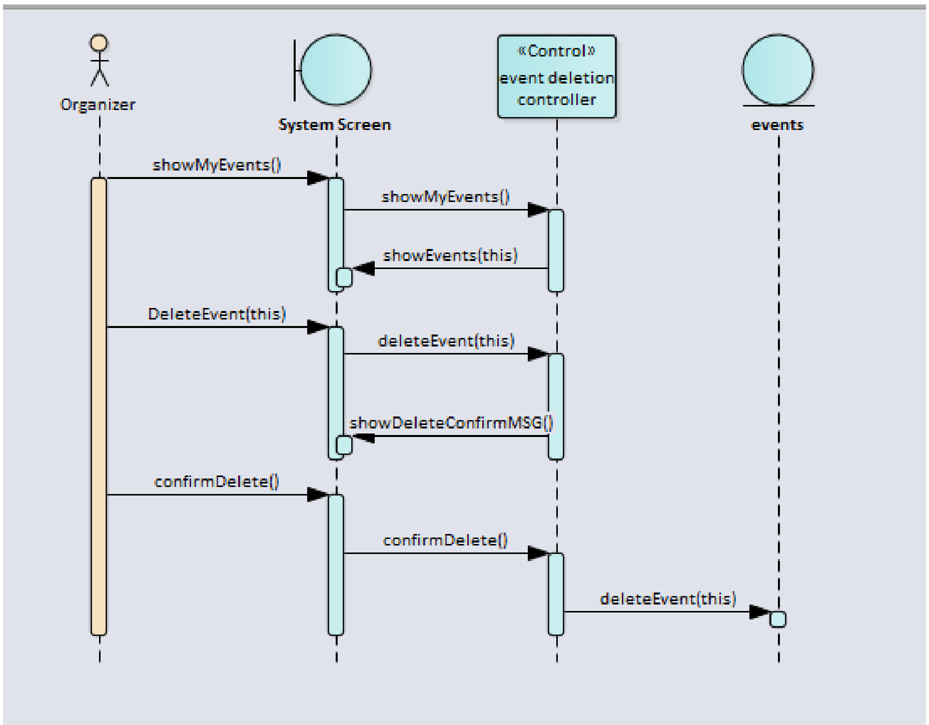


Figure 3.8: Sequence Diagram of <Delete Event>

d) SD004: Sequence diagram for Browse Event

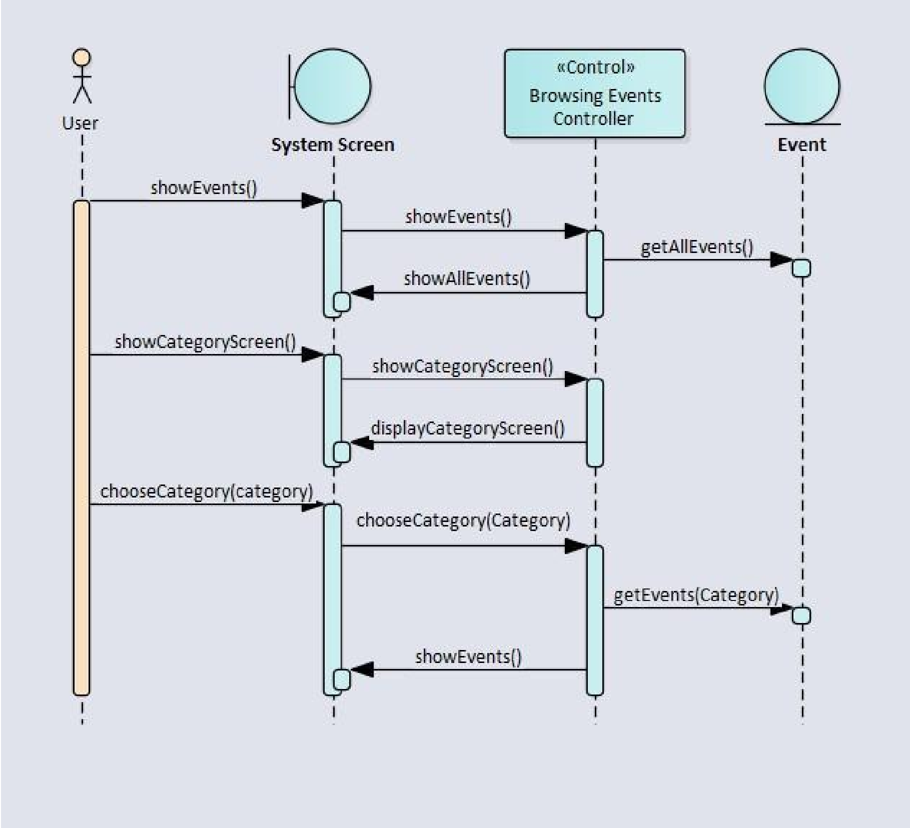


Figure 3.9: Sequence Diagram of <Browse Event>

e) SD005: Sequence diagram for Give Feedback

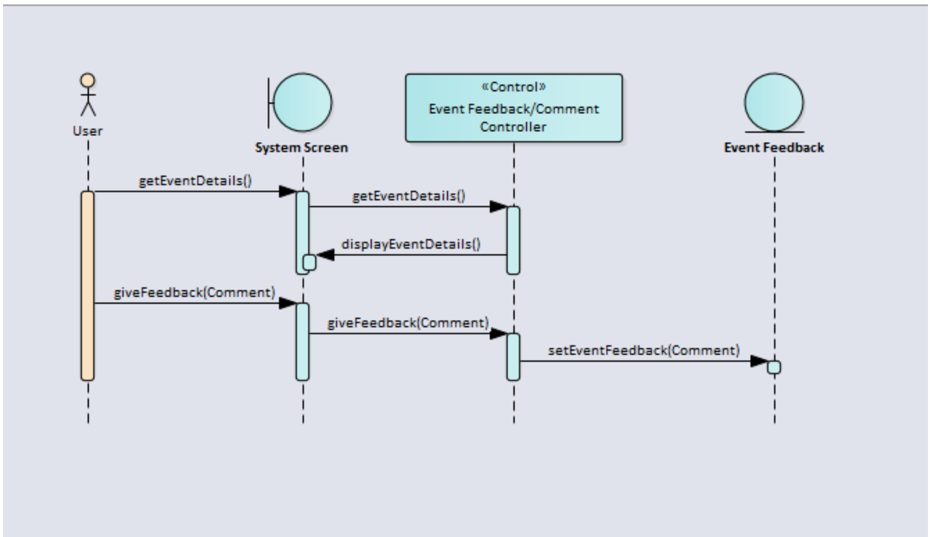


Figure 3.10: Sequence Diagram of <Give Feedback>

f) SD006: Sequence diagram for Give Rating

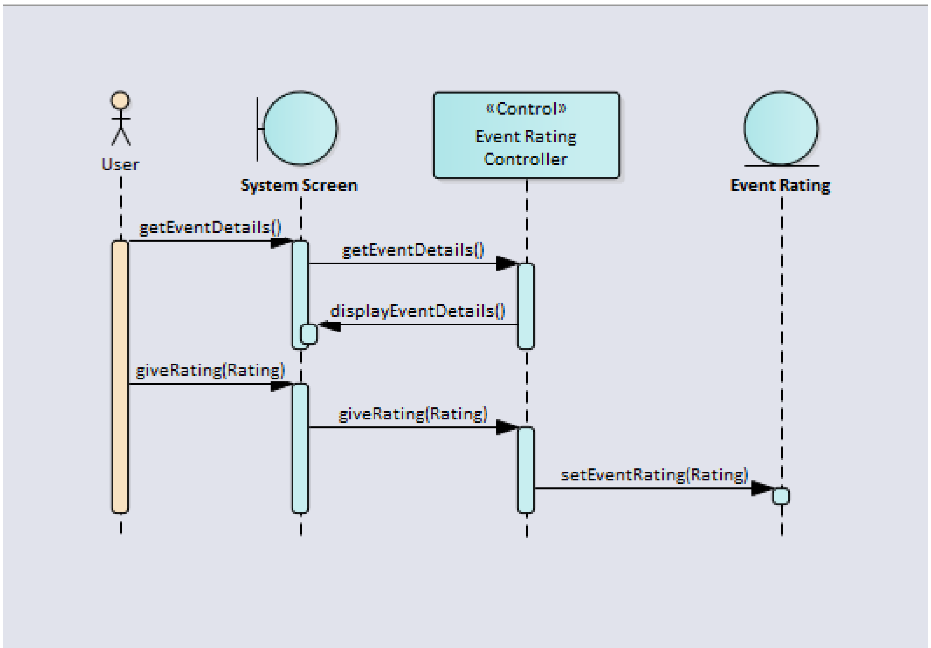


Figure 3.11: Sequence Diagram of <Give Rating>

3.3.3 Subsystem <Event Registration>

3.3.3.1 *P003*: Package <Event Registration>

3.3.3.2 Class Diagram

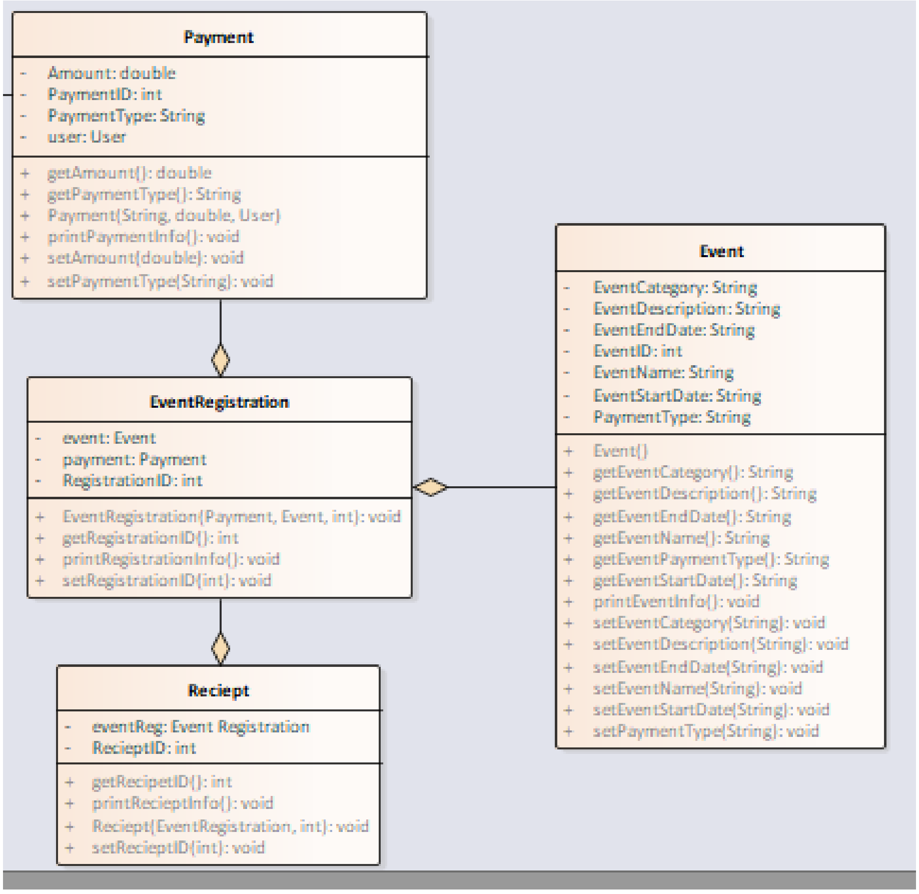

Payment
- Amount: double
- PaymentID: int
- PaymentType: String
- user: User
+ getAmount(): double
+ getPaymentType(): String
+ Payment(String, double, User)
+ printPaymentInfo(): void
+ setAmount(double): void
+ setPaymentType(String): void
EventRegistration
- event: Event
- payment: Payment
- RegistrationID: int
+ EventRegistration(Payment, Event, int): void
+ getRegistrationID(): int
+ printRegistrationInfo(): void
+ setRegistrationID(int): void
Reciept
- eventReg: Event Registration
- RecieptID: int
+ getRecipetID(): int
+ printRecieptInfo(): void
+ Reciept(EventRegistration, int): void
+ setRecieptID(int): void
Event
- EventCategory: String
- EventDescription: String
- EventEndDate: String
- EventID: int
- EventName: String
- EventStartDate: String
- PaymentType: String
+ Event()
+ getEventCategory(): String
+ getEventDescription(): String
+ getEventEndDate(): String
+ getEventName(): String
+ getEventPaymentType(): String
+ getEventStartDate(): String
+ printEventInfo(): void
+ setEventCategory(String): void
+ setEventDescription(String): void
+ setEventEndDate(String): void
+ setEventName(String): void
+ setEventStartDate(String): void
+ setPaymentType(String): void

Figure 3.12: Class diagram for <EventEegistrationPacakge>

3.3.3.3 Sequence Diagrams

a) SD001: Sequence diagram for Resgister for an event

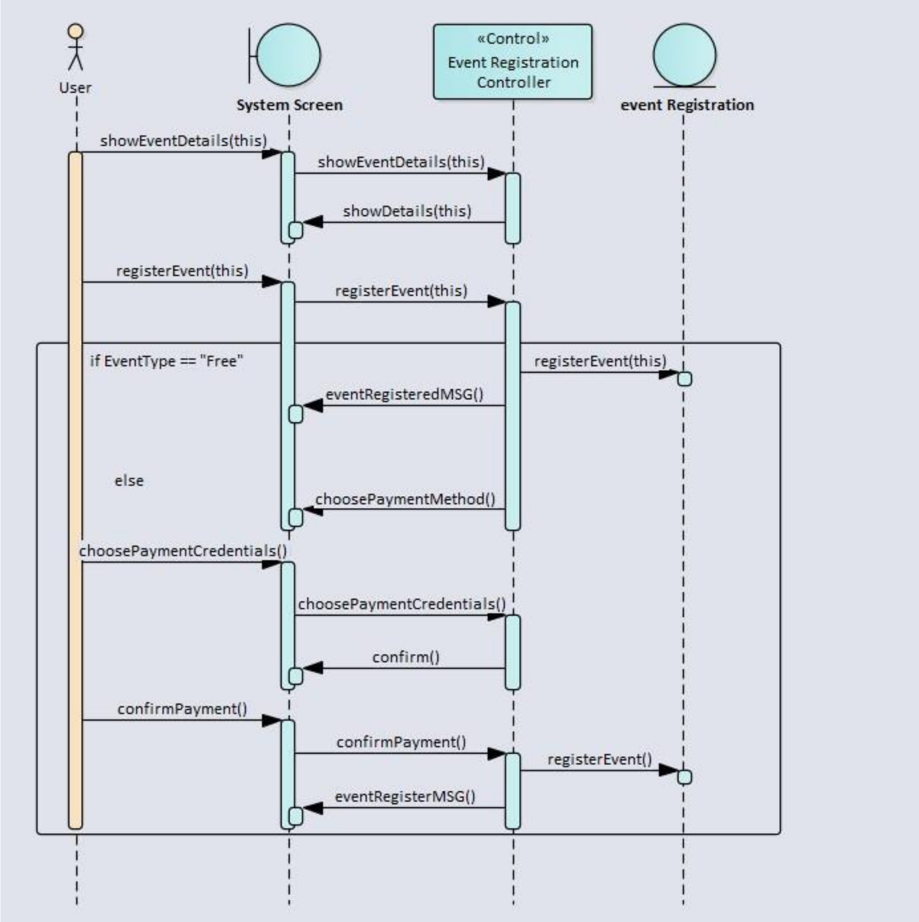


Figure 3.13: Sequence diagram for <Register for an event>

b) SD002: Sequence diagram for Payment

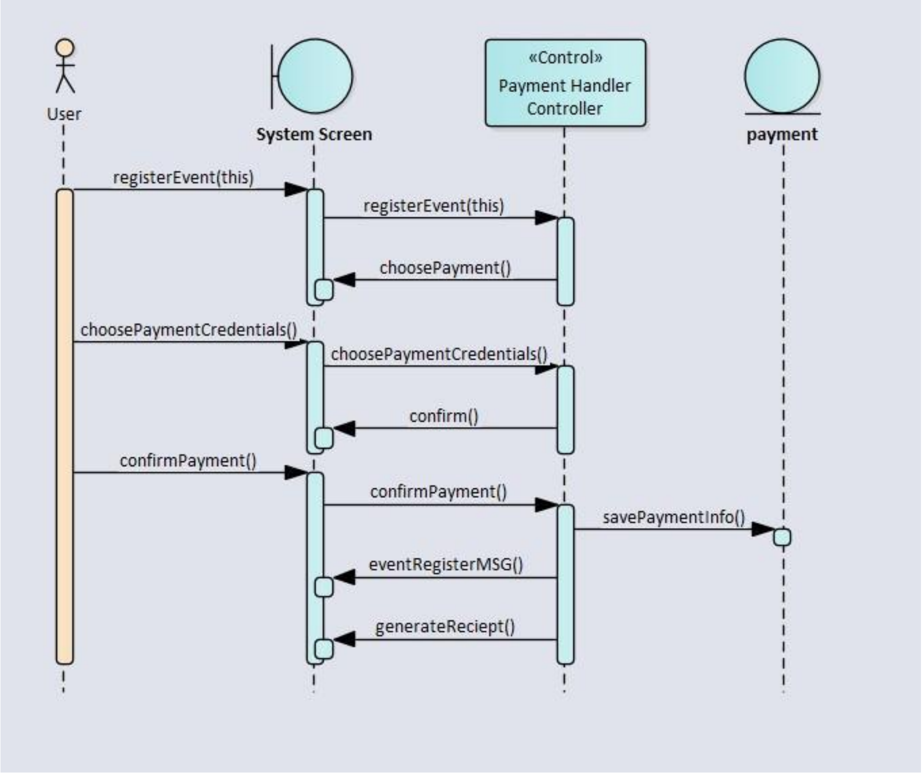


Figure 3.14: Sequence Diagram of <Payment>

c) SD003: Sequence diagram for Generate Reciept

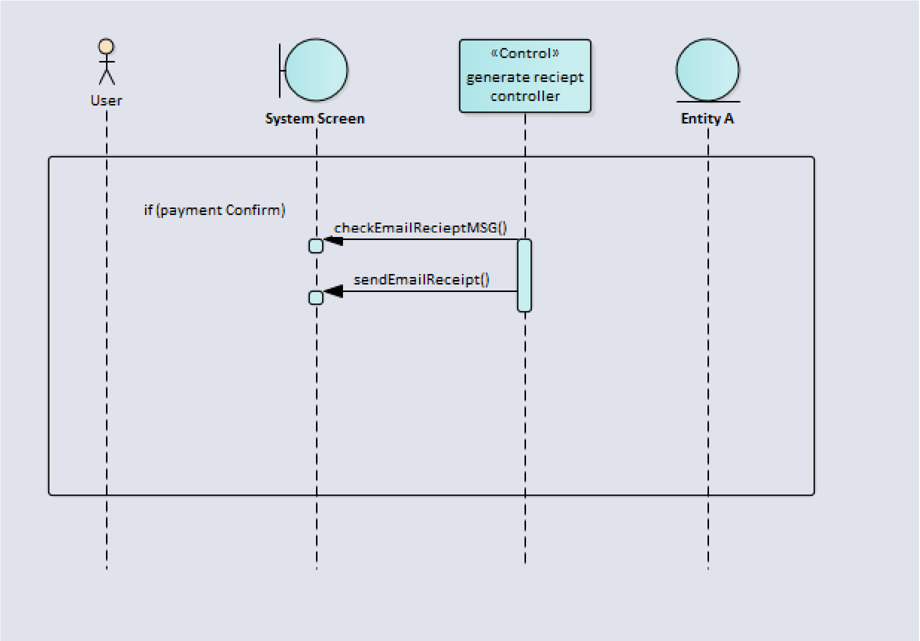


Figure 3.15: Sequence Diagram of <Generate Reciept>

4. Data Design

4.1 Data Description

the online event management system consists of 7 tables that store information about the (events) , (users) , (payment card) , (address) , (emails), (event payment) , (event registration) in which each one of those tables have connection with at least another table through the (foriegn key) and each table has their own (primary key) to mark a spcefic record.

4.2 Data Dictionary

Payment card Table

Attribute Name	Key Type	Data Type	Description
UserID	Foriegn Key	String	UserID contains the ID of the user which is primary key in it's own table that refers to the User who is the holder of that payment card
PaymentCardID	Primary Key	int	payment card id is the ID of the card that is unique and acts as an identifier for the card
CardName	Normal Key	String	card name is the name of the service that offers that payment card
CardNumber	Normal Key	int	card number is the number on the card that acts a serial number and not the

			password
CardPin	Normal Key	int	card pin is the password of the payment card and cannot be used unless the pin number is correct
CardBalance	Normal Key	double	card balance is the amount of balance that is left in the payment card

Table 4.1 : Data Dictionary for Table <Payment Card>

User Table

Attribute Name	**Key Type**	**Data Type**	**Description**
UserID	Primary Key	String	UserID contains the ID of the user which is primary key that refers to the specific User
FullName	Normal Key	String	FullName is ful name of the user of the system
UserName	Normal Key	String	UserName is the username of the user that shall be displayed while the user is on the system

Birthdate	Normal Key	String	Birthdate is the Birthay of the user to make sure he/she is above

			18

Table 4.2 : Data Dictionary for Table <User>

Address Table

Attribute Name	Key Type	Data Type	Description
UserID	Foriegn Key	String	UserID contains the ID of the user which is primary key in it's own table that refers to the User who is living in that address
City	Normal Key	String	City is the city in which the user is currently situated in.
Street	Normal Key	String	Street is the street that the user is currently in
Zip	Normal Key	String	Zip is the zipcode of the user's city or state that he/she lives in

Table 4.3 : Data Dictionary for Table <Address>

Email Table

Attribute Name	Key Type	Data Type	Description
UserID	Foriegn Key	String	UserID contains the ID of the user which is primary key in it's own table that refers to the User whom the email belong to
ID	Primary Key	String	ID is the ID of the email address to have a unique identifier so no one can duplicate emails
EmailAddr	Normal Key	String	EmailAddr is the email address of the user of the system
Password	Normal Key	String	Password is the password that the user has chosen for their account

Table 4.4 : Data Dictionary for Table <Email>

Event Table

Address Name	**Key Type**	**Data Type**	**Description**
EventID	Primary Key	String	EventID is the unique event ID to indicate the Event details of that event
EventName	Normal Key	String	EventName is the name of the event
EventDescription	Normal Key	String	EventDescription is the description of the event along with the necessary information
EventStartDate	Normal Key	String	EventStartDate is the start date of the event
EventEndDate	Normal Key	String	EventEndDate is the End date of the event
EventCategory	Normal Key	String	EventCategory is the category of the events
PaymentType	Normal Key	String	Payment Type is the payment type of the event whether free or paid

Table 4.5 : Data Dictionary for Table <Event>

EventPayment Table

Address Name	Key Type	Data Type	Description
EventPaymentID	Primary Key	String	EventPaymentID is the id of the payment that has been made for the event
UserID	Foriegn Key	String	UserID contains the ID of the user which is primary key in it's own table that refers to the User who made the payment
EventID	Foriegn Key	String	EventID is the unique event ID to indicate the Event details of that event and it is a primary key in it's own table
PaymentCardID	Foriegn Key	int	PaymentCardID is the primary key in it's own table and refers to the id of the card that the payment was made through

Table 4.6 : Data Dictionary for Table <EventPayment>

EventRegistration Table

Address Name	Key Type	Data Type	Description
EventRegID	Primary Key	String	EventRegID is the id of the event that has been registered for a specific user
UserID	Foriegn Key	String	UserID contains the ID of the user which is primary key in it’s own table that refers to the User who registered the event
EventID	Foriegn Key	String	EventID is the unique event ID to indicate the Event details of that event and is used to find the specific event that the user has registered for

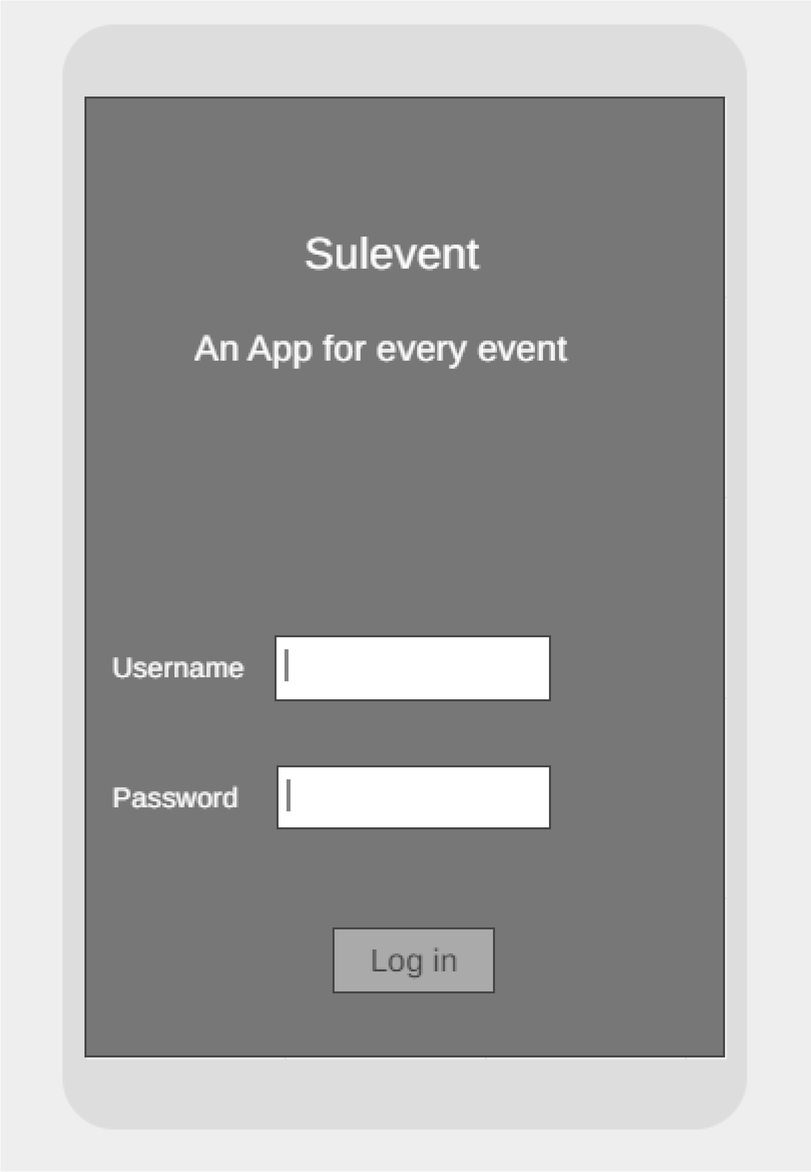
Table 4.7: Data Dictionary for Table <EventRegistration>

5. User Interface Design

5.1 Overview of User Interface

The user shall be able to use the system in a fairly easy manner in which the user is going to log in to the system and after logging it he/she will be shown all the events in the system currently labeled with categories in which the user can pick easily and the user can also create , modify , delete events by accessing the event management section through the slider menu and can click on an event to show the entire event information as well as paying for an event by choosing a payment method option and the system shall generate a receipt based on the payment if confirmed.

5.2 Screen Images



Sulevent

An App for every event

Username

Password

figure 5.3 : login structure of the system

This is the login page of the system in which the User logs in to the system by inserting username and password.

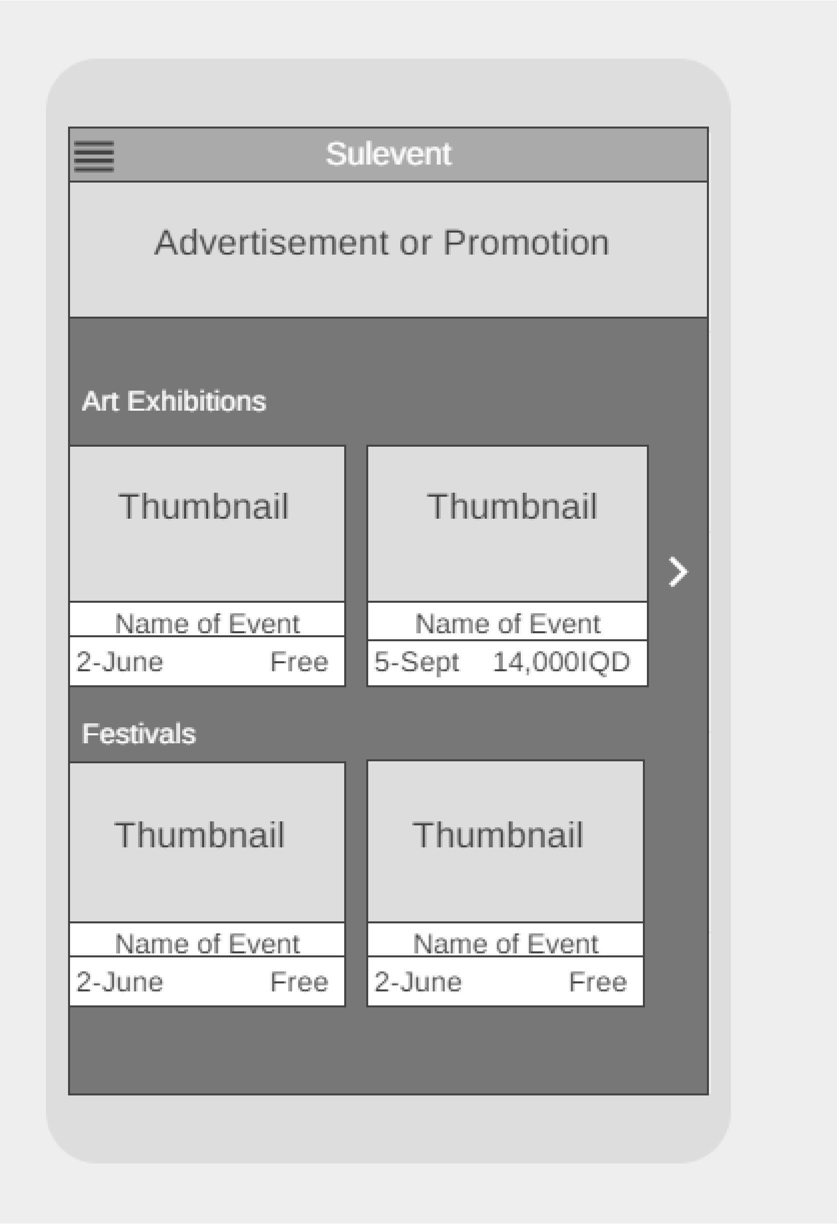


figure5.4 : the main screen after login

This is the main screen of the system after the login by the user in which all the events of the system are shown in each category in which the user can browse; they can click on a specific category on this screen or the category tab can be used to view all the categories of the system.

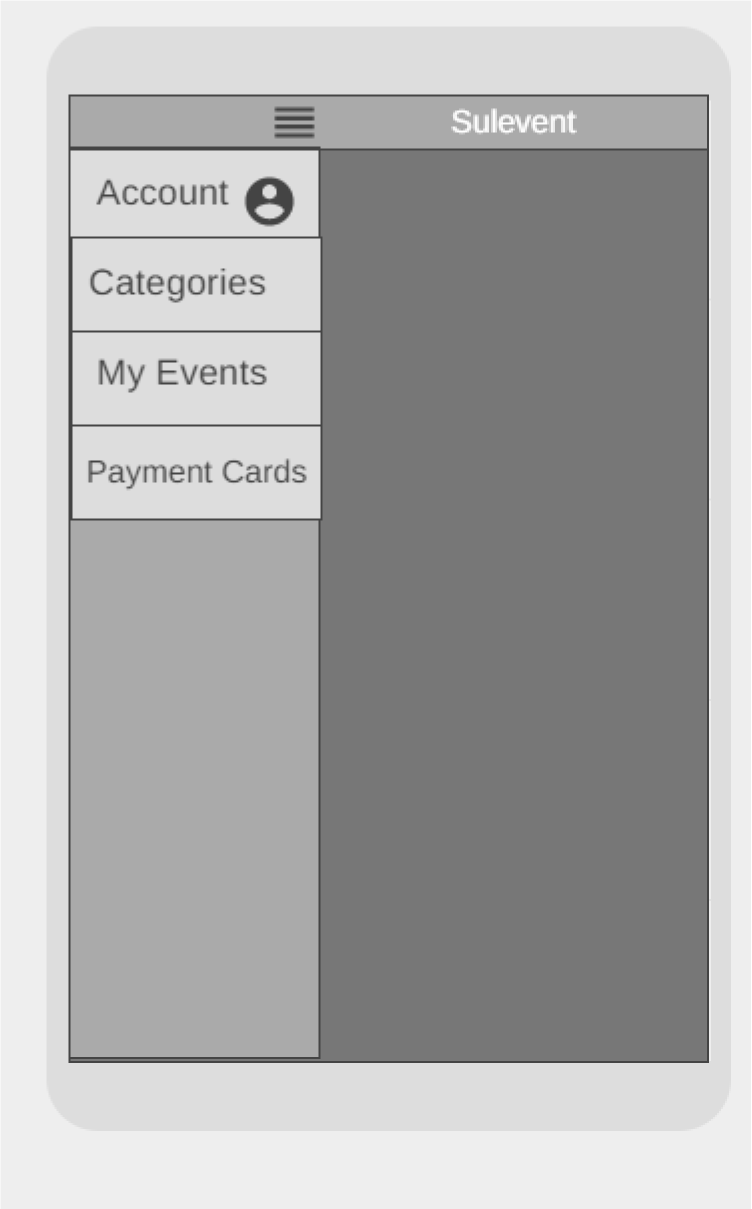


figure 5.5 : the slider that contains the options

this is the slider menu that contains the options in the system such as viewing the account of the user or viewing the categories by clicking on the category tab or show the events of the user by clicking on the my events tab , the user can also choose the payment cards and insert their information by clicking on the payment cards tab.

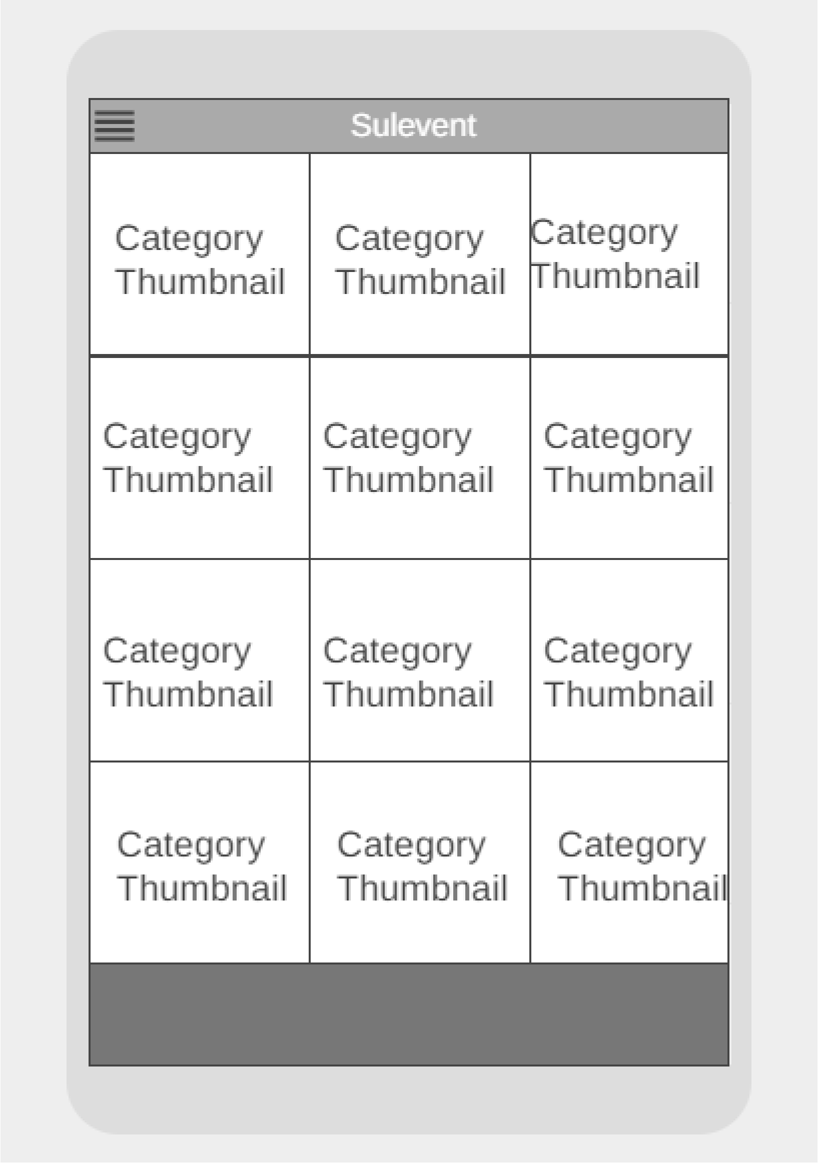
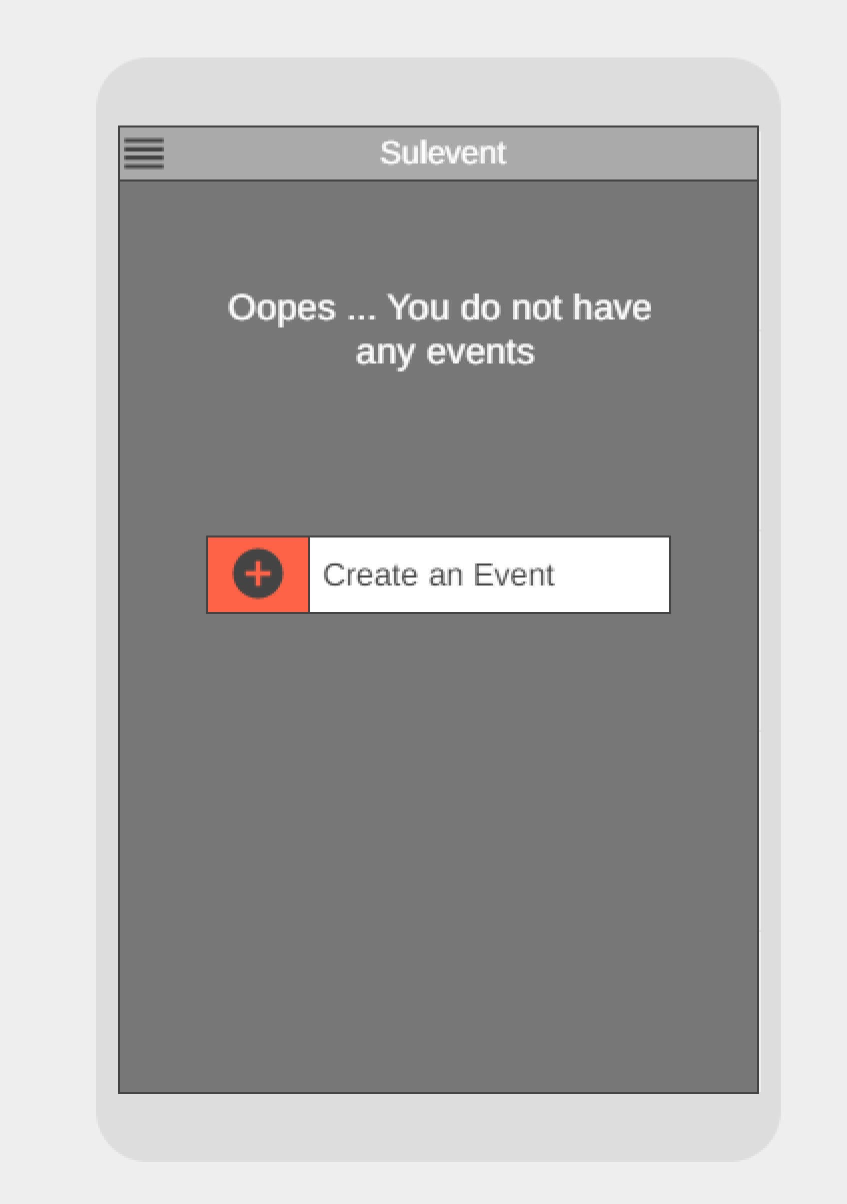


figure 5.6 : the categories

The user after clicking on the categories in the slider menu they are shown these tiles that contain a thumbnail of each category and once clicked it shows the events related to that certain category.



Oopes ... You do not have any events

figure 5.7 : create event

this menu option is shown after the my events tab has been clicked by the user of the system in which the events that the user has created are shown here but in this case there are no events for that user hence the message "Oops…You do not have any events" is shown along with the choice of creating an event in which the user is taken to the input form for the creation of the event once they click on the create an event button.

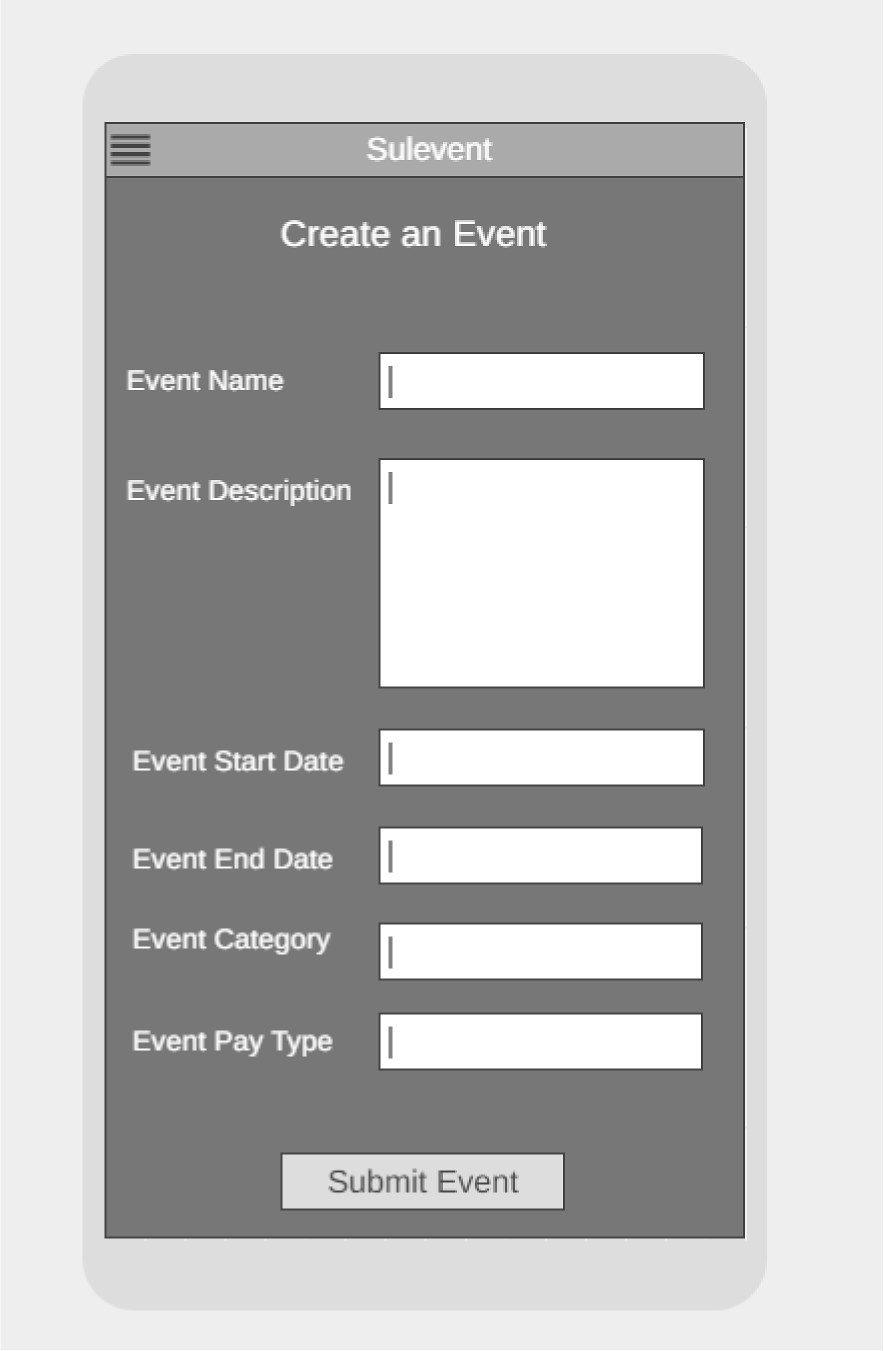


figure 5.8 : create event form

this is the input form for creating an event once the create an event button is clicked by the user it allows the user to input the name of the event , the description for the event , the event's start date , the event's end date , the event's category and the event pay type whether free or paid event. Then the submit event button is clicked in order to create the event.

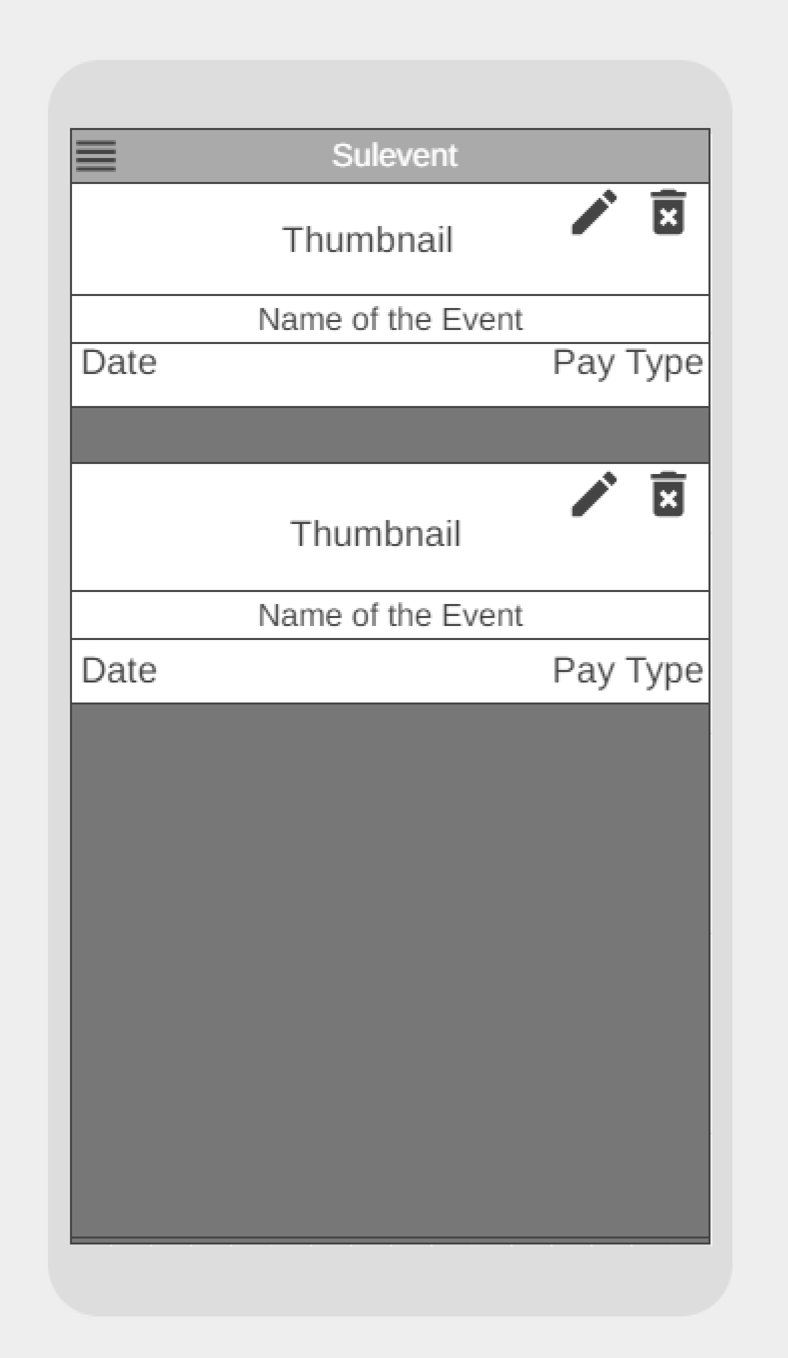


figure 5.9 : my events

This screen is shown once the user clicks on the my events tab in the slider menu in which the events of the user are shown and they are given the ability to either delete the event or to modify the existing information for that specific event.

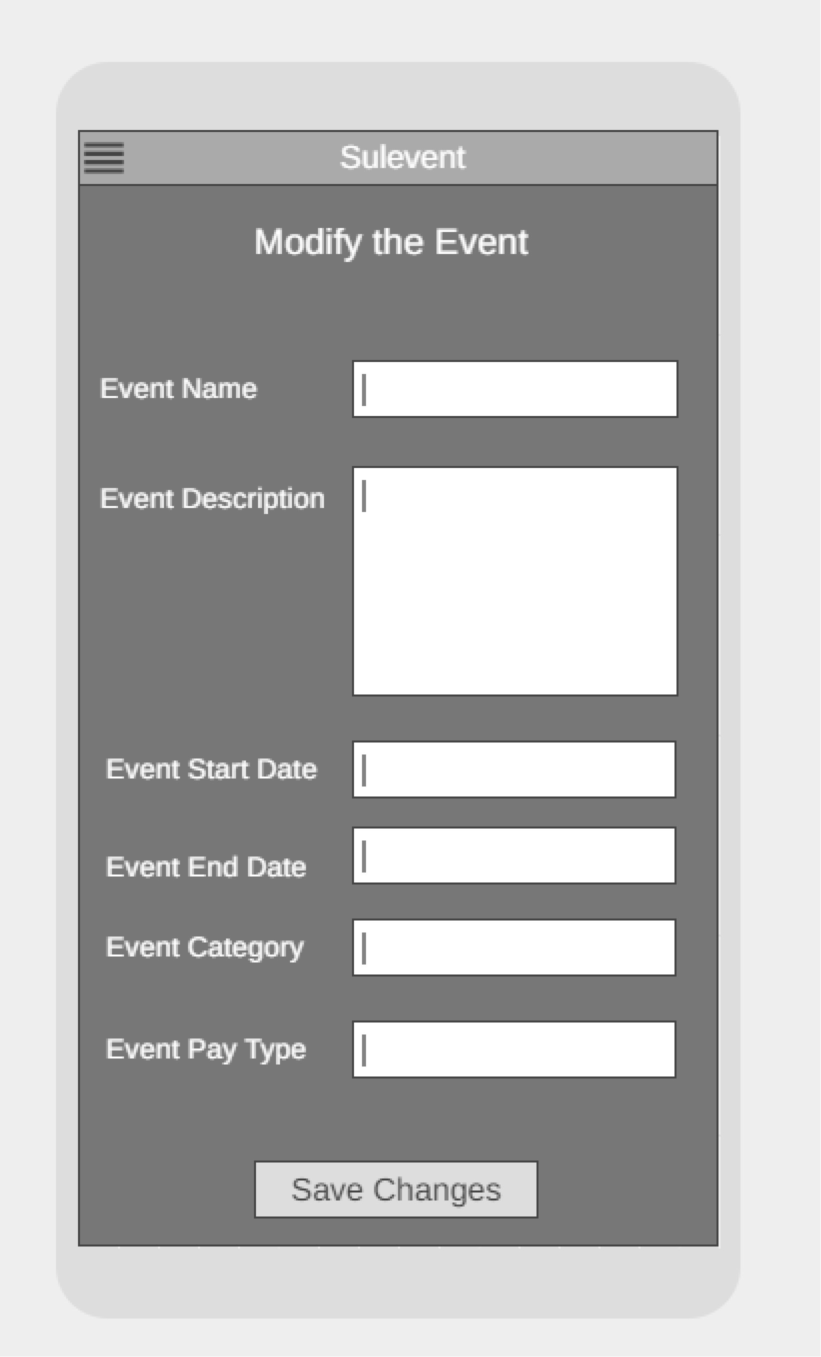


figure 5.10 : modify events

this screen is shown when the user clicks on the modification button in the my events tab to edit the information of the event that has been created, the user can then modify the event name , event description , the event start date , the event end date and the event category and the event pay type whether free or paid.

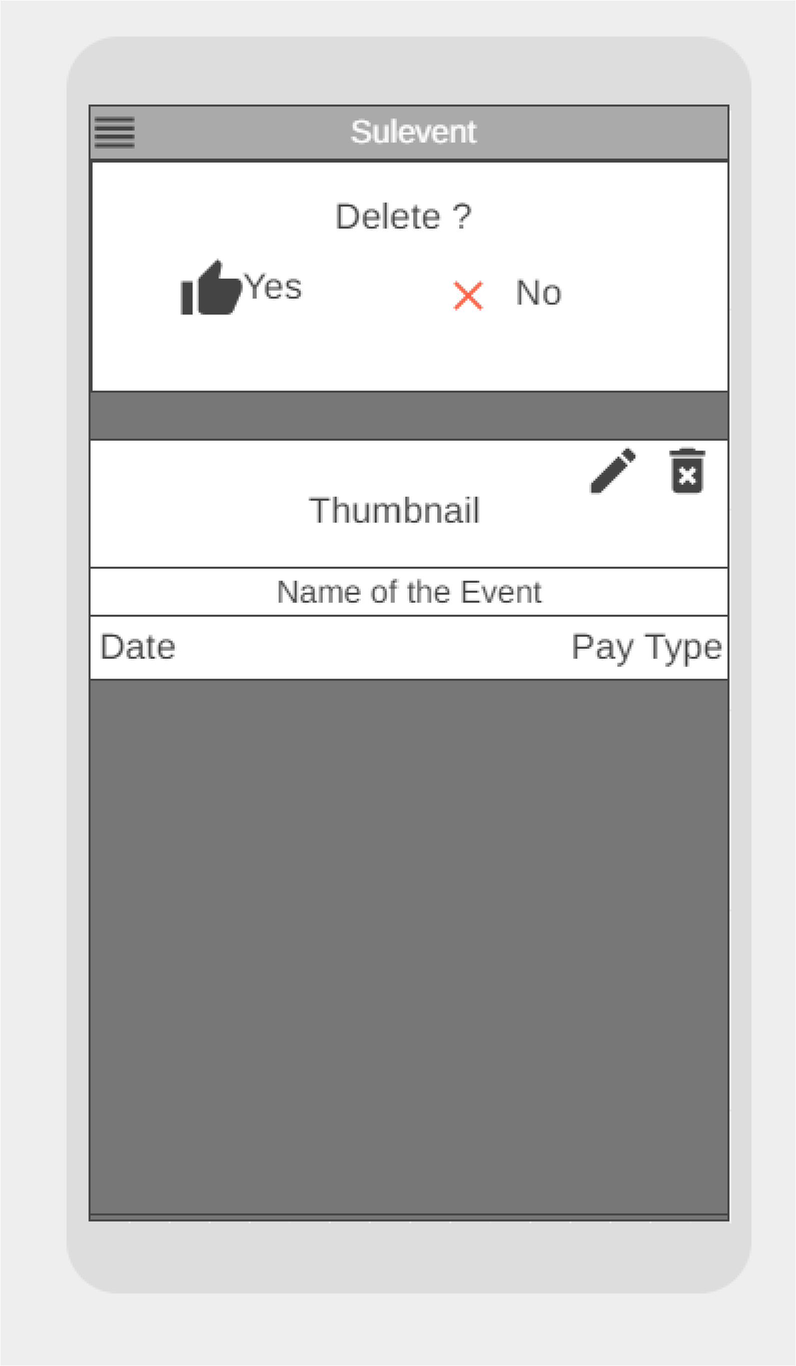


figure 5.11 : confirm delete event message

This message is shown when the delete icon is clicked and two choices are given to the user to either say no to refuse the deletion of the event or to say yes to confirm the deletion of the event.

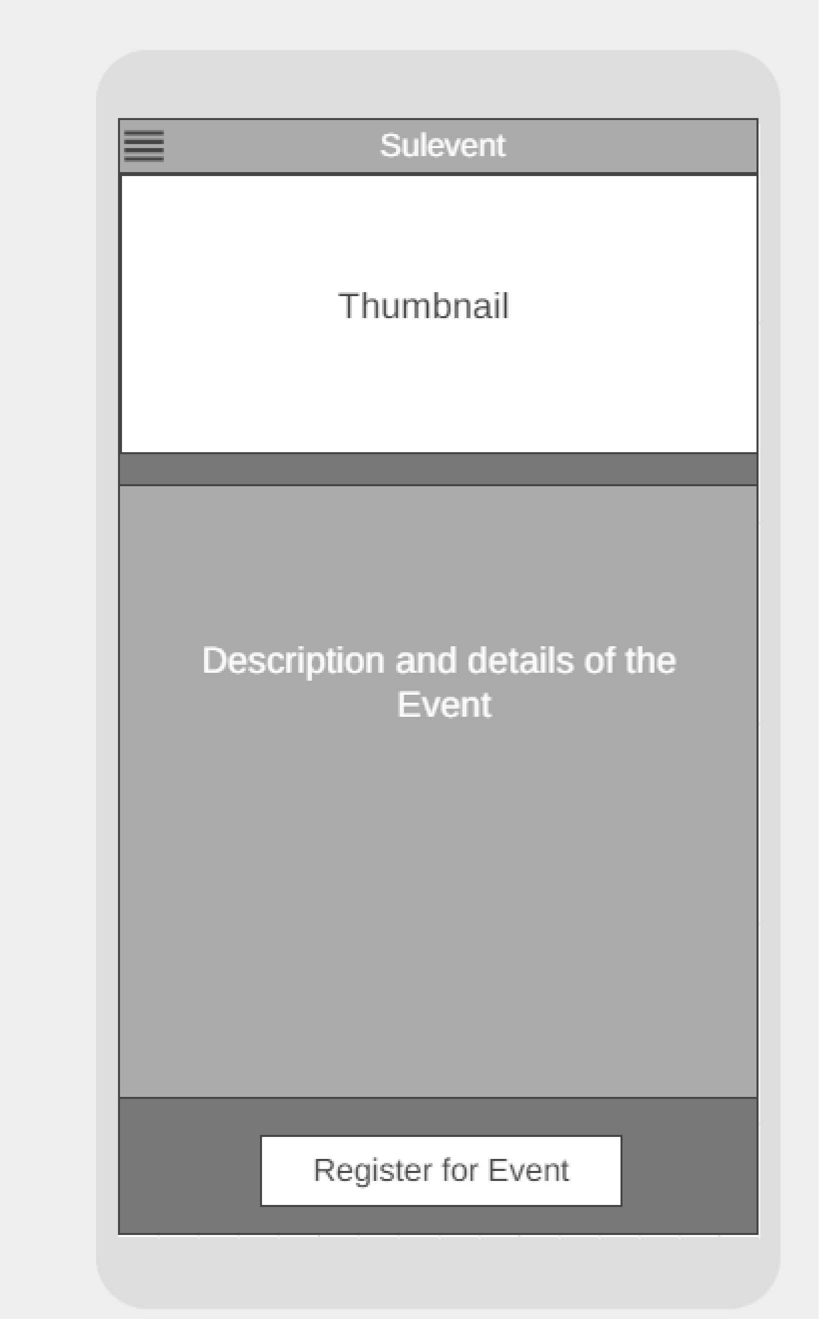


figure 5.12 : register for event

this screen is shown when the event is clicked to show the full details of the event as well as giving the user the ability to register for the event by clicking on the register for event button in which the user will be registered if the pay type of the event is free , however if not free then the user is given the choice to choose their payment card.

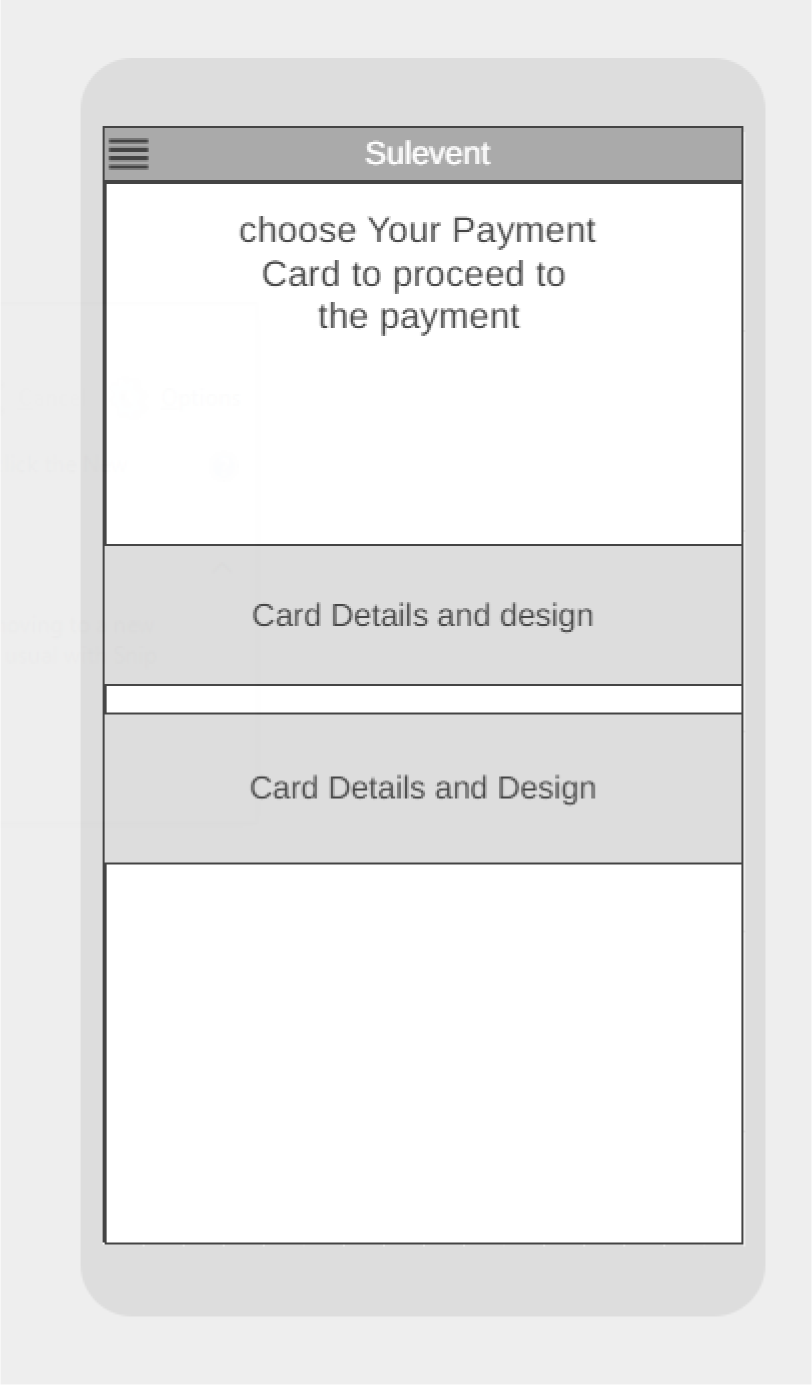


figure 5.13 : payment cards to choose from

This screen is shown to the user when the event pay type is not free and the user has clicked on register for the event , the choice is given to the user to choose their payment card to pay for the event.

6. Requirements Matrix

	P001	P002	P003
UC001		X	
UC002		X	
UC002		X	
UC004		X	
UC005			X
UC006	X		X
UC007	X		X
UC008		X	
UC009		X	

Table 6.1 : Requirements Matrix for <Online Event Management System>

7. Appendices

Provide appendices if any.