ALVA'S DEGREE COLLEGE

SUNDARI ANANDA ALVA CAMPUS, VIDYAGIRI MOODBIDRI-574 227

PROJECT REPORT

ON

"EVENTZO"

SUBMITTED TO MANGALORE UNIVERSITY, IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF DEGREE OF BACHELOR OF COMPUTER APPLICATION.

PROJECT ASSOCIATE

Shravan K S (Reg No. 161152195)

Pramod N (Reg No. 161152182)

Mohammed Afnan (Reg No. 161152168)

Under the valuable guidance of

External Guide

Mr. Haseeb

Android developer

Amitech Solution, Mangalore

Internal Guide

Ms. Abhijna A

Dept. of Computer Science

Alva's College, Moodubidri

Carried out at

Amitech Solutions

Mangalore

2018-2019

DEPARTMENT OF COMPUTER SCIENCE

ALVA'S COLLEGE

SUNDARI ANANDA ALVA CAMPUS, VIDYAGIRI MOODBIDRI-574 227

CERTIFICATE

This is to certify that the project report work titled "EVENTZO" has been carried out in **"**Amitech Solutions**"** by **Shravan K S (Reg No.161152195), Pramod N (Reg No. 161152182) and Mohammad Afnan (Reg No. 161152168).** the students of sixth semester BCA under the supervision and guidance of **Ms. Abhijna A** Department of Computer Science **Alva's College, Moodubidri.** This dissertation is submitted in partial fulfilment for the award of degree in BCA by Mangalore University during the year 2018-2019.

Internal Guide **Head of the department**

Submitted for the viva-voice examination held on _____ at Alva's College, Moodubidri.

External Examiner **Internal Examiner**

DECLARATION

We hereby declare that the project work entitled "EVENTZO" has been developed by us during December 2018- march 2019 under the guidance of **Ms. Abhijna A** (Dept. of Computer Science, Alva's college, Moodubidire) and **Mr. Haseeb** (Aimtech Solutions).

And we submit the same to the Mangalore University in partial fulfilment of requirement for the award of degree of "Bachelor of Computer Application" (BCA).

We further declare that the software developed during this project has been the result of our efforts and this software has not formed a basis for award of any other degree of any university

Place: Moodubidire

Date: 25/03/2019

By,

Shravan K S

Pramod N

Mohammed Afnan

ACKNOWLEDGEMENT

Grateful thanks are extended to all people who have made this project successful with their support and encouragement.

I express my heartly gratitude to **Prof. Dr.Kurian**, Principal of our college.

I thank **Mrs.Vanitha Prabhu**, Head of Department and my internal Guide **Ms.Abhijna.A,** Assistant Professor who has guided me and provided support and genuine replies to my queries. Her certain suggestions and interest in helping me complete the project has led me to the successful result of my project and also all the other Lectures who had been more of a support to me than ever.

I express my gratitude to my external guide **Mr. Haseeb** who has guided me in creating this project and also a great support. She has not only guided me in the project but also helped me by providing suggestions without which I couldn't improve at work. My special thanks to my dear friends who provided me their help, support and genuine feedbacks that helped me improve at my best.

Finally, I thank to my parents, who were there by me, financially supported throughout the project and provided the support I needed throughout this period.

By,

Shravan K S

Pramod N

Mohammed Afnan

TABLE OF CONTENTS

CHAPTER-1
PROJECT SYNOPSIS

PROJECT SYNOPSIS

1.1 Title of the Project:

"EVENTZO" - A media for uploading, viewing and booking the tickets for all the type of events.

1.2 INTRODUCTION:

This is the media which will connects to all the type of social events and contains information about them. It will allow the person to book tickets for a particular event hosted by another user or organizations.

In the EVENTZO application we notice the public about all the type of events. And also allow them to upload the information about the particular Event, in which they are participating or watching. EVENTZO is an application which lets the user to get the information (live messages, complete description, video link and articles etc.), about all the type of Events (Sports, Competitions, Social Events and Awarding functions etc.) which takes everywhere. Here a user can connect with another user to send and notifications to each other.

1.3 OBJECTIVE:

Our main aim is to computerize the above process and enable the public a beneficial methodology with an easy access.

- EVENTZO's aim is to provide information about all the type of social events in a single platform.
- Through the EVENTZO we can help the public to upload the Information about the all type of social event in which they are participating.
- Tickets booking for the Event hosted by another user.
- Let a person to upload the information about any kind of events in messages format for selected members.
- Providing notifications about an event for an authorised user.
- Grouping the multiple users with a particular group name so that it allows multiple users to send and receive notifications and information among the group.

1.4 PROJECT CATEGORY: Android based application (App)

1.5 HARDWARE AND SOFTWARE REQUIREMENTS:

1.5.1 Hardware requirements:

- Laptop
- Intel – core i5 processor
- 8 GB RAM
- Hard Disk

1.5.2 Software Components (Tools or Platforms):

- Android Studio
- WampServer
- Sublime Text

1.6 LANGUAGES TO BE USED:

1.6.1 Front – end:

- XML
- Java

1.6.2 Back- End:

- Server sided scripting - PHP
- MySQL

1.7 STRUCTURE OF THE PROGRAM:

1.7.1 Analysis: In the EVENTZO application we notice the public about all the type of events. And also allow them to upload the information about the particular Event in live messages format, in which they are participating or watching. We also allow them to host an event.

- User can upload information about the Events (Sports, Competitions, Social Events and Awarding functions etc.). And send notifications about them in which he is participating.
- Another user get notifications about the event (live messages, complete description, video link and articles etc.).

- User can host an Event: He needs create the event by giving complete description and tickets information for that event.
- Another user who wants to participate in that event can purchase the tickets.

1.7.2 Module description:

User Module: Each user will have an individual account. They can perform the following activities.

- Login
- Upload the information about an event (Sports, Competitions, Social Events and Awarding functions etc.).
- Provide notifications to selected users in the group.
- Receive notifications about an event from another user/organization.
- Fee Payment.
- Lodge Complaints.
- Search members or events.
- Purchase the Tickets for the Hosted event by another user.

Creation Module: This module allows the user to create an event by giving complete description about the event and the tickets if he/she is going to sell online. The following activities are performed during the creation of the event.

- Create event;
- Upload event information (Event name, type, description, image, date, time, video link etc.).
- Choose the hosting is free of cost or with cost.
- Giving tickets information to sell the tickets if the hosting is with cost.

Events View Module: This module allows the user to get the information about the events which has been created by the other members.

- **Home fragment:** allows the user to get the information about the events which is of free of cost.

- **Hosted fragment:** Allows the users to get the information about the events which is hosted by the other member in which user has to purchase the tickets for the next step.

Search Module: This module allows the user to search the events or the other members of EVENTZO and get the information about them for the next step.

- It allows the user to search the events.
- It allows the user to search the other members of the EVENTZO to add the particular user to the particular group.

Grouping Module: This module allows the particular user to create a group and add some other members of the EVENTZO to the group by the means of sending the notifications to them. The following activities are performed during the creation of the event.

- User will search other members in EVENTZO.
- User will add them to the existing group or he can create a new group and add them to it.

Send Notification Module: This module allows the particular user to send notification about particular event which he has created, user can send the notification publicly or to the particular group. The following activities are performed during the creation of the event.

- User will select the particular event.
- User will be redirected to the send notification activity where he can select the group name to which the notification should be sent.
- The notification received by the user will be in his/her Inbox.

Ticket Booking Module: This module allows the particular user to purchase the tickets for the events which have been hosted by other members of the EVENTZO or the EVENTZO itself. The following activities are performed during the creation of the event.

- User will select the particular event.

- User will be redirected to the Ticket purchasing process.
- User required to select the ticket type and have to mention the number of tickets he/she is going to purchase for the amount calculation purpose.
- User need to pay total amount to the given phone number in Google pay.
- He/she must paste the Transaction ID in the given space and then he can purchase the tickets.
- Before purchasing user need to accept the terms and conditions.

Account Module: This module allows the particular user to get his user information and edit his/her details. Other features of this module are as follows.

- **Edit account:** This feature allows the user to edit his details.
- This module also consists the information about events hosted or created by the particular user
- **Purchases**: This feature allows the user to get the information about the Tickets of the events purchased by him.
- **Delete account:** This feature allows the user to delete his/her own account in the EVENTZO.
- **Logout**: This feature allows the user to Sign out from his/her account.
- **Help & Feedback:** This feature allows the user to send his feedback to the EVENTZO.
- **Sold Tickets:** This feature allows the user to get the information about the sold tickets about the particular event created by him.

Live Information Module: This module allows the particular user to send information about the event in a message format. User can send the messages live during ongoing of the event. Other members of the EVENTZO can view these messages. The following activities are performed during the creation of the event.

- User will select the particular event.
- User will be redirected to the send messages activity where he can send the messages.
- This messages will be added to the events.

1.7.3 Data Structure:

- user
- booking
- comments
- enews
- event
- feedback
- tickets
- groups
- notice
- reaction

1.8 LIMITATIONS:

- Internet connection is must.
- Only authorized user can send notifications, upload information and get information about the particular event.
- Includes real money transaction.
- Application has a little variations on different versions of android operating systems.

1.9 FUTURE SCOPE OF THE PROJECT:

- In future, we provide scoreboards to upload the information about the particular sport.
- Bank transaction will be implemented for the transaction purpose.
- In future, we provide multiple images uploading, video uploading and live video uploading.
- Location sharing and tracing will be added in the future.
- We provide multiple themes to upload the information about an event.
- We provide themes for the scoreboards to upload the information about the sports.
- In future, we tie up with the state, national and multinational, Sports and Event hosting companies to utilize our applications for hosting.

- We can communicate with the public with the help of SMS service.
- We can implement finger print technology for authenticity.
- We can implement speech recognition for further support

CHAPTER-2

SOFTWARE REQUIREMENT SPECIFICATIONS

SOFTWARE REQUIREMENT SPECIFICATION

2.1 INTRODUCTION

Software requirement and specification is a complete description about the behaviour of the system to be developed. It includes a set of use cases that describes all the interactions that the users will have with the software. In addition to use case, the SRS also contain non-functional requirement which impose constraints on the design or implementation.

EVENTZO is the media which will connects to all the type of social events and contains information about them. It will allow the person to book tickets for a particular event hosted by another user or organizations.

2.1.1 Purpose

This project aim is to develop a platform which provides details about all type of social events and allow each member of the EVENTZO to share the details of the particular event in which he/she is participating and also allow him/her to host an event and sell the tickets of that event online. This project is to allow the each users to share the details of the event by providing complete description that is live messages, complete description, video link and articles etc. about all the type of Events like Sports, Competitions, Social Events and Awarding functions etc. which takes place in all over the world.

Through this application the public (user) can register them self. By providing all the required information and here a user can connect with another user to send and notifications to each other.

2.1.2 Scope

The scope of EVENTZO system is to provide all the event related services online. The services provided in which first one is information uploading (live messages, complete description, video link and articles etc.), second one is accessing events (about all type of events like Sports, Competitions, Social Events and Awarding functions etc.), third one is allowing the each members for selling tickets for the hosted events online.

2.1.3 Definition, Acronyms, Abbreviations

Terms	Meaning
DFD	Data Flow Diagram
ER Diagram	Entity Relation Diagram
SRS	Software Requirements Specification
PHP	Hypertext Pre-processor
DB	Data Base
XML	Extensible Markup Language
CFD	Control Flow Diagram
SQL	Structured Query Language
GUI	Graphics User Interface

2.1.4 References

Books

1] Donn Felker, **Android Application Development for Dummies**, Wiley Publishing Inc., 2011.

2] **Pankaj Jalote,** an Integrated Approach to Software Engineering**, 2nd** Edition, Narosa Publishing House, 2004

3] Luke Welling and Laura Thomson, **PHP and MySQL Web Development**, Pearson, 2010

Websites

1. http://www.youtube.com/Jerry-Banfield
2. http://www.youtube.com
3. http://www.google.com
4. https://www.linkedin.com/Online-Training/Android-App-Development

2.1.5 Overview

The developer is responsible for the development of the underlying database structure that shall serve as online uploading and downloading vehicle. The developer is also responsible for development of both the user and EVENTZO

interface for the system that shall both make use of the underlying database structure.

2.2 OVERALL DESCRIPTION

2.2.1 Product perspective

This product is an entirely new product. It is not a component of the whole system. The EVENTZO is a Social Events management system. In this system, the user will host the event, send the information about it in a continues message format and can sell the tickets for the hosted event online.

2.2.2 Product functions

The following list of function descriptions explains the major features of the EVENTZO.

a) Account registration

The registration function shall allow users to create secure accounts. The account will track the user's name, mobile number, and profile picture, date of birth, gender, email id and password.

Rationale: This provides security to the account member by setting up an account that is password protected. This also offers convenience so the user only has to enter the information listed above once and then it is stored in the account.

b) Account login

The login function shall allow account members to enter their user name and password. Once verified, users will be able to access account history and update their account information.

Rationale: this provides a method by which the users can access the restricted operations.

c) Create Event: This module allows the user to create an event by giving complete description about the event and the tickets if he/she is going to sell online. The following activities are performed during the creation of the event.

- Create event;
- Upload event information (Event name, type, description, image, date, time, video link etc.).
- Choose the hosting is free of cost or with cost.
- Giving tickets information to sell the tickets if the hosting is with cost.

d) Create group: This module allows the particular user to create a group and add some other members of the EVENTZO to the group by the means of sending the notifications to them. The following activities are performed during the creation of the event.

- User will search other members in EVENTZO.
- User will add them to the existing group or he can create a new group and add them to it.

e) Search: This module allows the user to search the events or the other members of EVENTZO and get the information about them for the next step.

- It allows the user to search the events.
- It allows the user to search the other members of the EVENTZO to add the particular user to the particular group

f) Send or receive notifications: This module allows the particular user to send notification about particular event which he has created, user can send the notification publicly or to the particular group. The following activities are performed during the creation of the event.

- User will select the particular event.
- User will be redirected to the send notification activity where he can select the group name to which the notification should be sent.
- The notification received by the user will be in his/her Inbox.

g) Ticket Booking: This module allows the particular user to purchase the tickets for the events which have been hosted by other members of the EVENTZO or the

EVENTZO itself. The following activities are performed during the creation of the event.

- User will select the particular event.
- User will be redirected to the Ticket purchasing process.
- User required to select the ticket type and have to mention the number of tickets he/she is going to purchase for the amount calculation purpose.
- User need to pay total amount to the given phone number in Google pay.
- He/she must paste the Transaction ID in the given space and then he can purchase the tickets.
- Before purchasing user need to accept the terms and conditions.

h) Payment module:

This module helps to manage all the payment process via online. The following activities are performed during the creation of the event.

- User need to pay total amount to the given phone number in Google pay.
- He/she must paste the Transaction ID in the given space and then he can purchase the tickets.
- Before purchasing user need to accept the terms and conditions.

i) Account: This module allows the particular user to get his user information and edit his/her details. This function also contains the information about the events hosted by the particular user, ticket purchases, etc.

j) Help &Feedback:

This function allows the user to send the feedback to EVENTZO and allows him to lodge complaints about the application. And also allows him to contact the staff. This module is handled by staff.

k) Purchases:

This function allows the user to get the information about the Tickets of the events purchased by him.

l) Sold Tickets:

This function allows the user to get the information about the sold tickets about the particular event created by him.

m) Live Information: This module allows the particular user to send information about the event in a message format. User can send the messages live during ongoing of the event. Other members of the EVENTZO can view these messages. The following activities are performed during the creation of the event.

- User will select the particular event.
- User will be redirected to the send messages activity where he can send the messages.
- This messages will be added to the events.

n) Account log out

The account logout module shall allow account members to exit their account for security purposes.

Rationale*:* This allows account members to exit their accounts, and prevent others from accessing it.

o) Delete account

This feature allows the user to delete his/her own account in the EVENTZO.

2.2.3 User characteristics

- Users of the website must provide required information for registration and login to the account.
- Users of the website must know how to navigate in a website.

2.2.4 Assumptions and Dependencies

Assumption

Since the EVENTZO is an application but its features is only accessible through the Internet. It is assumed that the end user has a connection to the Internet. In

addition, that the user has to download the app from the respected website or respected online stores.

The number of users and services on the EVENTZO depends on various factors like

- Power source.
- Systems (User system).
- Communication mediums (Wireless).
- Internet connections.

2.3 SPECIFIC REQIREMENTS

2.3.1 External Interface Requirements

2.3.1.1 User interfaces

The system will provide the ability for the users and the EVENTZO staff to store information via the internet. It has many user interfaces for different modules and functions.

User: The public (user) will be allowed login in order to perform any operations. These operations will include uploading the information of the particular event in which he/she is participating, hosting and selling the tickets about that event online, updating their profile and fee payment.

2.3.1.2 Hardware Interfaces

Processor	Intel(R) Core(TM) i5-8250U CPU @ 1.60Hz 1.80GHz
Hard disk	200GB recommended or higher
Memory	4GB of RAM or higher
Display	Standard Output display
Keyboard	Standard Qwerty Keyboard for Interface
Mouse	Standard mouse

2.3.1.3 Software Interfaces

Operating System	Windows XP and Above
Other technologies	WampServer, Sublime Text, Android Studio
Front End/IDE	XML, Java
Back End	PHP, MySQL server

2.3.1.4 Communication Interfaces

This is Android application system and communication is done through internet and internet protocols (TCP/TP).

2.4 FUNCTIONAL REQUIREMENTS

a) Register

- The system shall allow a non-registered user to create a secure account.
- The system shall require the following information from the user: Name, mobile no, profile picture, date of birth, gender, email id, password.
- The system shall ask the user for a username and password.
- The system shall confirm the username and password are acceptable.
- The system shall store the information in the database.

b) Login

- The system shall allow a registered user to log-in to their account.
- The system shall require a username/email and password from the user.
- The system will verify the username/email and password, and the user will be considered “logged-in”.

c) Home

- This is the main page for all the pages.
- The home page will allow the users to search the members and events which are required.

- This function will allow the user get the information about important events, subscribed events, and the events shared by the connected members etc.

e) Inbox

This page displays the notifications which has come to the user.

f) Hosted

This page will allow the user to book the tickets of the event which is hosted by the other member. This page only consists the information about the events in which tickets need to be purchased for the further step.

g) Search

This page will allow the user to search the particular event and members of the EVENTZO.

h) Account

This page consists of personal information about the user, information about the events created by the particular user and other information. And it also consists the purchased tickets. This page consists of the following information.

i. User name, user id/email, password, date of birth, gender, profile picture etc.
ii. Live events created by the particular user.
iii. Hosted events.
iv. Purchases
v. Link to the Help and feedback activity.
vi. Delete account

i) Create

This page will allow the user create an event. This page requires the description about the event which is going to be created by the particular user.

j) Groups

This page consists the number information about the groups created by the user to send notifications.

k) Purchases

This page consists the information about the tickets of particular events purchased by the user.

l) Help & Feedback

This page consists the set of methods to send complaints and feedback to the EVENTZO.

m) Hosted Events.

This page consists information about the events hosted by the user.

n) Event Description

This page consists the description about the events created by the user. Description consists of event name, image, location, time, date, video link, live information etc.

o) Live Information

This page provides the set of methods for the user to send continues messages for the particular event, and also allows the other members to view those messages.

2.5 PERFORMANCE REQUIREMENTS

The performance requirements are as follows:

- System login/logout shall take less than 10 seconds.
- Searches provides you required information within 10 seconds.
- Response taken by the system is very less with the minimum internet speed.

2.6 DESIGN CONSTRAINTS

The EVENTZO conform to the following design constraints:

- Supported only in android platform.
- App is need to be installed in the user's mobile for the service.

2.7 SYSTEM ATTRIBUTES

a) Reliability

The average time to failure shall be 30 days. In the event that a server does crash, a backup server will be up and running within the hour.

b) Availability

The services of the EVENTZO shall be available to users 24 hours a day, 7 days a week, with the exception of being down for maintenance no more than one hour a week. If the system crashes, it should be back up within one hour.

c) Security

Users will be able to access only their own personal information and not that of other users. The other process will also be handled through a secure server to ensure the protection of user's credit card and personal information.

d) Maintainability

Any updates or defect fixes shall be able to be made on server-side computers only without any patches required by the user.

e) Portability

The application can be run in only android based systems.

2.8 OTHER REQUIREMENTS

Need a system or mobile with Android operating system by the user to install the application and use the services.

CHAPTER-3
SYSTEM DESIGN

SYSTEM DESIGN

3.1. INTRODUCTION

System design is a primary phase of the software development. System design aims to identify the modules that should be in the system. The specification of these modules and how they interact with each other are the desired results. The goal of the design process is to produce a module or representation of a system which can be used later to build that system. It is a plan for the solution of the system. Design includes requirement specification and final solution for satisfying the requirements. The system design attention is given to what components can be implemented in the software is considered.

This document is the design report for the EVENTZO (A media for uploading, viewing and booking the tickets for all the type of events) system. This is mainly about 'how to do' and will help provide an insight to the whole system design and implementation of the EVENTZO system.

This software has the following three main components:

- Implementation of different users
- Management of user processes and implementation of related modules.
- Implement complaint / feedback option for users. The user can lodge a complaint on problems in the application and payment modules. And the can give feedback for the working of the system and suggest better idea for the application.
- With the help of this website the registered user can use the services provided. The services provided in which first one is information uploading (live messages, complete description, video link and articles etc.), second one is accessing events (about all type of events like Sports, Competitions, Social Events and Awarding functions etc.), third one is allowing the each members for selling tickets for the hosted events online.

This project is aimed to developing an EVENTZO system for the users. In which we notice the public about all the type of events and allow them to upload the information about the particular Event live, in which they are participating or watching and also allow them to host an event and sell tickets for that event. The entire project has been developed keeping in view of the distributed client-server computing technology, in mind. This project is to create an e-information about the services provided and computerize certain services.

3.2 APPLICABLE DOCUMENTS

The document used for preparing system design is Software Requirement Specification (SRS)

3.3 FUNCTIONAL DECOMPOSITION

3.3.1. User

The user will be allowed to lodge complaints by providing the basic information about them however; they must login in order to perform any other operations. The set of operations provided in which first one is information uploading (live messages, complete description, video link and articles etc.), second one is accessing events (about all type of events like Sports, Competitions, Social Events and Awarding functions etc.), third one is allowing the each members for selling tickets for the hosted events online.

3.4 DESCRIPTION OF PROGRAMS

3.4.1. Context flow diagram (CFD)

A context Flow diagram (CFD) in software engineering and systems engineering is a diagram that defines the boundary between the systems, or part of a system, and its environment, showing the entities that interact with it. This diagram is a high level view of a system. It is similar to a block diagram.

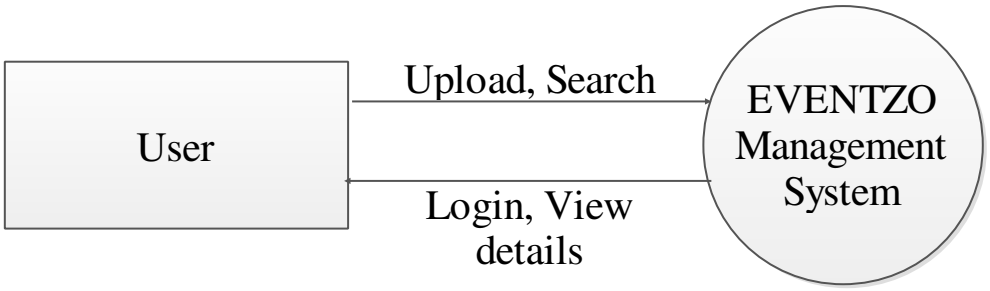


Fig 3.4.1.1. CFD for EVENTZO management System

3.4.2 Data flow diagram

A data flow diagram is a graphical representation of the data flow of a particular information system. A data flow diagram can also be used for the visualization for the data processing. It is common practice for the designer to draw a context level DFD. It shows the interactive between the system and the outside entities. The context level DFD, is then exploded to show more detailed of the system being modelled

A DFD represents flow of data of a system. Data flow diagrams are commonly used during problem analysis. It views as a function that performs the input into the desired output. DFD shows movement of data through the different transformations or processes in the system.

Data flow diagrams are also known as bubble charts DFD is a designing tool used in the top-down approach to Systems Design. This context-level DFD is next expanded, to produce a Level 1 DFD that shows some of the detail of the system being modelled. The Level 1 DFD shows how the system is divided into sub-systems (processes), each of which deals with one or more of the data flows to or from an external agent, and which together provide all of the functionality of the system as a whole. It also identifies internal data stores that must be present in order for the system to do its job, and shows the flow of data between the various parts of the system.

Notations in the DFD

Symbols	Description
	The circle or the bubble represents a process. A process is named and each process is represented by a name circle
	The source or sink is represented as a rectangular box. The source or sink is the net originator or the consumer of the data that flows in the system
	The database is represented with the open box symbol
	The arrow represents the flow of data in the system. The labelled arrow enters or leave the process or source/sink

3.4.2.1 DFD of User Module for the EVEZTZO system

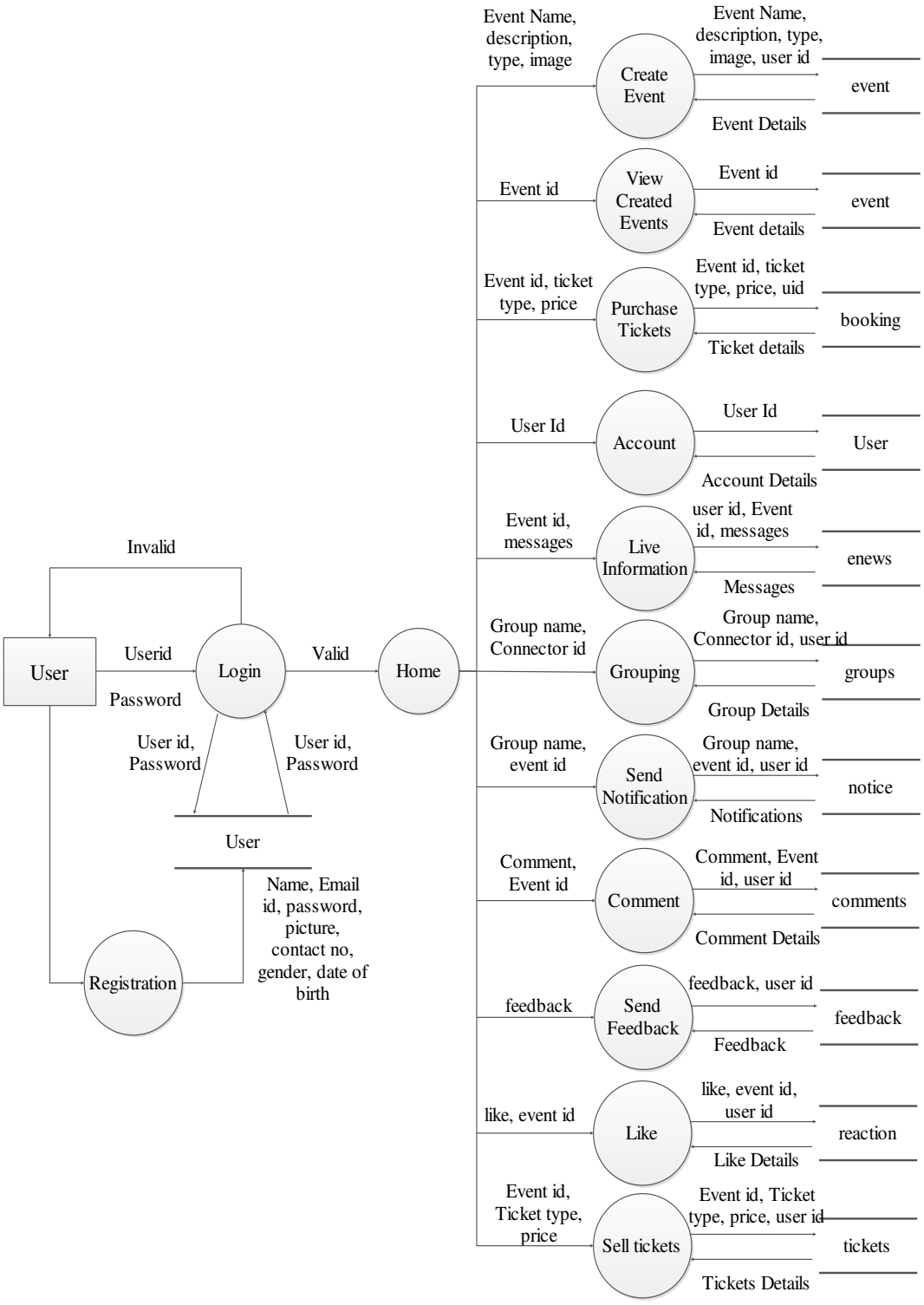


Fig 3.4.2.1 DFD of User Module for the EVEZTZO system

3.4.2.2 DFD for Creation Module

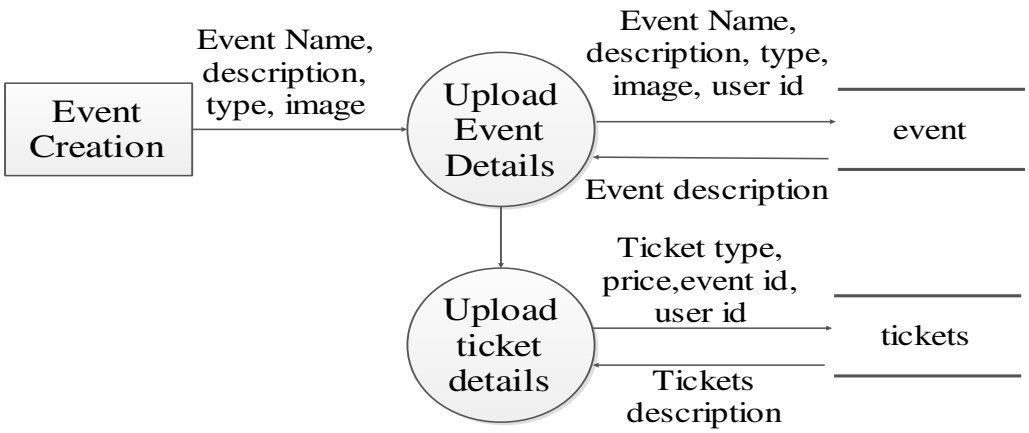


Fig 3.4.2.2 Level 1 DFD for Creation Module

3.4.2.3 DFD for Events View Module

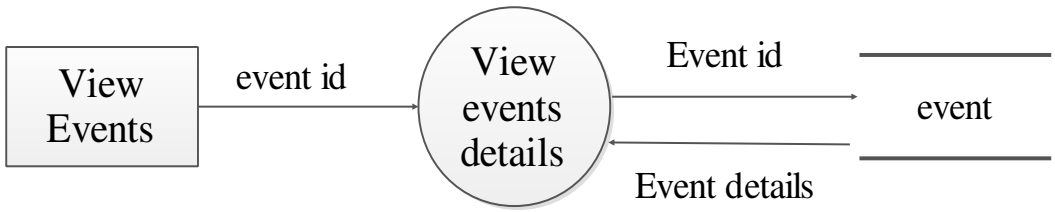


Fig 3.4.2.3 Level 1 DFD for Creation Module

3.4.2.4 DFD for Search Module

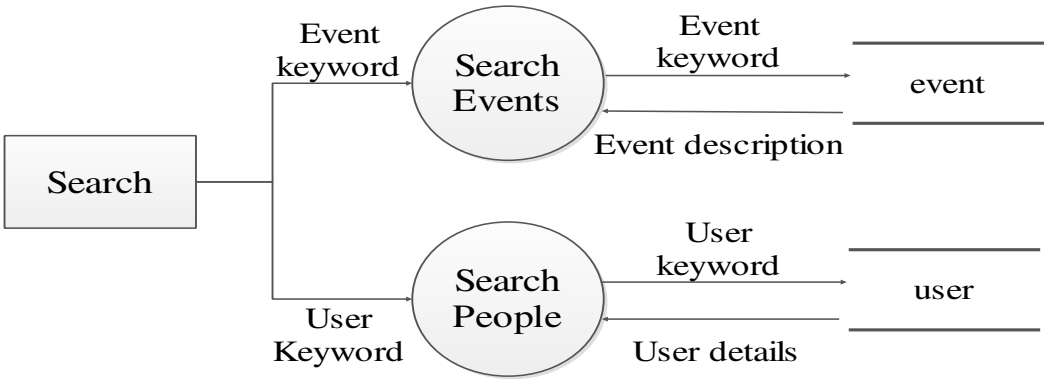


Fig 3.4.2.4 Level 1 DFD for Search Module

3.4.2.5 DFD for Grouping Module

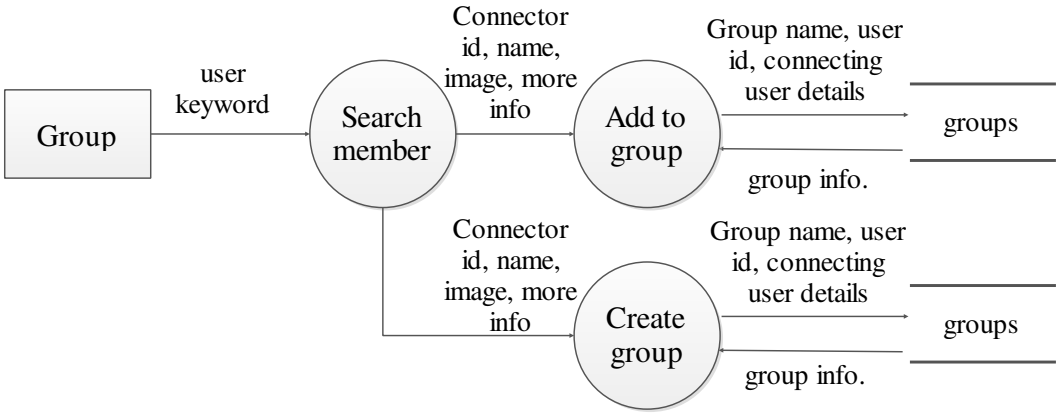


Fig 3.4.2.4 Level 1 DFD for Grouping Module

3.4.2.6 DFD for Send Notification Module

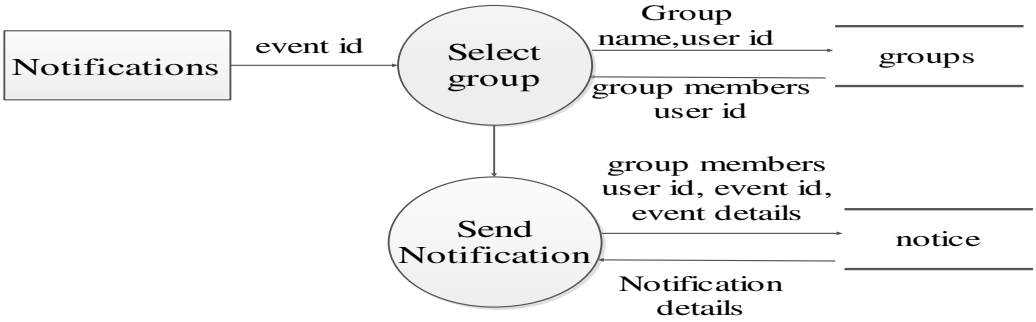


Fig 3.4.2.5 Level 1 DFD for Send Notification Module

3.4.2.7 DFD for Ticket Booking Module:

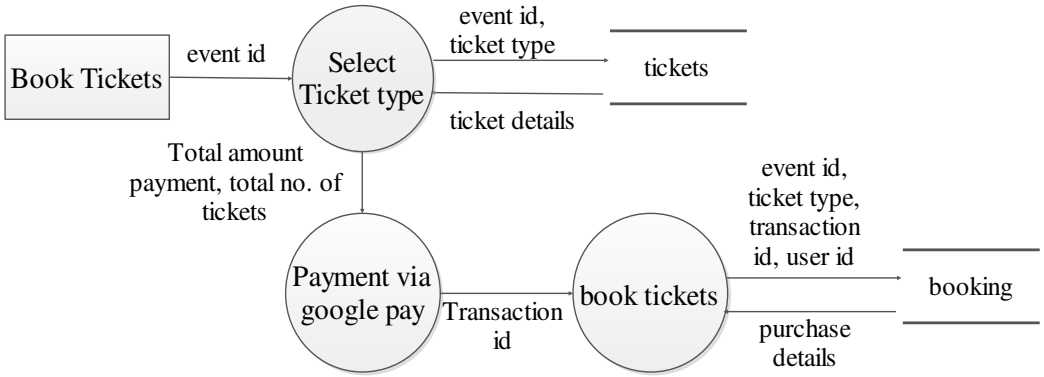


Fig 3.4.2.7 Level 1 DFD for Ticket Booking Module

3.4.2.8 DFD for Live Information Module:

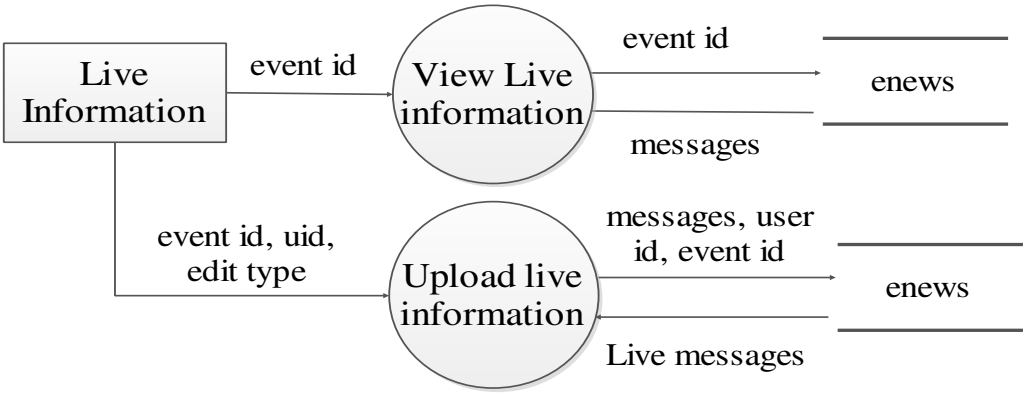


Fig 3.4.2.8 Level 1 DFD for Live Information Module:

3.4.2.9 DFD for Account Module:

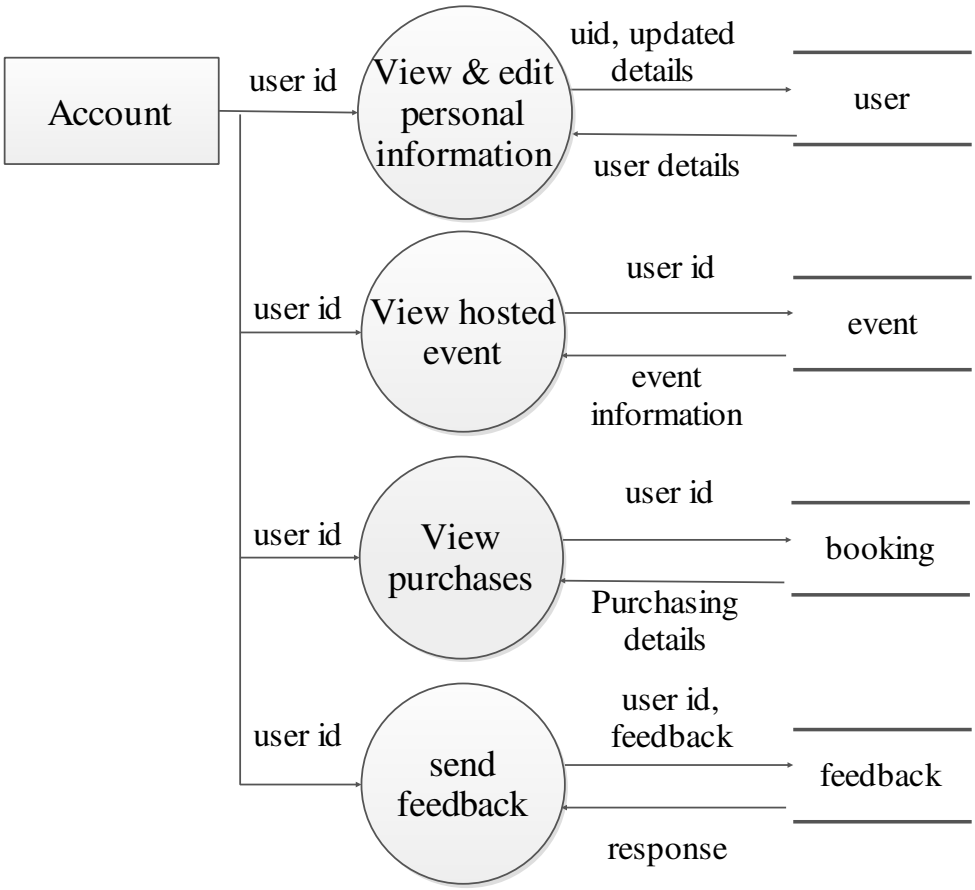


Fig 3.4.2.9 Level 1 DFD for Account Module:

3.4.3 Description of Components

a) Registration

This system allows a non-registered user to create a secure user account. This component takes a name, mobile no, email-id and password as input and will be redirected to the login page. The registered user must login after registration.

b) Login

The system shall allow a registered user to log-in to their account. The system shall require a username and password from the user. The system will verify the username and password, and the user will be "logged-in".

c) Home page

The system shall allow a registered and logged-in user to perform operations such as uploading the Event Information, accessing the uploaded events, hosting/creating the events, sharing and subscribing the events and purchasing the tickets for hosted events.

e) Inbox

This system consists of three parts one is messages, second is notifications and third is subscription. This system shall allow the user to get the notifications of the events and messages sent by connected member about the events.

c) Create Event: This module allows the user to create an event by giving complete description about the event and the tickets if he/she is going to sell online. The following activities are performed during the creation of the event.

- Create event.
- Upload event information (Event name, type, description, image, date, time, video link etc.).
- Choose the hosting is free of cost or with cost.
- Giving tickets information to sell the tickets if the hosting is with cost.

d) Create group: This module allows the particular user to create a group and add some other members of the EVENTZO to the group by the means of sending the notifications to them. The following activities are performed during the creation of the event.

- User will search other members in EVENTZO.
- User will add them to the existing group or he can create a new group and add them to it.

e) Search: This module allows the user to search the events or the other members of EVENTZO and get the information about them for the next step.

- It allows the user to search the events.
- It allows the user to search the other members of the EVENTZO to add the particular user to the particular group

f) Send or receive notifications: This module allows the particular user to send notification about particular event which he has created, user can send the notification publicly or to the particular group. The following activities are performed during the creation of the event.

- User will select the particular event.
- User will be redirected to the send notification activity where he can select the group name to which the notification should be sent.
- The notification received by the user will be in his/her Inbox.

g) Ticket Booking: This module allows the particular user to purchase the tickets for the events which have been hosted by other members of the EVENTZO or the EVENTZO itself. The following activities are performed during the creation of the event.

- User will select the particular event.
- User will be redirected to the Ticket purchasing process.
- User required to select the ticket type and have to mention the number of tickets he/she is going to purchase for the amount calculation purpose.

- User need to pay total amount to the given phone number in Google pay.
- He/she must paste the Transaction ID in the given space and then he can purchase the tickets.
- Before purchasing user need to accept the terms and conditions.

i) Account: This module allows the particular user to get his user information and edit his/her details. This function also contains the information about the events hosted by the particular user, ticket purchases, etc.

j) Help &Feedback:

This function allows the user to send the feedback to EVENTZO and allows him to lodge complaints about the application. And also allows him to contact the staff. This module is handled by staff.

k) Purchases:

This function allows the user to get the information about the Tickets of the events purchased by him.

l) Sold Tickets:

This function allows the user to get the information about the sold tickets about the particular event created by him.

m) Live Information: This module allows the particular user to send information about the event in a message format. User can send the messages live during ongoing of the event. Other members of the EVENTZO can view these messages. The following activities are performed during the creation of the event.

- User will select the particular event.
- User will be redirected to the send messages activity where he can send the messages.
- This messages will be added to the events.

n) Account log out

The account logout module shall allow account members to exit their account for security purposes.

Rationale*:* This allows account members to exit their accounts, and prevent others from accessing it.

o) Delete account

This feature allows the user to delete his/her own account in the EVENTZO.

CHAPTER-4
DATABASE DESIGN

DATABASE DESIGN

4.1. INTRODUCTION

The description of the database is called the database scheme. A database scheme is specified during database creation and is not accepted to change frequently. Most data model has certain conversations for diagrammatically displaying scheme. The scheme diagram displays only some aspects of scheme, such aspects are not specified in the scheme diagram that is neither the data type of each data item that change frequently.

The data in the database is particular moment in timer is called a database state.

Schema can be defined in the following three levels.

4.1.1. Internal level

The internal Level has an internal scheme. It describes the physical storage structure of the database. The internal schema uses a particular data model and describes the complete details of the data storage and access paths of the database.

4.1.2. Conceptual level

The conceptual level has conceptual schema, which describes the structure of whole database for a community of users. The conceptual schema hides the details of the physical storage structure and concentrates on describing entities, datatypes relationship user operation and constraints. A high-level data model or an implementation data model can be used in this model.

4.1.3. External level

External level or view includes a number of external schema or view. Each internal schema describes the part interested in and hides the rest of the database from the user group. A high level data model or an implementation model can be used at this level.

4.2. WHAT IS DATABASE?

A database is an organized collection of data. It is the collection of schemas tables, queries, reports, views, and other objects. The data are typically organized to model aspects of reality in a way that supports processes requiring information. Which can be of any size and complexity. By using the concept of database, we can easily store and retrieve the data. The major purpose of the database is to provide the information, which utilizes data that the system needs according to its own requirements.

4.3. DATABASE DESIGN

Database design is the process of producing a detailed data model of database. This data model contains all the needed logical and physical design choices and physical storage parameters needed to generate a design in a data definition language, which can then be used to create a database. A fully attributed data model contains detailed attributes for each entity.

The term database design can be used to describe many different parts of the design of an overall database system. Principally, and most correctly, it can be thought of as the logical design of the base data structures used to store the data. In the relational model, these are the tables and views. In an object database, the entities and relationships map directly to object classes and named relationships. However, the term database design could also be used to apply to the overall process of designing, not just the base data structures, but also the forms and queries used as part of the overall database application within the database management system (DBMS)

4.4. TABLE STRUCTURE

4.4.1. Table name: user

Field name	Data type	Size	Constraints	Description
uid	int	11	Not null. pk	User id
fname	varchar	30	Not null	First name

lname	varchar	30	Not null	Last name
email	varchar	40	Not null	Email id
gender	varchar	10	Not null	Gender of user
phone	bigint	20	Not null	Phone number
dob	Varchar	20	Not null	Date of birth
password	Varchar	15	Not null	Password of the user
user_image	Varchar	500	Not null	Image of user
moreinfo	Longtext	---	null	Info about event
status	int	1	Not null	status

4.4.2. Table name: event

Field name	Data type	Size	Constraints	Description
event_id	int	11	Not null, pk	event id
user_id	int	11	Not null,	User id
event name	Varchar	40	Not null	Event name
description	Text	---	Not null	Description of event
synposis	Text	---	Not null	Basic description of event
type	Varchar	20	Not null	Type of the event
date	Date	---	Not null	Date of event
time	Time	---	Not null	Time
location	Text	---	Not null	Location of event
image	Varchar	500	Not null	image
status	Varchar	10	Not null	status
video	Text	---	Not null	Video
hosttype	Varchar	20	Not null	Type of the event creation

4.4.3. Table name: ticket

Field name	Data type	Size	Constraints	Description
tid	int	11	Not null ,Pk	Ticket id
uid	int	11	Not null	User id
event id	int	11	Not null	Event id
type	varchar	30	Not null	Type of ticket
price	int	10	Not null	Price of ticket
nooftrickets	int	10	Not null	Number of the ticket

4.4.4. Table name: booking

Field name	Data type	Size	Constraints	Description
bid	int	11	Not null,pk	Booking id
event id	int	11	Not null	Event id
user id	int	11	Not null	User id
transactrionid	Varchar	20	Not null	Transaction id
event name	Varchar	50	Not null	Event Name
ticket type	Varchar	50	Not null	Ticket type
noofticket	int	11	Not null	No of ticket
totalamount	int	10	Not null	Total amount
fname	Varchar	60	Not null	First name
lname	Varchar	20	Not null	Last name
phone	int	10	Not Null	Phone number
email	Varchar	50	Not null	Email id
status	int	1	Not null	Status

4.4.5. Table name: enews

Field name	Data type	Size	Constraints	Description

messageid	int	20	Not null, pk	Id of message
event id	int	20	Not null	Id of event
user id	int	20	Not null	Id of user
messeages	Text	---	Not null	Information of event

4.4.6. Table name: groups

Field name	Data type	Size	Constraints	Description
uid	int	20	Not null	User id
connectedid	int	20	Not null	Connected id
gropname	Text	---	Not null	Group name
fname	Varchar	30	Not null	First name
lname	Varchar	30	Not null	Last name
image	Varchar	500	Not null	Image
gender	Varchar	10	Not null	Gender of the user
dob	Varchar	20	Not null	Date of the birth
moreinfo	Text	---	Not null	More information

4.4.7. Table name: notice

Field name	Data type	Size	Constraints	Description
notificationid	int	20	Not null., pk	Notification id
uid	int	20	Not null	User id
grouponame	Text	---	Not null	group name
event_id	int	11	Not null	Event id
event name	Varchar	50	Not null	Event name
event image	Varchar	500	Not null	Event images
synopsis	Varchar	500	Not null	Basic description
type	Varchar	20	Not null	Type of sending notification

4.4.8 Table name: comments

Field name	Data type	Size	Constraints	Description
c_id	int	11	Not null, pk	comment id
user id	int	11	Not null	Name of the user
event id	int	11	Not null	Id of the event
comments	Varchar	100	Not null	comment
fname	Varchar	30	Not null	First name
lname	Varchar	30	Not null	Last name
image	Varchar	500	Not null	Image

4.4.9. Table name: feedback

Field name	Data type	Size	Constraints	Description
fid	int	11	Not null, pk	Feedback id
uid	int	11	Not null	User id
feedback	Text	---	Not null	feedback
status	int	1	Not null	Status

4.4.10. Table name: reaction

Field name	Data type	Size	Constraints	Description
rid	int	20	Not null, pk	Reaction id
event_id	int	20	Not null.	Event id
uid	int	20	Not null	User id

4.5. ENTITY RELATIONSHIP DIAGRAM

Entity relationship diagram is used in modern database software. Software **engineering is to illustrate logical structure of database. It is a relational schema** database, modelling method, used to model a system and approach. This

approach is commonly used in database design. The diagram created using this are called entity relationship diagram.

The ER diagram depicts the various relationship among entities, considering each object as an entity. Relationship depicts the relationship between data objects. The ERD is the notation that is used to conduct the data modelling activity.

Entity

Entity is a thing, which we want to store information. It is an elementary basic building block of storing information about business process. An entity represents an object desired within the information system about which you want to store information.

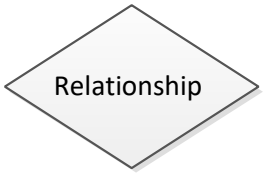
Relationship

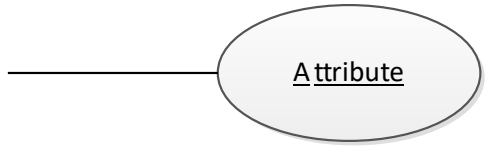
A relationship is a named connection, associated between entities, or used to relate two or more entities with some common attributes or meaningful interaction between the object.

Attributes

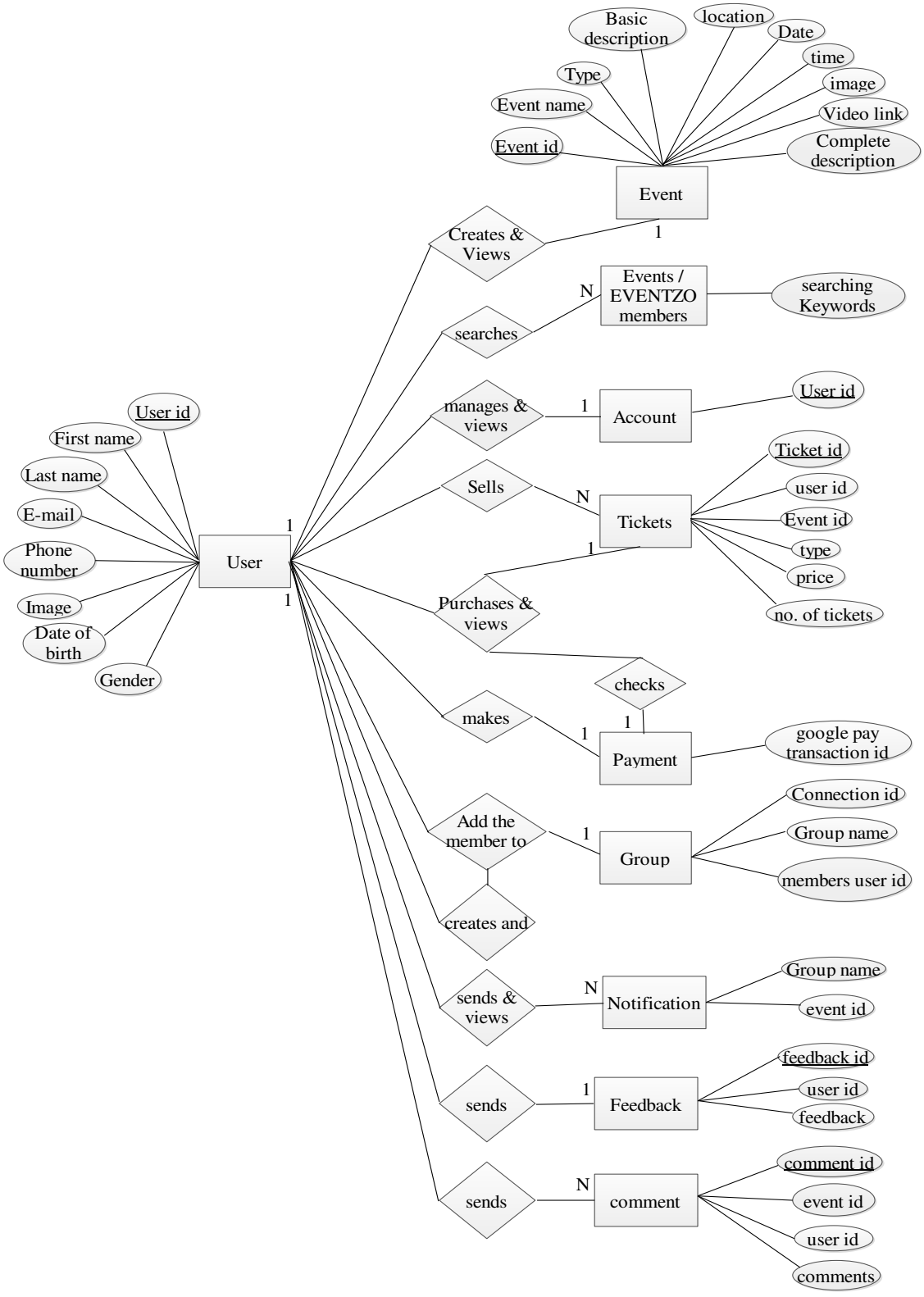
Attributes are the properties of entities and relationship. Description of the entity. Attributes are elementary pieces of information attached to an entity.

Notations and Meaning of ER-Diagram

Notation	Description
	The rectangle represents Entity
	The diamond shape represents the relationship between the two entity
	The oval shape represents the attributes

	The oval shape with underline represents the key-attributes

4.5.1 ER Diagram:



CHAPTER-5
DETAILED DESIGN

DETAILED DESIGN

5.1. INTRODUCTION

This document is the design report for EVENTZO system. This is mainly about 'How to do' and will help provide an insight to the whole system design and implementation of the EVENTZO.

This software has the following main components

- Implement the different users
- Implement the event accessing – The user can access any type of social event which is uploaded.
- Implement the event creation - The user can upload the information about any kind of social event in which he/she is participating.
- Implement the event hosting – The user can create or host any kind of social event and provide the notifications about that.
- Implement the ticket selling – The user is allowed to sell the tickets online for the particular event hosted by them.
- Implement the group creation – The user who has created the event is allowed to create a group and add the members to it, this group is used to send the notifications about a particular event to the connected members of the group.
- Implement the Live information – The user is allowed to share the information about any kind of the social in a continues message format so that other users can get the information about it.
- Implement the search – The user can search the events or the other member by specifying a keyword about them.
- Implement the inbox – The user is allowed to get the notifications.

This project is aimed to develop an EVENTZO system, a media for uploading accessing and booking the tickets for the particular event. The entire project is developed by keeping in view of distributed client server computing technology, in mind. This project is to create an e-Information about the events. Through this application, any person who wants to send information about the particular social

event can upload the information and send the notifications about publicly or to a selected members and also he can host a social event and sell tickets for that hosted event.

5.2. APPLICABLE DOCUMENT

The Detailed Design refines the System Design document hence the first applicable document here is system design. We also refer the data structures. Hence, the second applicable document is database design.

5.3. STRUCTURE OF THE SOFTWARE PACKAGE

5.3.1. User

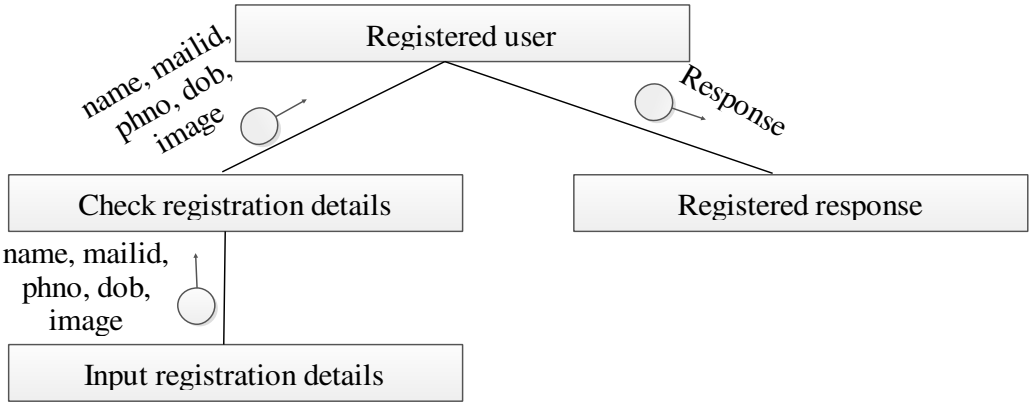
Each user will have an individual account. They can perform the following activities.

- Login
- Upload the information about an event (Sports, Competitions, Social Events and Awarding functions etc.).
- Provide notifications to connected and selected users.
- Receive notifications about an event from another user/organization.
- Fee Payment.
- Lodge Complaints.
- Search members or events.
- Purchase the Tickets for the Hosted event by another user.
- Sell the tickets for the created event.

5.4. MODULAR DECOMPOSITION OF THE COMPONENTS

5.4.1. Registration

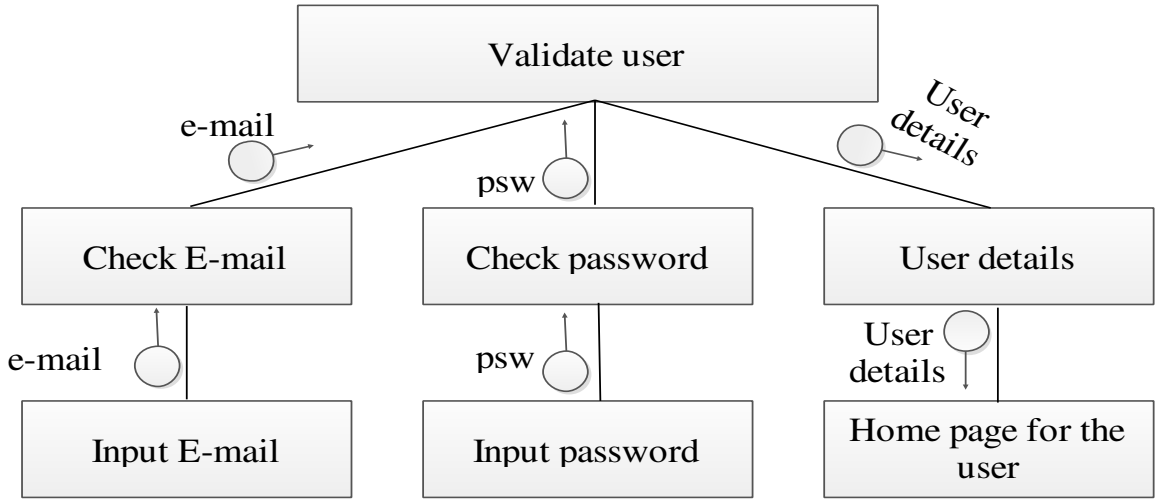
This system allows a non-registered user to create a secure user account. This component takes a name, AADHAR number, mobile no, email-id and password as input and will redirected to the log in page after registration. The registered user must login after registration.



5.4.1. Structured chart for registration.

5.4.2. Login

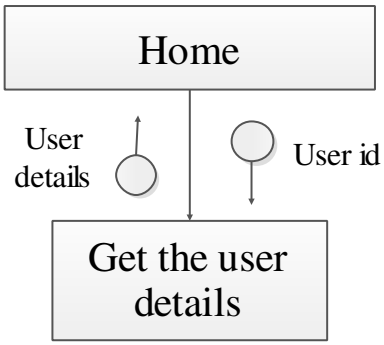
The system shall allow a registered user to login to their account. The system shall require a username and password from the user. The system will verify the username and password, and the user will be considered "logged-in".



5.4.2. Structured chart for login.

5.4.3. Home

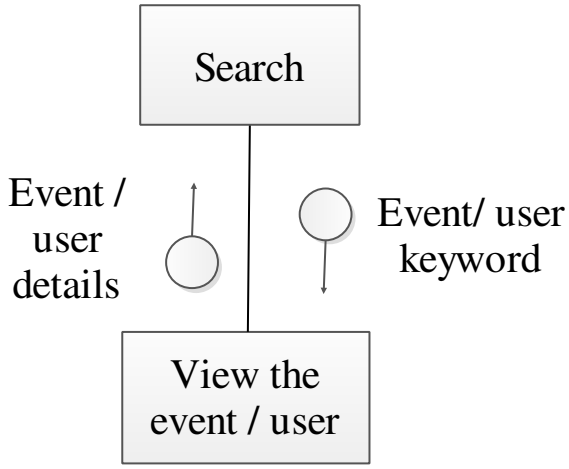
This home page is the main page, this will allow the user to get the personal account administration options and event information.



5.4.3. Structured chart for home.

5.4.4 Search

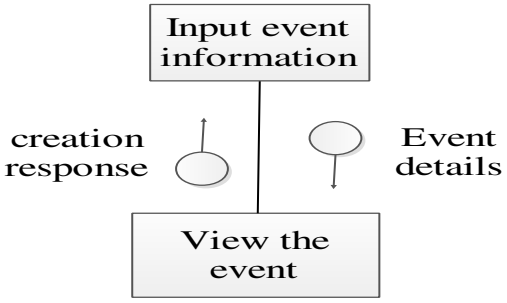
This section allows the EVENTZO members to Search the social event or any other members of EVENTZO for further processes



5.4.4. Structured chart for Search

5.4.5 Create events

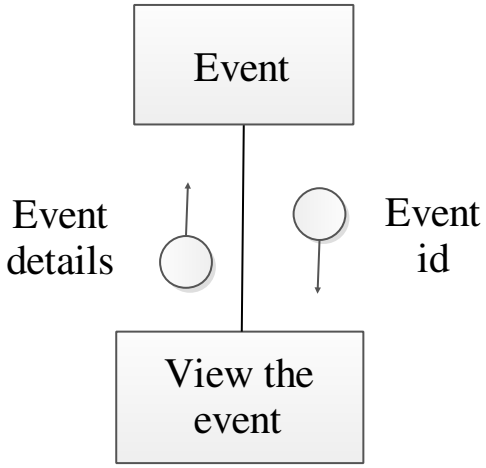
This section allows the EVENTZO members to create the social event in which he is participating, or hosting the event.



5.4.5. Structured chart for Create events

5.4.6. View events

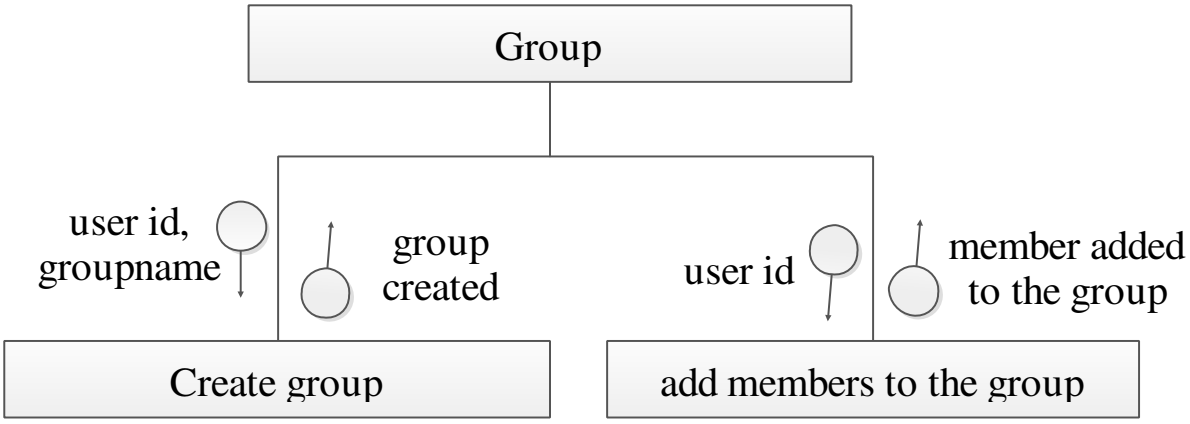
This section allows the EVENTZO members to view the information about the social event which has been created by the other members of the EVENTZO.



5.4.6. Structured chart for View events

5.4.7 Creating groups;

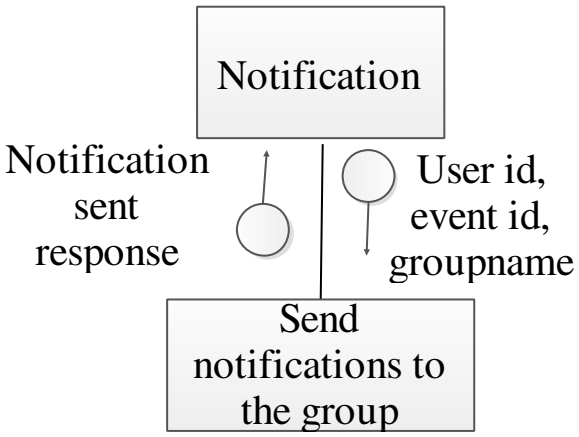
This section allows the EVENTZO members to create a group by adding other members to the group for the purpose of sharing the notification about an event.



5.4.7. Structured chart for creating groups

5.4.8 Send Notification

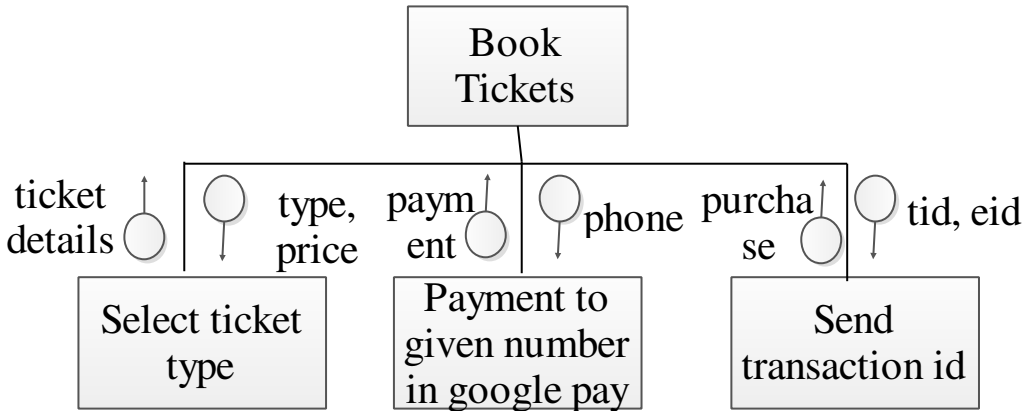
This section allows the EVENTZO members to send the notification about an event to a group of members.



5.4.8. Structured chart for Send notification

5.4.9 Ticket booking

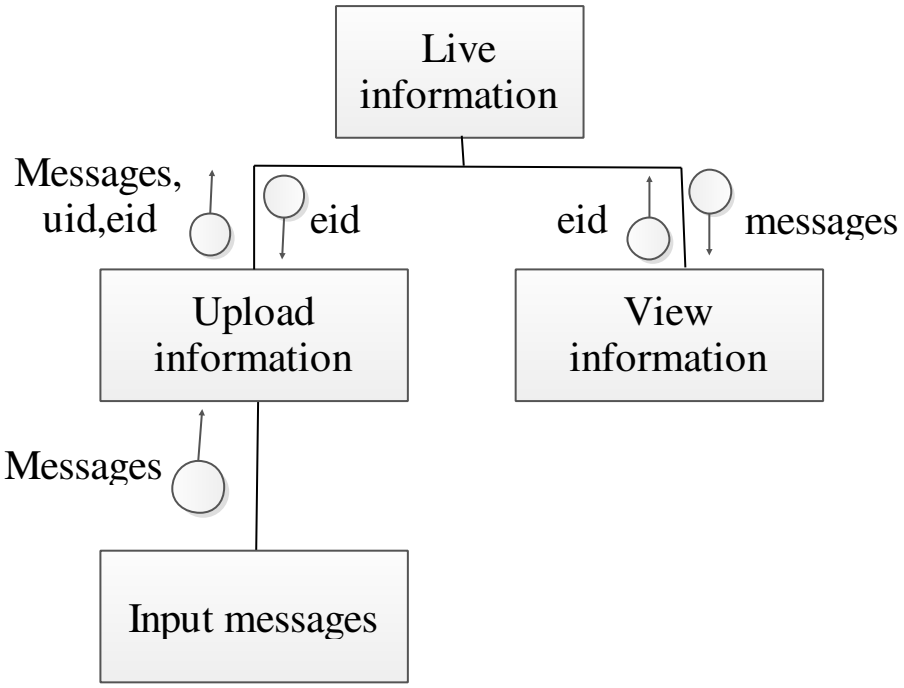
This section allows the EVENTZO members to purchase the tickets for the events hosted by other users.



5.4.9. Structured chart for Ticket booking

5.4.10. Live information

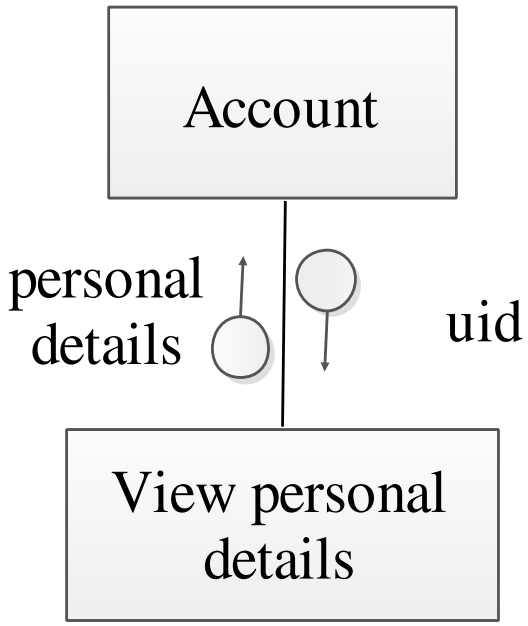
This section allows the EVENTZO members to send the live information about an event in a message format. And also view the information uploading by other members of the EVENTZO.



5.4.10. Structured chart for Live information

5.4.11 Account information

This section allows the user to view and edit the personal details of him.

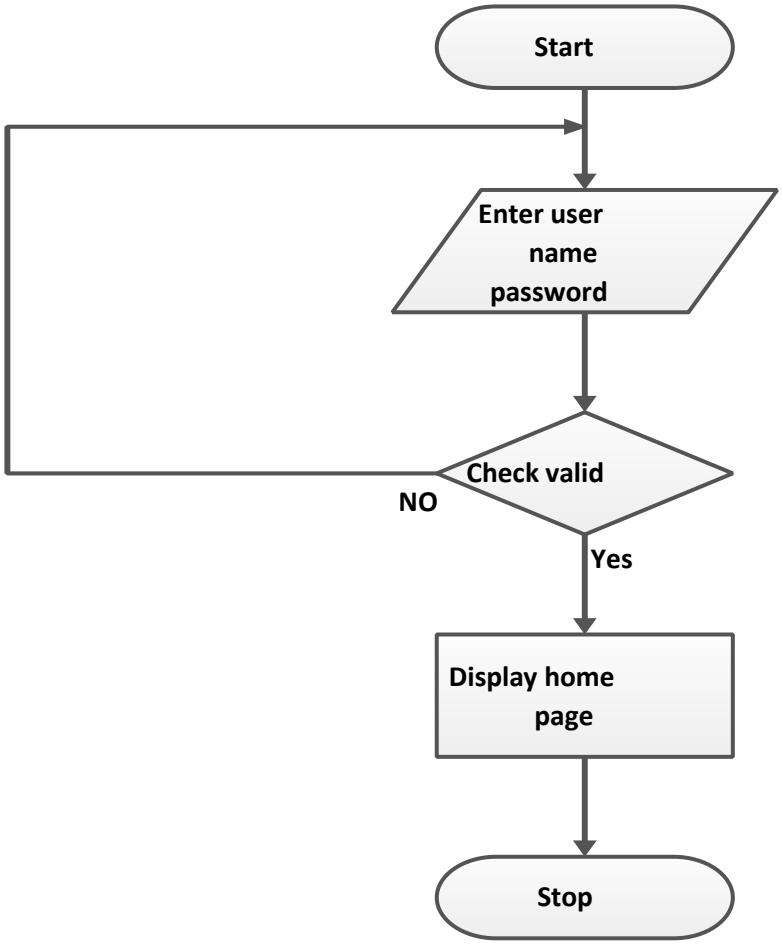


5.4.11. Structured chart for Account information

5.5 DETAILED DESIGN:

5.5.1 Login:

- **Input:** Email and password.
- **Procedural Details (Flowchart)**



- **File I/O Interface**

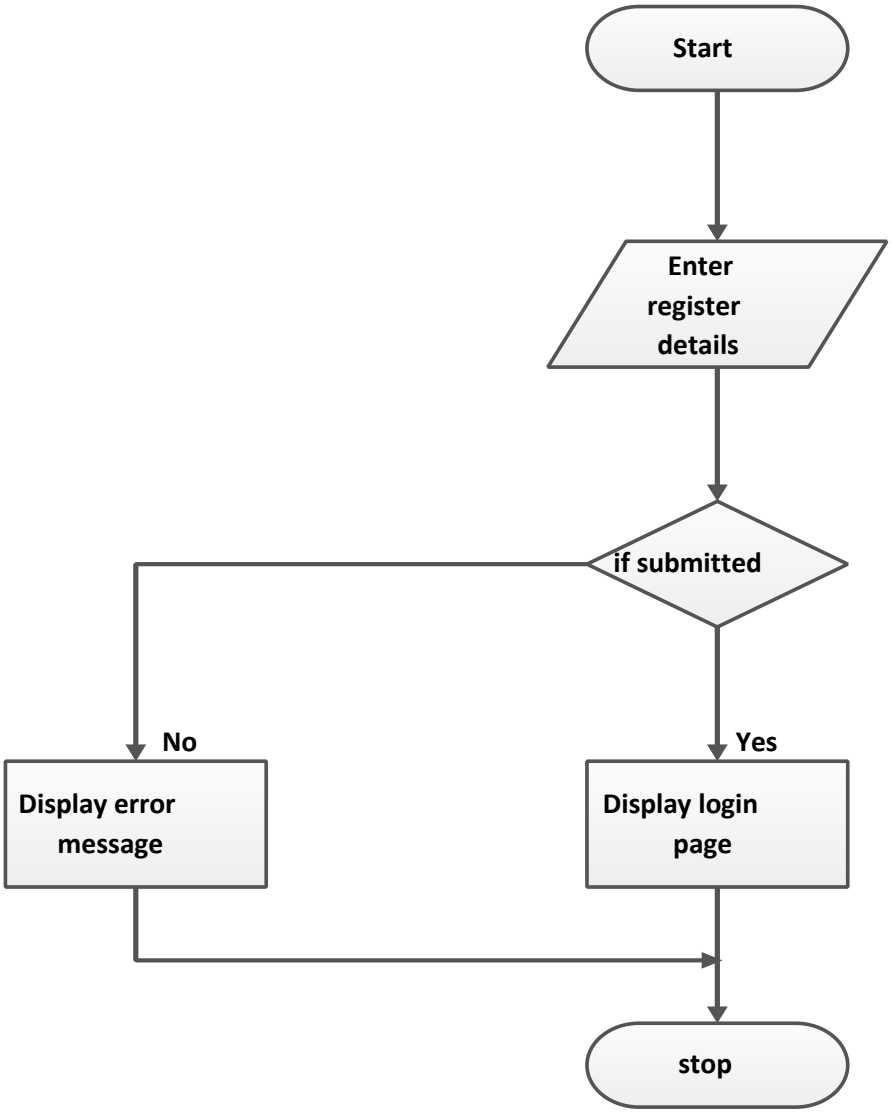
  Graphical User Interface

- **Output**

  If all the entered data will be matched with respect to the database, then user will be redirected to homepage.

5.5.2. Registration:

- **Input:** First name, Last name, contact number, image, email id, password, gender, date of birth.
- **Procedural Details (Flowchart)**



- **File I/O Interface**

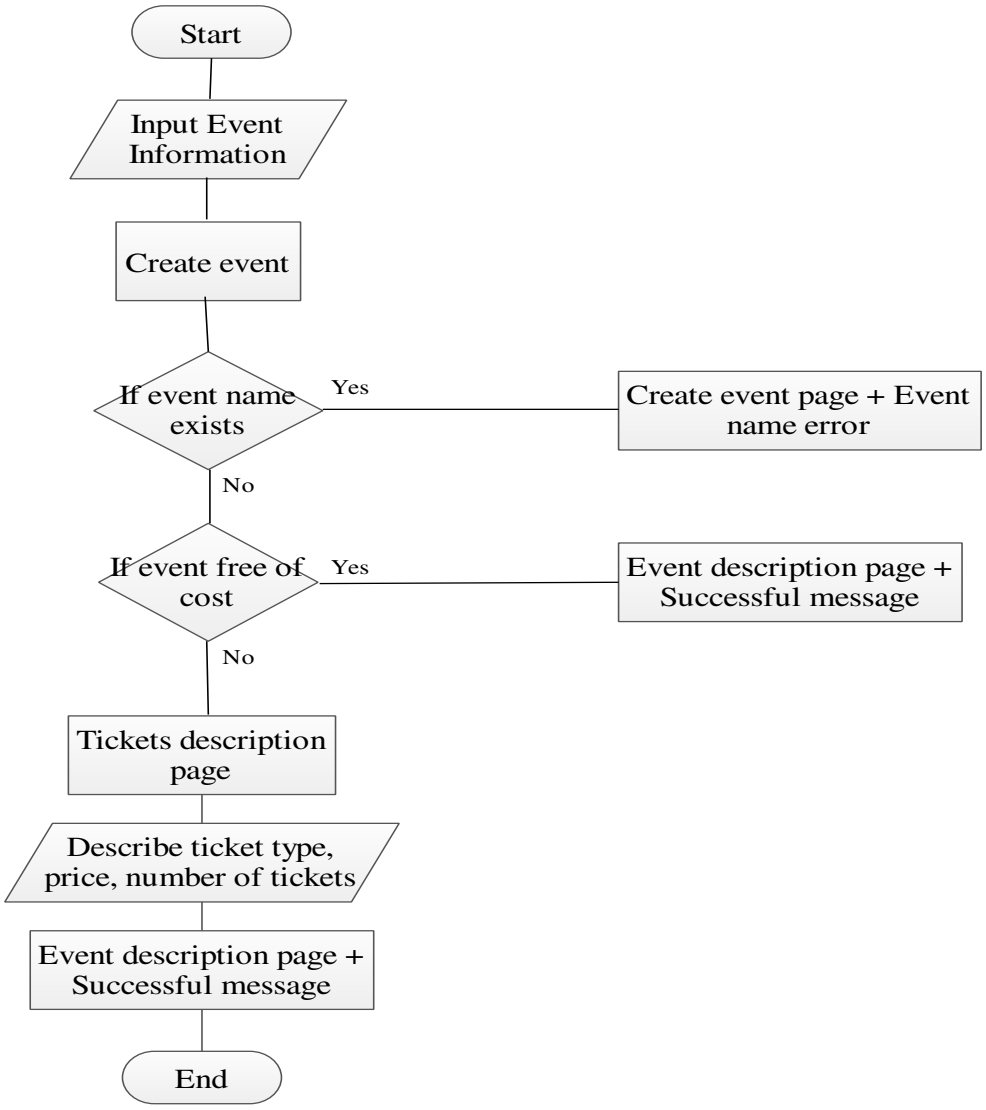
  Graphical User Interface

- **Output**

  All the entered data will be stored in database and could be viewed by the user in account. On successful login user will be taken to login page.

5.5.3. Creation of the Event:

- **Input:** type, event name, basic description, location, date, time, video link, image, description.
- **Procedural Details (Flowchart)**



- **File I/O Interface**

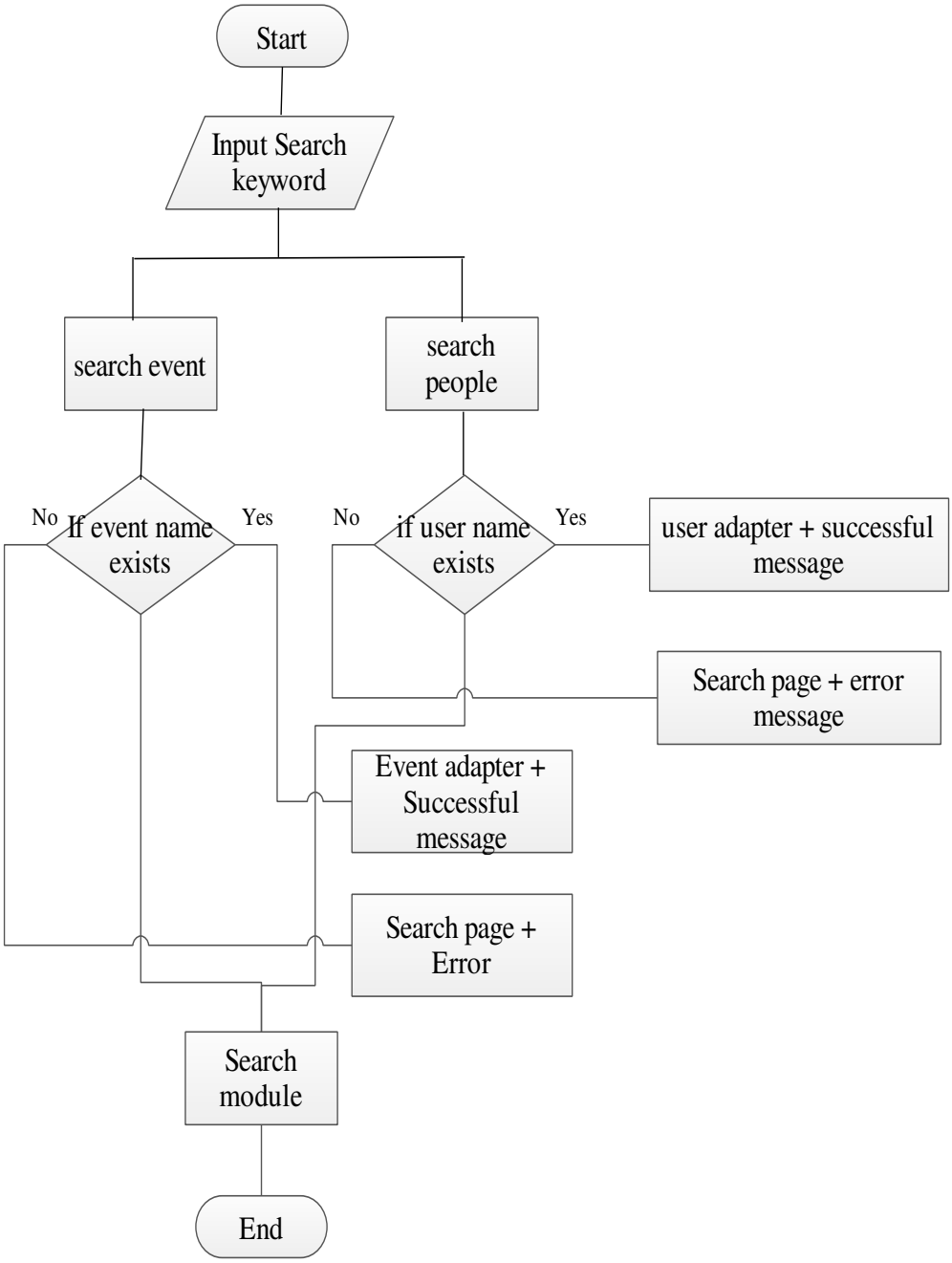
  Graphical User Interface

- **Output**

  All the entered data will be stored in database and could be viewed by the user in account. And it also can be viewed by the other user.

5.5.4. Search:

- **Input:** keyword about an event or a particular user.
- **Procedural Details (Flowchart)**



- **File I/O Interface**

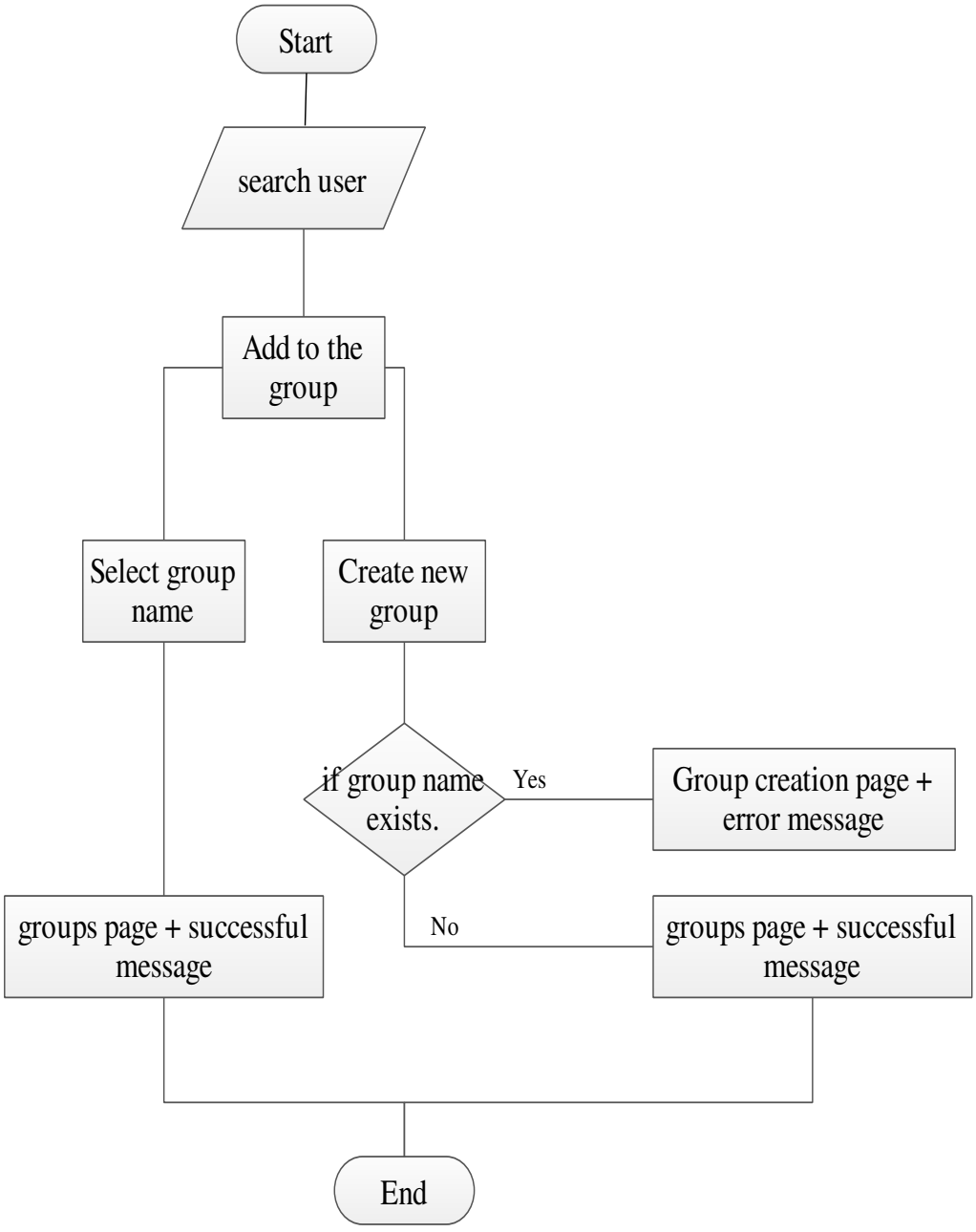
  Graphical User Interface

- **Output**

  The data information corresponding to the given keyword is accessed from the database.

5.5.5. Group creation:

- **Input:** user id, group name.
- **Procedural Details (Flowchart)**



- **File I/O Interface**

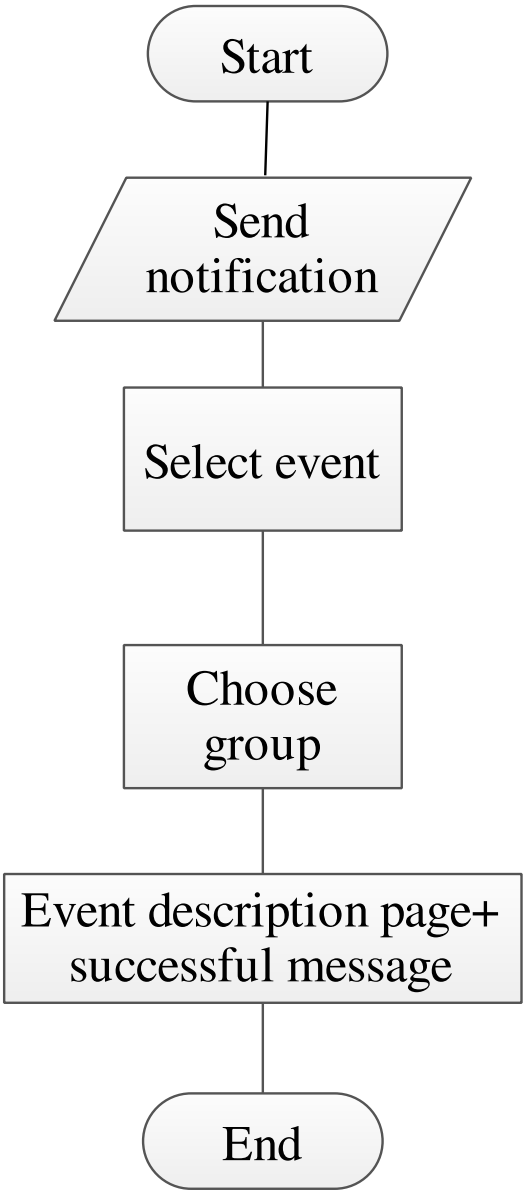
  Graphical User Interface

- **Output**

  The given user id will be stored with the given group name in the database, and the particular user can be accessed as member of that particular group.

5.5.6. Send notification:

- **Input:** user id, group name, event id.
- **Procedural Details (Flowchart)**



- **File I/O Interface**

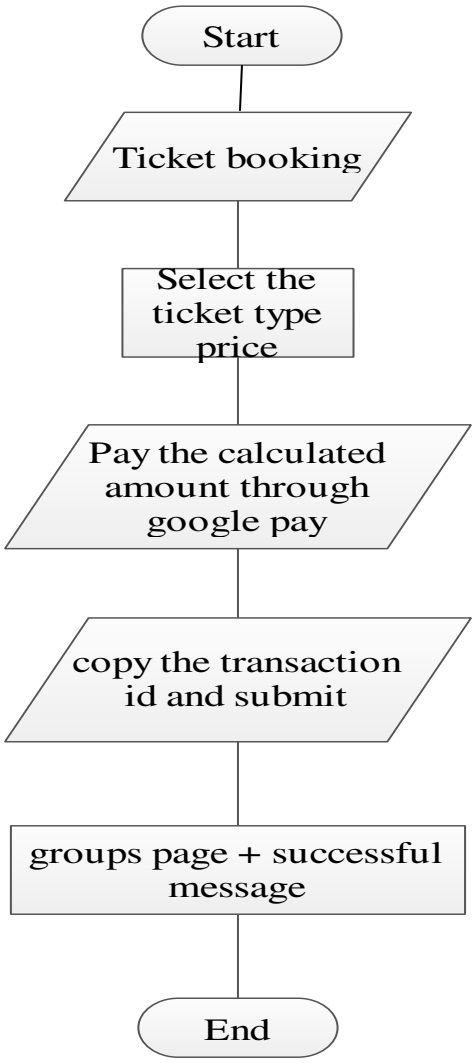
  Graphical User Interface

- **Output**

  Here notification information will be stored in the database and can be accessed by the connected members of the group

5.5.7. Ticket booking:

- **Input:** user id, ticket details, event id, transaction id.
- **Procedural Details (Flowchart)**



- **File I/O Interface**

  Graphical User Interface

- **Output**

  Here the tickets data will be inserted into the database and can be further used by the user for the ticket verification.

CHAPTER-6
CODING

CODING

6.1. INTRODUCTION:

The goal of coding or programming phase is to translate the system design, produced during the designing phase, into code in a given programming language which can be executed by a computer and that performs the computation specified by the design. During the implementation, it should be kept in mind that the programs should not be constructed so that they are easy to write, but that they are easy to read and understand.

6.2. PROGRAMMING PRACTICES:

6.2.1. Top-down & Bottom-up approaches:

In a top-down implementation, the implementation starts from the top of the hierarchy and proceeds to the lower levels. First the main module is implemented, and their subordinates, and so on. In a bottom-up implementation, the process is the reverse. The development starts with implementing the modules at the bottom of the hierarchy and proceeds through the higher levels until it reaches the top.

Top-down and bottom-up implementation should not be confused with top-down and bottom up design, here the design is being implemented, and if the design is fair detailed and complete, its implementation can proceed in either the top-down manner.

6.2.2. Structured Programming:

The basic objective of the coding activity is to produce programs are easy to understand. A program has a static structure as well as a dynamic structure. The static structure is the structure of the text of the program, which is usually just a line organization of statements of the program. The dynamic structure of the program is the sequences of statements executed during the execution of the program. In other words, both the static structure and the dynamic behaviour are sequences of statements; when the sequence representing the static structure is fixed, the sequences of statements it executable can change from execution to execution.

6.2.3. Information Hiding:

A software solution to a problem always contains data structures that are meaningfully represent information in the problem domain. That is, when software is developed results a problem, the software uses some data structures to capture the information in the problem domain. Any software solution to a problem contains data structures that represent information in the problem domain. In the problem domain, in general, only certain operations are performed on some information. That is, a piece of information in the problem domain is used only in a limited number of ways in the problem domain.

6.2.4. Programming style:

Here we will list some general rules that can be applied for writing good code.

I. **Names:** Selecting module and variable names is often not considered important novice programmers. Most variable sin a program reflect some entity in the problem domain, and the modules reflect some process. Variable names should be closely related to the entity they represent, and module names should reflect their activity.

II. **Control constraints:** As discussed earlier, it is desirable that as much as possible single entry, single exit constructs be used. It is also desirable to use a few standard control constructs rather than using a wide variety of constructs, just because they are available in the language.

III. **Information hiding:** As discussed earlier, information hiding should be supported where possible. Only the access functions for the data structures should be made visible while hiding the data structures behind these functions.

IV. **User-defined types:** Modern languages allow users to define data types when such facilities are available, they should be exploited where applicable.

V. **Nesting:** The different control constructs, particularly the if-then-else, can be nested. If the nesting becomes too deep, the programs become harder to understand. In case of deeply nested if-then-else, it is often difficult to determine if statement to which a particular else clause is associated. If possible, deep nesting should be avoided.

6.2.5 Verification:

Verification of the output of the coding phase is primarily interested for detecting errors introduced during this phase. That is, the goal of verification of the code produced is to show that the code is consistent with the design is supposed to implement. It should be avoided out that by verification we mean providing correctness of programs. Program verification method fall into two categories- static and dynamic methods.

In dynamic methods, the program executes some test data and the outputs of the program are examined to determine if there are any errors present.

Static techniques, on the other hand, do not involve actual program execution on actual numeric data, though it may involve some form of conceptual execution

6.2.5.1. Code Reading:

Code reading involves careful reading of the code by the programmer to detect any discrepancies between the design specifications and the actual implementation. It involves determining the abstraction of a module and then comparing it with its specifications. The process of code reading is best done by reading the code inside-out, starting with the innermost structure of the module. First determine its abstract behaviour and specify the abstraction. Then the higher-level structure is considered, with the inner structure replaced by its abstraction.

This process is continued until we reach the module or program being read.

6.2.5.2. Static analysis:

Analysis of programs by methodically analysing the program text is called static analysis is usually performed mechanically by the aid of software tools. During static analysis the program itself is not executed, but the program text is the input to the tools. The aim of the static analysis tools is to detect errors or potential errors or to generate information about the structure of the program that can be useful for documentation or understanding of the programs. An advantage is that static analysis sometimes detects the errors themselves, not just the presence of errors as in testing. This saves the effort of tracing the error from the data that

reveals the presence of errors. Static analysis can provide insight into the structure of the program.

6.2.5.3. Symbolic execution:

Inputs to the program are not numbers but symbols representing the input data, which can take different values. The execution of the program proceeds like normal execution, except that it deals with values that are not numbers but formulates consisting of the symbolic input values. The outputs are symbolic formulae of input values. These formulae can be checked to see if the program will behave as expected. This approach is called as symbolic execution.

Authentication code:

```
$result=mysql_query("SELECT `email`, `password` FROM `user` WHERE email='$email' and password='$pwd'") or die(mysql_error());
```

Validation code:

```
<EditText
   android:hint="E-mail"
   android:inputType="textWebEmailAddress/>

<EditText
   android:inputType="textPassword"
   android:hint="password"/>
```

Insert code:

```
   $q=mysqli_query($conn,"INSERT INTO `user`(`fname`, `lname`, `email`, `gender`, `phone`, `dob`, `password`, `user_image`,`status`) VALUES ('$fname','$lname','$email','$selectedgender','$mobileno','$dateofbirth','$password','$name','1')")or die(mysqli_error($conn));
```

Delete code:

```
$q=mysqli_query($conn,"DELETE FROM `user` WHERE uid='$uid'");
```

Update code:

```
$q=mysqli_query($conn,"UPDATE user SET fname='$fname' & lname='$lname' & email='$mailid' & gender='$selectedgender' & phone='$mobileno' & dob='$dateofbirth' & password='$password' & user_image='$userimage' & moreinfo='$moreinfo' WHERE email='$mailid'");
```

Connection code:

```
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname="eventzo";

// Create connection
$conn = mysqli_connect($servername, $username, $password,$dbname);

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
// echo "Connected successfully";
?>
```

CHAPTER-7
TESTING

TESTING

7.1. INTRODUCTION:

Testing is the major quality control measure used during software development. It is a basic function to detect errors in the software. During the requirement analysis and design the output of the document that is usually textual and non-executable after the coding phase the computer programs are available that can be executed for testing purpose. This implies that testing not only has to uncover errors introduce during the previous phase. The goal of testing is to uncover requirement, design, and coding errors in the program.

Testing determines whether the system appears to be working according to the specifications. It is the phase where we try to break the system and we test the system with real case scenarios at a point.

The implementation is the final and important phase. It involves user-training, system testing in order to ensure successful running of the proposed system. The user tests the system and changes are made according to their needs. The testing involves the testing of the developed system using various kinds of data. While testing, errors are noted and correctness is the mode.

7.2. OBJECTIVES OF TESTING:

The objectives of testing are:

- Testing is a process of executing a program with the intent of finding errors.
- A Successful test case is one that uncovers an as- yet-undiscovered error.

System testing is a stage of implementation, which is aimed at ensuring that the system works accurately and efficiently as per the user need, before the live operation commences. As stated before, testing is vital to the success of a system. System testing makes a logical assumption that if all parts of the as system are correct, the goal will be successfully achieved. A series of tests are performed before the system is ready for the user acceptance test.

7.3 TEST REPORTS:

7.3.1. Unit Testing:

Unit testing focuses efforts on the smallest unit of software design. This is known as module testing. The modules are tested separately. The test is carried out during programming stage itself. In this step, each module is found to be working satisfactory as regards to the expected output from the module.

7.3.2. Integrate Testing:

Data can be lost across an interface. One module can have an adverse effect on another, sub functions, when combined, may not be linked in desired manner in major functions. Integration testing is a systematic approach for constructing the program structure, while at the same time conducting test to uncover errors associated within the interface. The objective is to take unit tested modules and builds program structure. All the modules are combined and tested as a whole.

7.4 TEST CASES

7.4.1 Login form:

Sl No	Test Condition	Expected Result	Result
1	If admin clicks on login button without entering username and password.	Invalid username	Successful
2	If username is blank but password is entered.	Invalid username	Successful
3	If password is blank but username is entered.	Invalid password.	Successful
4	If the username or password is incorrect.	Invalid username and password.	Successful
5	If the valid username and valid password is entered	System displays main page.	Successful

7.4.2. User Registration Form:

Sl No	Test Condition	Expected Result	Result
1	When the user clicks “Sign Up”	It adds new record to database.	Successful
2	If the first name field is blank and clicks “signup”button	Please enter the first name.	Successful
3	If you enter an invalid phone number with characters	Please match the format requested.(10 numeric character only)	Successful
4	If the phone number is empty	Please enter the phone number.	Successful
5	If the Email is not in proper manner	Please enter the email in a proper format	Successful
6	If the e-mail field is empty	Please enter the e-mail.	Successful
7	If the date of birth is empty	Please select the date of birth	Successful
8	If the image is empty	Please select the proper image	Successful
9	If the terms and conditions is not checked	“Sign up” will be invisible	Successful

7.4.3. Home page:

Sl No	Test Condition	Expected Result	Result
1	When the user clicks the “Home” in the menu item.	It displays the home page.	Successful
2	When the user clicks the “Hosted” in the menu item.	It displays the details about hosted event.	Successful
3	When the user clicks the	It displays the	Successful

	"Inbox" in the menu item.	notifications page	
4	When the user clicks the "Account" in the menu item.	It displays user's personal information	Successful
5	When the user clicks the "Create" in the menu item.	It displays the event creation page	Successful
6	When the user clicks the "Groups" in the menu item.	It displays the groups information created by the user	Successful
7	When the user clicks the "Search" in the menu item.	It displays the searching page	Successful

7.4.4. Create page:

Sl No	Test Condition	Expected Result	Result
1	If user clicks on "NEXT"	It adds new record to the database	Successful
2	If the hosting type is not checked	After clicking "NEXT" it displays tickets information page	Successful
3	If the Event type is not selected	Please select the proper event type	Successful
4	If the Event name is empty.	Please enter the event name	Successful
5	If the basic description is empty	Please enter the basic description	Successful
6	If the location	Please enter the location.	Successful
7	If date is empty	Please enter the date	Successful
8	If the time is empty	Please enter the time	Successful
9	If the description is empty	Please enter the description	Successful

7.4.5. Ticket information page

Sl No	Test Condition	Expected Result	Result
1	If the ticket type is empty	Please enter the ticket type	Successful
2	If the ticket price is empty	Please enter the price of the ticket	Successful
3	If the number of tickets is empty	Please enter the number of tickets to sell.	Successful
4	When the user clicks “add one more ticket type”	Ticket type will be added and the same page will be displayed for further ticket adding.	Successful
5	When the user clicks “Submit”	Ticket type will be added and the home page will be displayed.	Successful

7.4.6. Search page:

Sl No	Test condition	Expected Result	Result
1	If a user clicks on “people”	User data will be accessed from the user database	Successful
2	If the user clicks on ”Events”	Data will be accessed from the event database	Successful
3	If the search item is blank and clicks the search button	Please enter the search item	Successful
4	If the search button clicked	Information regarding the particular search item will be displayed	Successful

		in the recycler	

7.4.7. Group creation:

Sl No	Test Condition	Expected Result	Result
1	If a user clicks on “add to the group” which is below the user details	Group creation page will be displayed	Successful
2	If the user clicks on “create” when the group name is empty	Please enter the group name	Successful
3	If the user clicks on “create”	User information will be inserted into the group	Successful
4	If the user clicks on any one of the group name in the recycler.	User information will be inserted into the group	Successful

7.4.8. Acount page

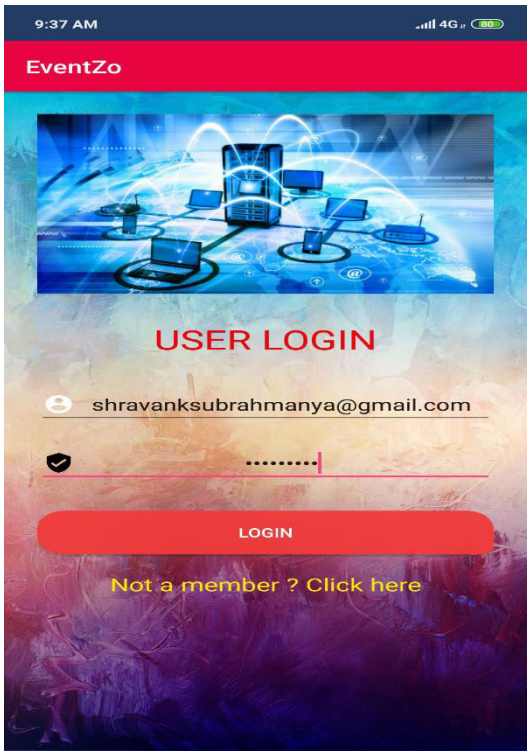
Sl No	Test Condition	Expected Result	Result
1	If a user clicks on “Edit account”	Edit account page will be displayed	Successful
2	If the user clicks on “Live events”	Live Events page will be displayed	Successful
3	If the user clicks on “Hosted events”	Hosted events page will be displayed	Successful
4	If the user Clicks on “ Purchases”	Purchases page will be displayed	Successful
5	If the user clicks on “Help & Feedback”	Feedback page will be displayed	Successful
6	If the user clicks on “delete account”	Users account will be deleted and login page	Successful

		will be displayed	
7	If the user clicks on “Logout”	Login page will be displayed	Successful

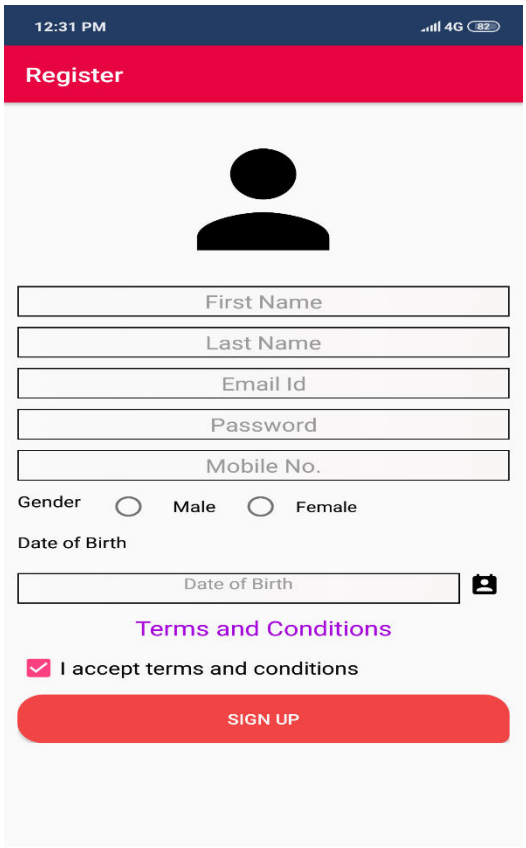
CHAPTER-8
SCREENSHOTS

SCREENSHOTS

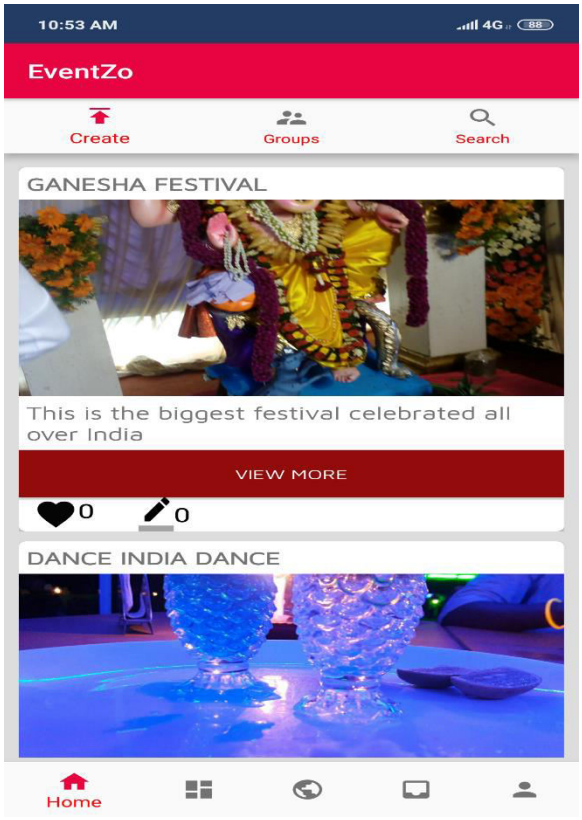
Login page:



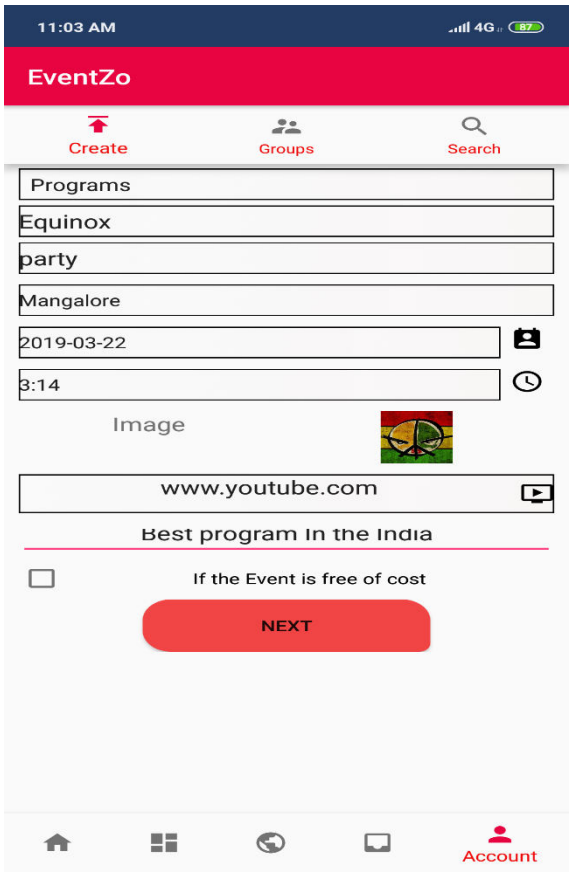
Registration page:



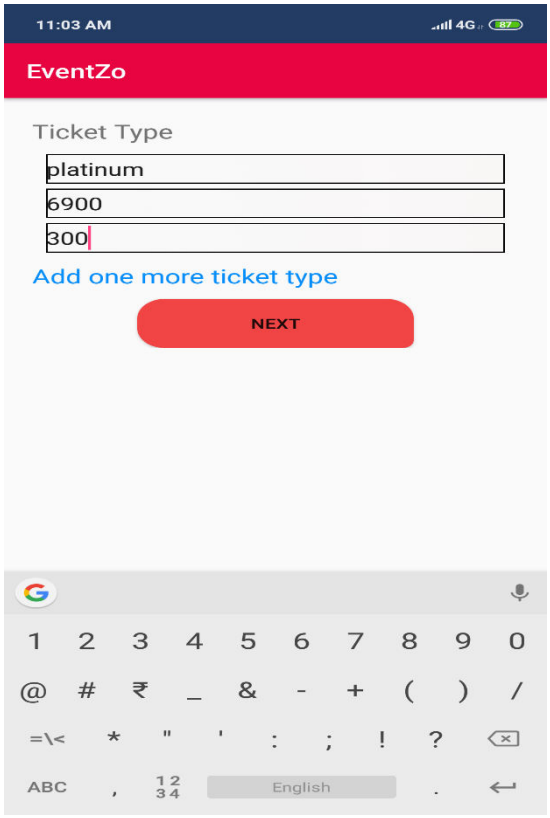
Home page:



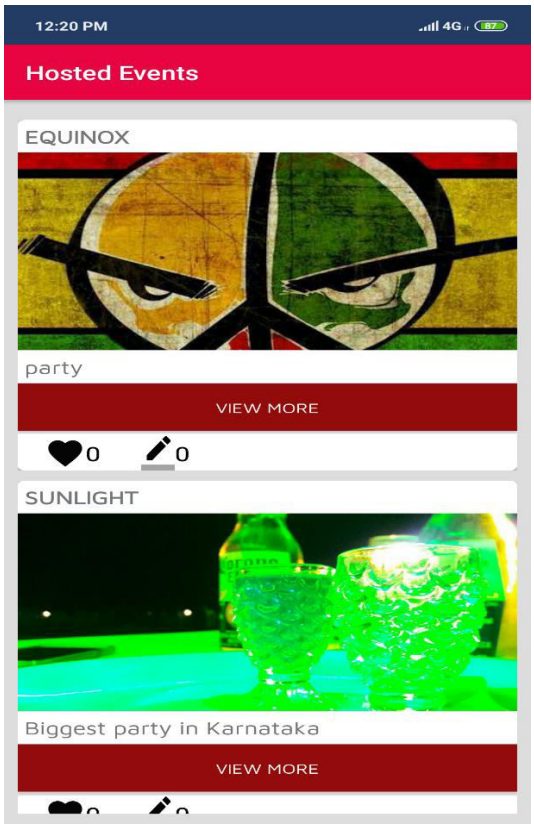
Event creation page:



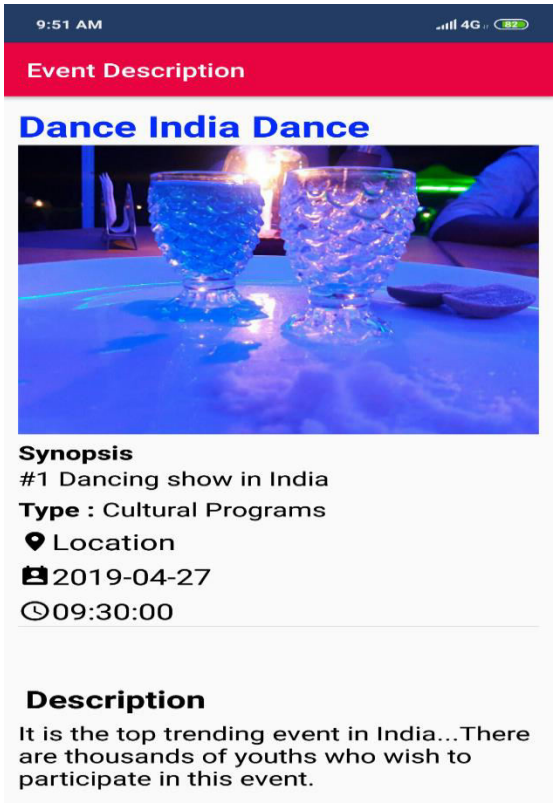
Tickets Information page:



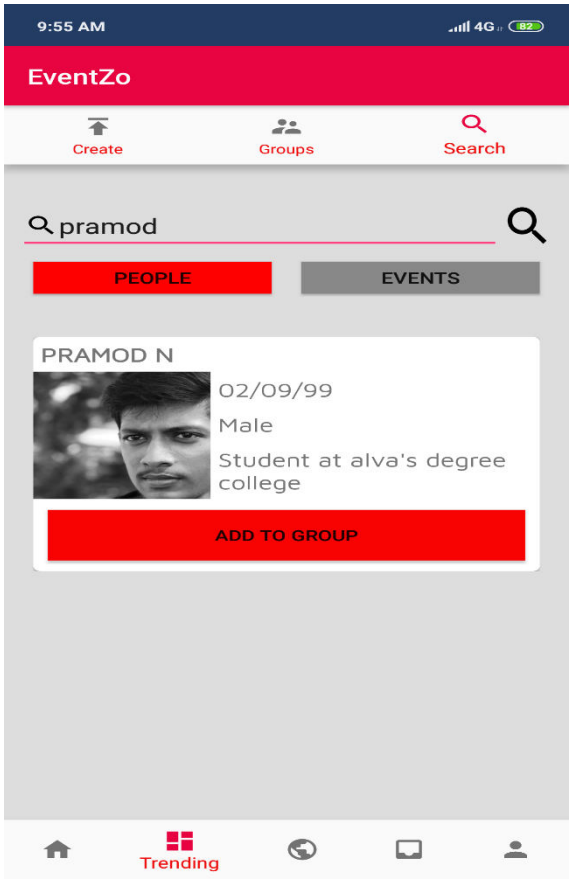
Hosted Events page:



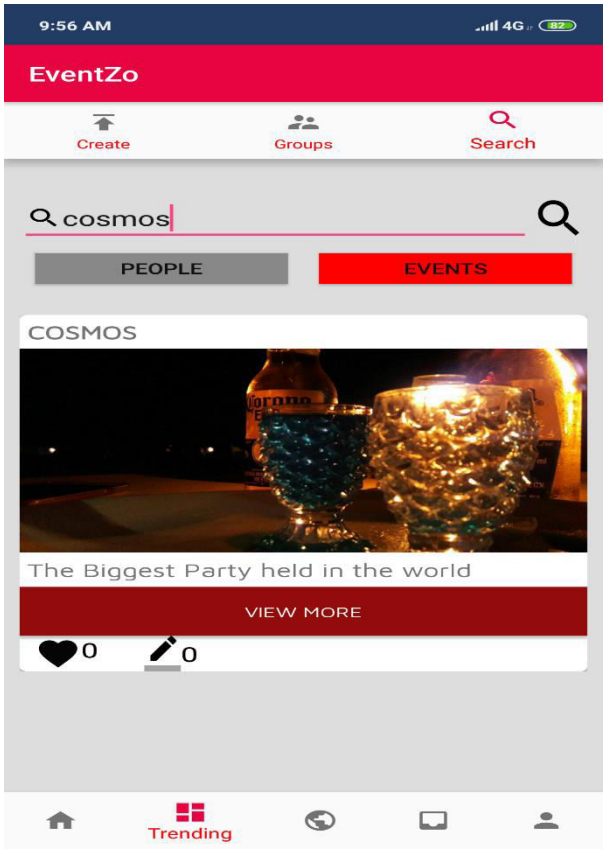
Events Description page:



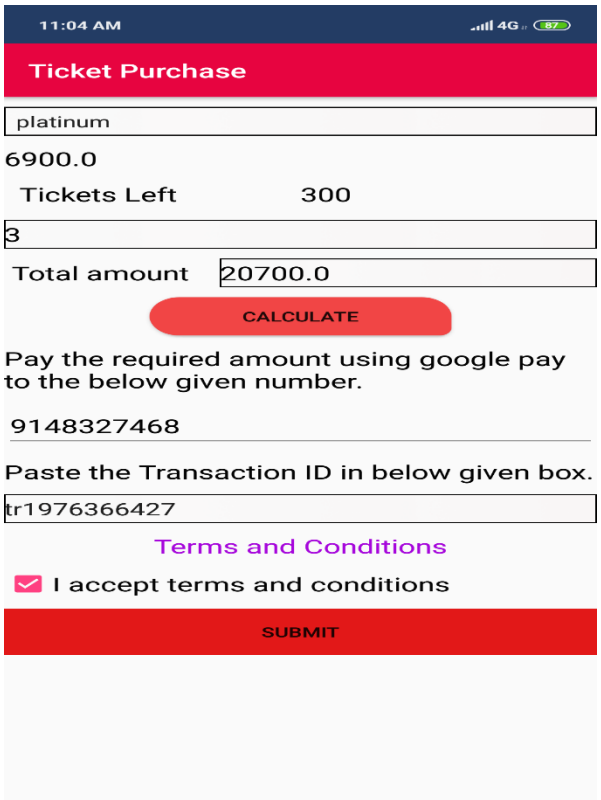
Search page: search people



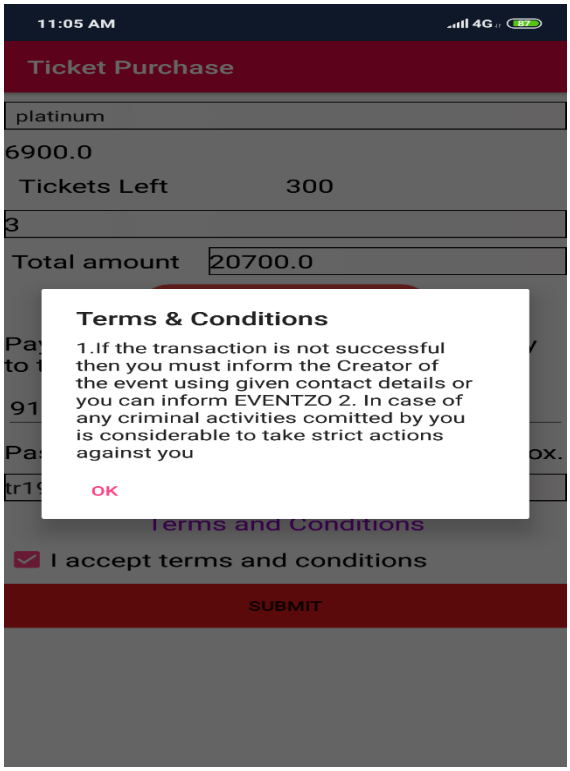
Search page: search events



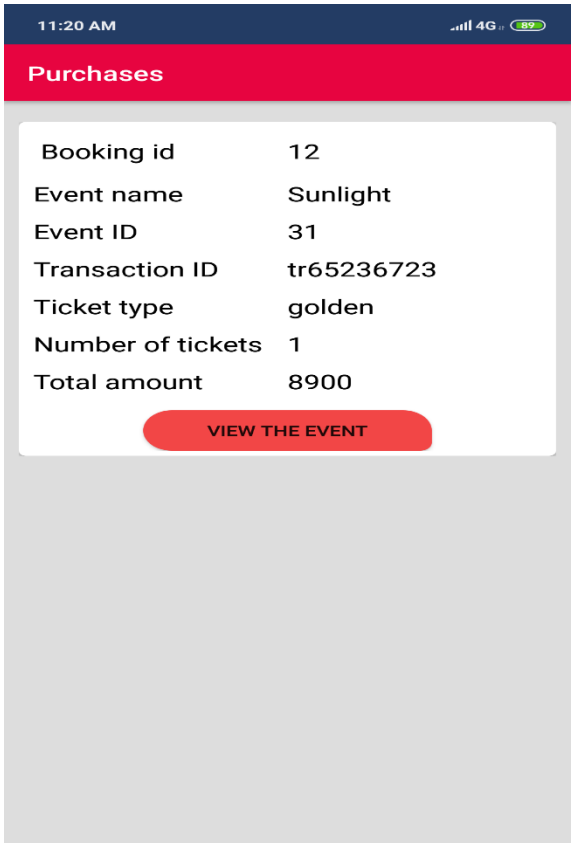
Ticket Purchase page:



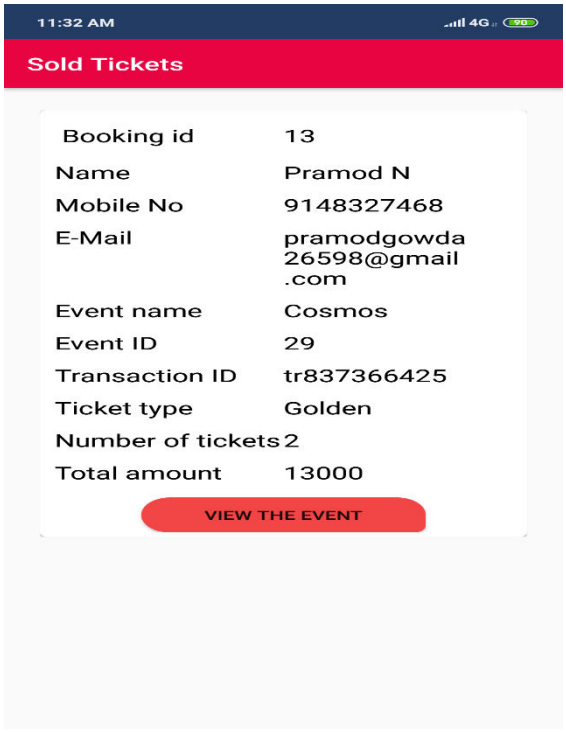
Ticket Purchase Terms and Conditions:



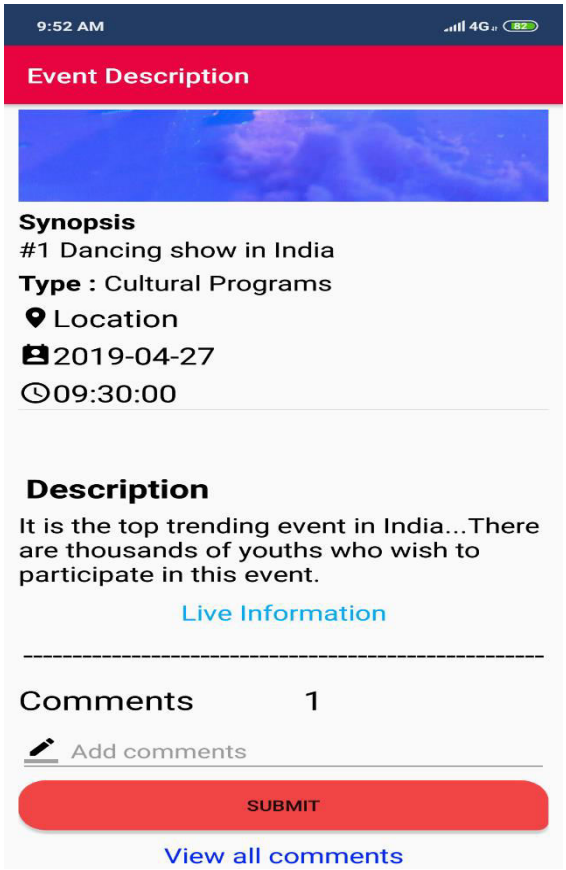
Purchases page:



Sold Tickets page:



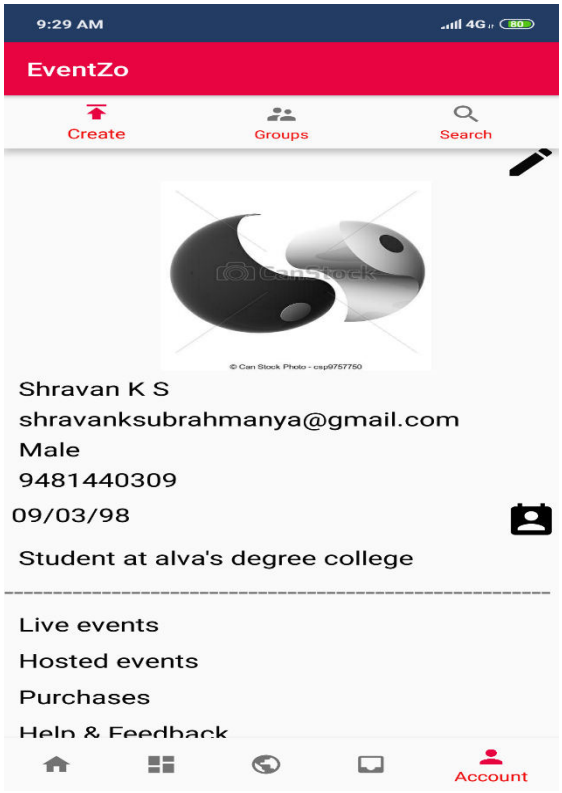
Description page Comment part:



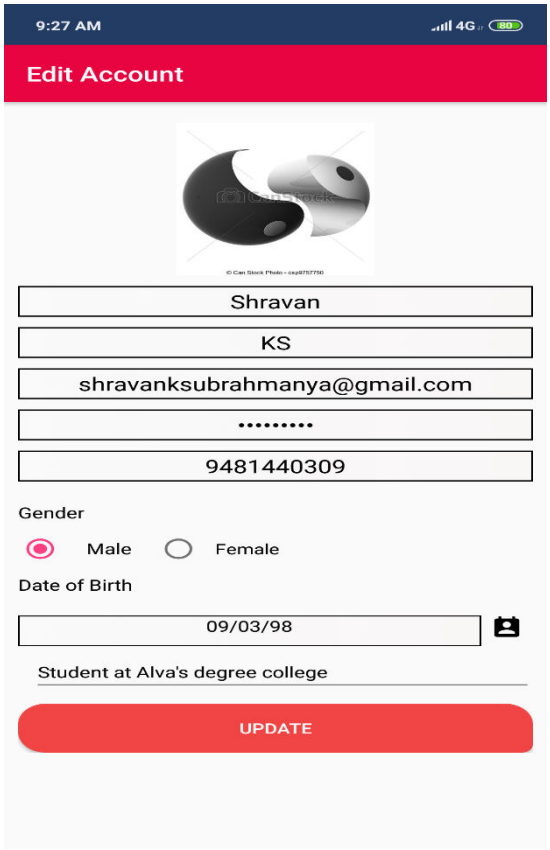
View all comments page:

Inbox page:

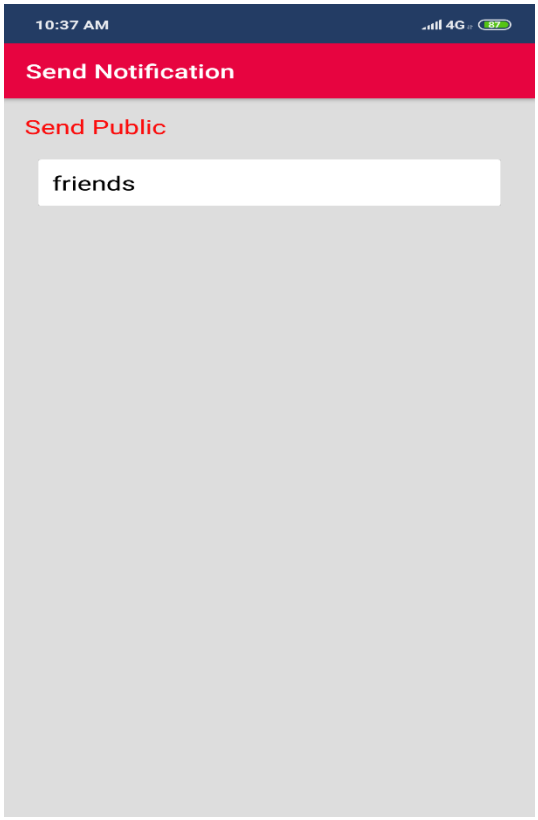
Account page:



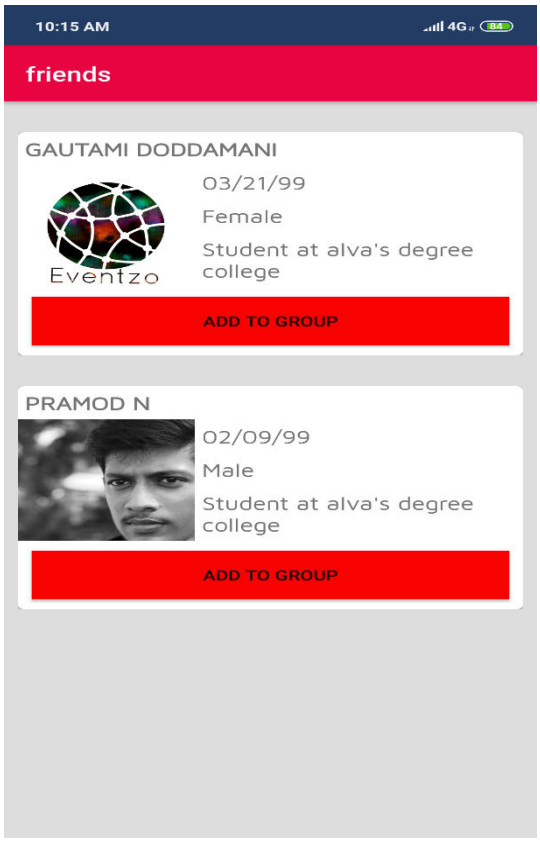
Edit Account page:



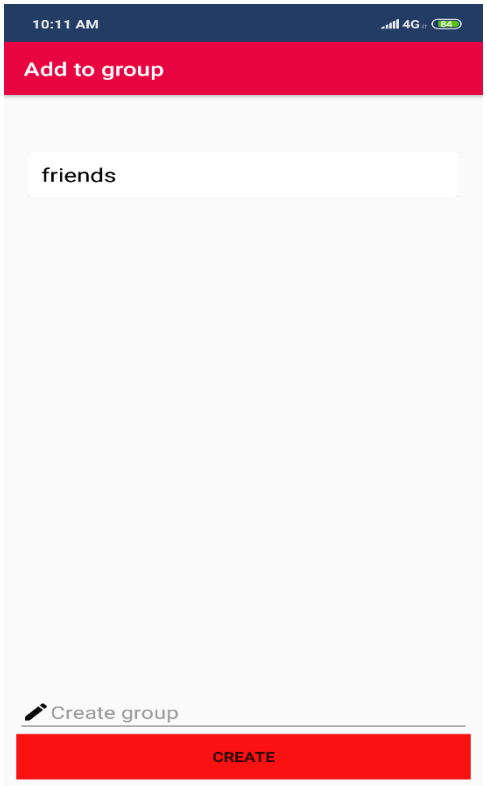
Groups page:



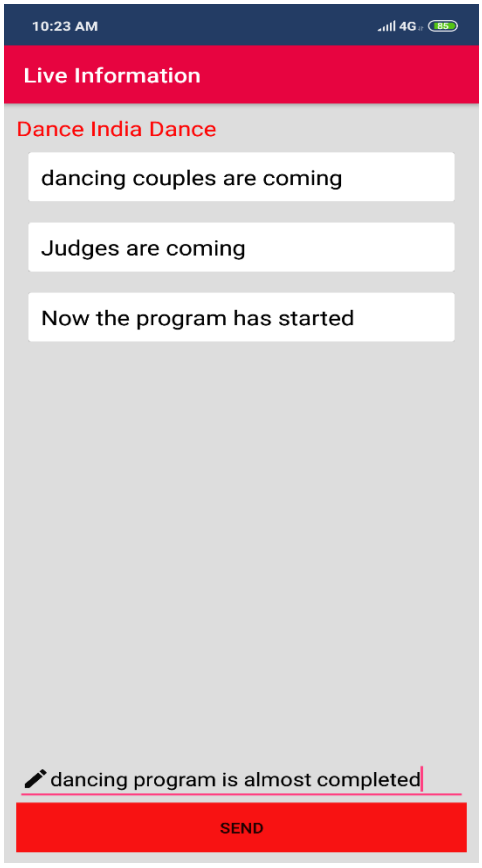
Groups description page:



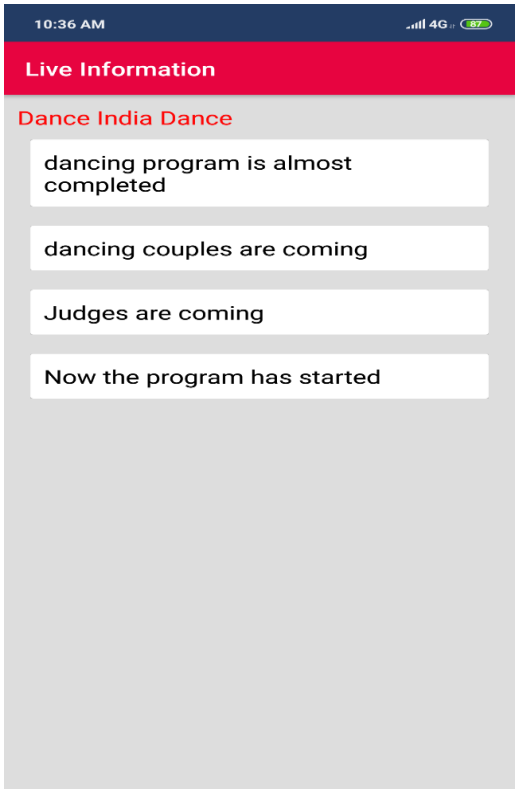
Groups Creation page:



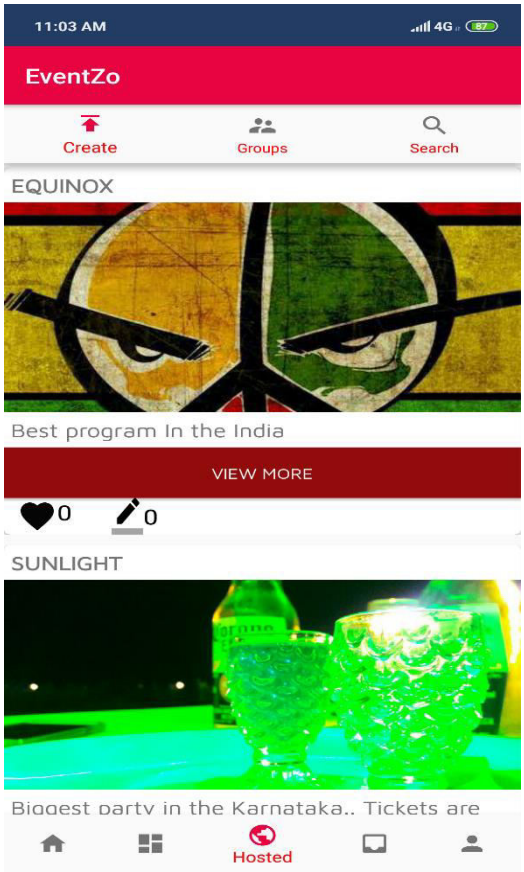
Live information page: for the event creator



Live information page: for the event viewer



Hosted page:



CHAPTER-9
USER MANUAL

USER MANUAL

9.1. ANDROID STUDIO 3.2

Android Studio is the official IDE from **Google** for developing android applications for android devices. It consists of all the API required to create an app. It has emulators on which you can test the developed app. Before the introduction of Android studio, android applications were mostly developed using Eclipse IDE.

Android studio gives us a proper environment for developing and for coding of android apps. Most of the apps available in play store is developed in java only, because kotlin is a new language comes in android development.

It has a strong editor tool for developing creative UI and emulators for different versions to test and simulate sensors without having actual Android devices. It also has a very useful Gradle plugin using which you can create application files (apks) with different configurations. Moreover it makes exporting and uploading apk on play store easy with a single click.

In **Android Studio**, **Gradle** is a custom build tool used to build android packages (apk **files**) by managing dependencies and providing custom build logic. APK **file** (**Android** Application package) is a specially formatted zip **file** which contains. Byte code. Here java and XML are the main programming languages for developing the frontend application.

Some important features of android studio is as follows.

- Kotlin programming language support.
- Android profiler: memory, CPU, network.
- Java 8 language features built-in.
- Faster build times.
- Device file explorer.
- Android Instant Apps support.
- Adaptive icon wizard.

9.2. SUBLIME TEXT

Sublime Text is a proprietary cross-platform source code editor with a Python application programming interface (API). It natively supports many programming languages and mark up languages, and functions can be added by users with plugins, typically community-built and maintained under free-software licenses.

This is the main platform used for backend coding that is PHP (Hypertext Pre-processor) and MySQL queries.

The following is a list of features of Sublime Text.

- "Go to Anything," quick navigation to files, symbols, or lines
- "Command palette" uses adaptive matching for quick keyboard invocation of arbitrary commands
- Simultaneous editing: simultaneously make the same interactive changes to multiple selected areas
- Python-based plugin API
- Project-specific preferences
- Extensive customizability via JSON settings files, including project-specific and platform-specific settings
- Cross-platform (Windows, macOS, and Linux) and Supportive Plugins for cross-platform
- Compatible with many language grammars from TextMate

9.3. WAMPSERVER

WAMP. Stands for "Windows, Apache, MySQL, and PHP." **WAMP** is a variation of LAMP for Windows systems and is often installed as a software bundle (Apache, MySQL, and PHP). It is often **used** for web development and internal testing, but may also be **used** to serve live websites.

WampServer is a Web development platform on Windows that allows you to create dynamic Web applications with Apache2, PHP, MySQL and MariaDB. Best of all, **WampServer** is available for **free** (under GPML license) in both 32 and 64 bit versions.

Some of the features WAMPSERVER are as follows.

- preconfigured WAMP-System with Apache, PHP, MySQL, phpMyAdmin, Mercury/32 Mail-Server and SQLite
- Equally useful for developers as for productive usagee thanks to switchable settings between security- and developer-functionality
- Centralized GUI for all important features of the above applications
- Assistant to operate the applications as a restricted user
- Add-on system for reloading and integration of other applications (such as Tomcat, Python, SVN, etc.) from the network
- 1-Click Installer for popular web applications such as phpBB, WordPress u.v.a.
- Assistant for performance tuning with automatic Evaluation of the System configuration (number of CPUs, RAM, etc.)
- Editor for the Windows HOSTS file to create "virtual domains"
- dynamic and automatic updates of individual components (Apache, MySQL, etc.) directly from the software
- Focus on a lightweight and fast system - the complete setup consists of only a 33MB file

9.4 EXECUTION METHODS:

User: Each user will have an individual account. They can perform the following activities.

Creation: This module allows the user to create an event by giving complete description about the event and the tickets if he/she is going to sell online.

Events View: This module allows the user to get the information about the events which has been created by the other members.

Search: This module allows the user to search the events or the other members of EVENTZO and get the information about them for the next step.

Grouping: By clicking the user's details the particular user will be redirected to a group creation activity where he will add another user to the group or he can create another group by adding the particular member.

Send Notification: After clicking the send notification text in the event description page the user will be redirected to send notifications page where he will select the group name to which the notifications should be sent.

Ticket Booking: This module allows the particular user to purchase the tickets for the events which have been hosted by other members of the EVENTZO or the EVENTZO itself. Here user will be redirected to the ticket purchase page after clicking the book tickets button in the event description page.

Account: This module allows the particular user to get his user information and edit his/her details when he clicks on the Account icon in the user's home page.

Sold Tickets: This feature allows the user to get the information about the sold tickets about the particular event created by him. When the particular user clicks the sold tickets in the event description page the tickets information will be available to him.

Live Information Module: This module allows the particular user to send information about the event in a message format. User can send the messages live during ongoing of the event. Other members of the EVENTZO can view these messages

CHAPTER-10
FUTURE ENHANCEMENT
&
CONCLUSION

FUTURE ENHANCEMENT

The project was developed to help end users to interact with the client easily.

The future enhancements that are possible in the project are as follows.

- In future, we provide scoreboards to upload the information about the particular sport.
- Bank transaction will be implemented for the transaction purpose.
- In future, we provide multiple images uploading, video uploading and live video uploading.
- Location sharing and tracing will be added in the future.
- We provide multiple themes to upload the information about an event.
- We provide themes for the scoreboards to upload the information about the sports.
- In future, we tie up with the state, national and multinational, Sports and Event hosting companies to utilize our applications for hosting.
- We can communicate with the public with the help of SMS service.
- We can implement finger print technology for authenticity.
- We can implement speech recognition for further support.

CONCLUSION

The application is built with Android (in which Java, XML as frontend), MySQL and PHP as backend. The scope of EVENTZO system is to provide all the event related services online. The services provided in which first one is information uploading (live messages, complete description, video link and articles etc.), second one is accessing events (about all type of events like Sports, Competitions, Social Events and Awarding functions etc.), third one is allowing the each members for selling tickets for the hosted events online.

At the heart of the "EVENTZO System" Includes the following features.

- Event Viewing: - The application is very simple to use with efficient user friendly interface which allow the end users to View the event information which has been created by the other users.
- Event Creation: -This application allows each end users to create an event or share the information about an event in which he or she is participating. This application guarantees that data is available 24x7
- Ticket Selling and purchasing: - This application allows each users to sell the tickets for the particular event which has been created by him. And it also allows a particular user to purchase a ticket for the events created by another user.

CHAPTER-11
BIBLIOGRAPHY

BIBLIOGRAPHY

 Books

1] Donn Felker, **Android Application Development for Dummies**, Wiley Publishing Inc., 2011.

2] **Pankaj Jalote,** an Integrated Approach to Software Engineering**, 2nd** Edition, Narosa Publishing House, 2004

3] Luke Welling and Laura Thomson, **PHP and MySQL Web Development**, Pearson, 2010

 Websites

1] http://www.youtube.com/Jerry-Banfield

2] http://www.youtube.com

3] http://www.google.com

4] https://www.linkedin.com/Online-Training/Android-App-Development