Department of Information Technology

University of Gujrat

Event Scheduling System

Submitted By:

M. Talha Tarar	**16491556-013**
M. Hamza Javed	**16491556-023**
Ahmed Jamal	**16491556-040**

Supervised By:

Mam Sabiqa Ashraf

STATEMENT OF SUBMISSION

This is certified that **M. Talha Tarar**Roll No. **16491556-013, Muhammad Hamza Javed** Roll No.**16491556-023** and **Ahmed Jamal** Roll No. **16491556-040** has successfully completed the final year project named as **Event Scheduling System** at the Department of Information Technology, University of Gujrat, to fulfill the requirement of the degree of **Bachelor of Science in Information Technology**.

Project Supervisor	Project Coordination Office Faculty of C&IT -UOG

Head of the Department

Acknowledgement

We truly acknowledge the cooperation and help make by **Mr. Haider Ali Khan**, Chairman, Department of Information Technology, University of Gujrat. He has been a constant source of guidance throughout the course of this project. We would also like to thank **Mam Sabiqa Ashraf** for his help and guidance throughout this project. We are also thankful to our friends and families whose silent support led us to complete our project.

1- Muhammad Talha Tarar.
2- Ahmed Jamal.
3- Muhammad Hamza Javed.

Date: ……………….

Abstract

This report documents the development of the system “Event Scheduling System”, an android application. System requirements are specified and best choice is made among them. Event Scheduling System is basically the system that will convert the existing manual or online event organizing/scheduling system to one complete digital system. The issues in manual system affects the people time consuming, quality of service, lack of supervision. People face problems while organizing an event. Existing system is not so efficient to help both Event Organizer and Customer to get their job done. This system will help User to organizer any kind of event by just one click. This system will also help Event Organizers to advertise and expand their business by analyzing it through our system.

Event Scheduling System helps to resolve many of the issues and digitized the system i.e. to overcome the problems in existing system and provide common platform both for user and event organizer, help event organizer to easily manage and provide services to clients etc. This system will help in ease of doing business with complete integrity and reliability.

[redacted]

List of Tables

Chapter 1: Project Proposal

Final Year Project Proposal

1. Introduction

The continuous advancement in the information and communication technology as contributed a lot to the universe, making life easier for individual, organization, institution and every field now see the importance of ICT. In the contemporary world, smartphones have made lives of people much easier and comfortable. Smart phones have made the world like a global village. Android phones are the primary reason for bringing in comfort into people's daily life. It has improved the living standards and people's approach to things.

Making contact with event organizer by going to event venue is time consuming. Similarly, it is difficult for event organizer to manage their business and take business decisions which helps them to attract more customers. Chakwal had no facility to organize event via online. Peoples had to face many difficulties to find the best place for their event. Going to event venues is more expansive, taking lot of time and very difficult. Event organizers had no such system which help them to take decisions right for their business.

"Event Scheduling System" is an android application that will facilitate the user to immediately find best place for their event in his own locality and help Event organizer to provide services to attract clients for their business.

2. Project Title: -

"Event Scheduling System".

3. Project Overview statement:

Event Scheduling System is an android application. This application will provide online platform both for Event Organizer and Customers to interact with each other. Existing system is not so efficient to help both Event Organizer and Customer to get their job done. This system will help User to organizer any kind of event by just one click. This system will also help Event Organizers to advertise and expand their business by taking important business decision with the help of our application.

4. Project Overview Statement Template

Project Title:
"Event Scheduling System"

Project Manager: Mam Sabiqa Ashraf

Project Members:

Name	Registration #	Email Address	Signature
M. Hamza Javed.	16491556-023	rajahamza710707@gmail.com	
M. Talha Tarar.	16491556-013	talhatarar4@gmail.com	
Ahmed Jamal	16491556-040	ahmedjamal69@gmail.com	

Project Goal:

Objectives:

Sr. #	
1	Help user to find best venue for their event.
2	Users can get suggestions based upon his interests.
3	User can organize events easily.
4	Help Event Organizer to sell his services.
5	Help Event Organizer in business decisions.
6	User and Event Organizer can easily communicate with each other.

Project Success criteria:
Project Success criteria is to overcome the problems of common users by helping them to organize their events in best venues and to help Event Organizer to grow their customer base.

Assumptions, Risks and Obstacles:
Smart Phones, Internet Connectivity

Organization Address (if any):

Type of project: □ Research ■ Development

Target End users:

Development Technology: ■ Object Oriented □ Structured

Platform: □ Web based □ Distributed
□ Desktop based ■ Android and Web
□ Other________________________

Approved By: Mam Sabiqa Ashraf

Date:

5. Project Goals & Objectives:

Goals: -

- To overcome the problems in existing system and provide common platform both for user and event organizer.
- Help users to easily organize their events.
- Help event organizer to easily manage and provide services to clients.

Objectives: -

- Event Organizer and User can easily interact with each other.
- Organizing and managing issues related to events can be resolved through this application.
- Provide guidelines for using this application.
- User can easily communicate with other users and event organizers.

6. High-level system components:

- User Registration/ Log In
- User can search event organizers.
- Check nearby venues using google map.
- Organize Event.
- Payment using QR code
- Event Organizer Registration/ Log In
- Provide different services related to events.
- Add different packages to attract clients.
- Accept orders.
- Analyze user interest.
- Get Payment using QR code.

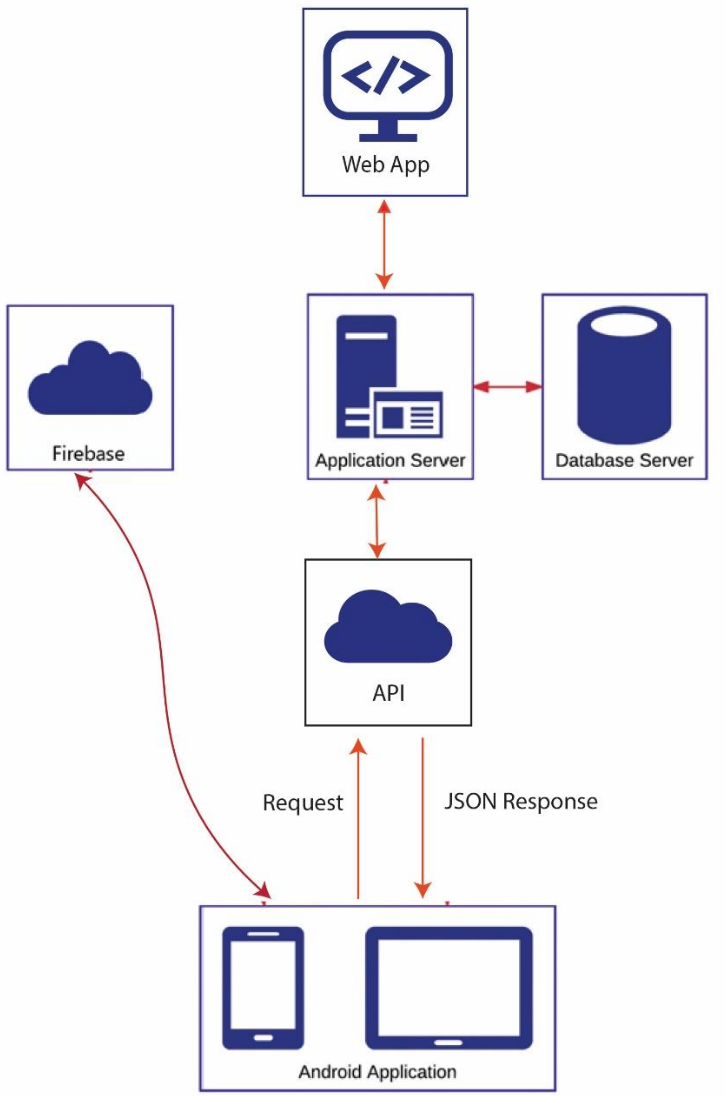
7. List of optional functional units:

- Rate Event organizer.
- Give feedback.
- Post updates about different events.
- View history.

8. Exclusions:

- Desktop Application
- IOS based mobiles

9. Application Architecture:



10.Gantt chart

N o.	***Activity***	***Duration***	***Deliverables***
1.	Requirement Gathering	3 weeks	Deliverable 1
2.	Requirement Analysis	2 weeks	Deliverable 2
3.	Design	1 month	Deliverable 3
4.	Coding	2 months	Deliverable 4
5.	Testing	2 weeks	Deliverable 5

6.	Debugging	3 weeks	Deliverable 6
7.	Implementation	1 weeks	Deliverable 7

11. Hardware and Software Specification:

- Mobile should be quad core with minimum 1GB of Ram
- Minimum android version 5.0 or above

12. Tools and Technologies used with reasoning:

- Android Studio IDE(Platform).
- Java (Language for android application development).
- XML(For application layout).
- Firebase (Main Database platform, Cloud Functions).
- HTML,CSS,Bootstrap(Construction and designing admin panel)
- JavaScript, JQuery, AJAX(Language for web based admin panel)

Chapter 2: First Deliverable

2.2. Project/Product Feasibility Report

When a project is started the first matter to establish is to assess the feasibility of a project or product. Feasibility means the extent to which appropriate data and information are readily available or can be obtained with available resources such as staff, expertise, time, and equipment. It is basically used as a measure of how practical or beneficial the development of a software system will be to you (or organization). This activity recurs throughout the life cycle.

There are many types of feasibilities:

- Technical
- Operational
- Economic
- Schedule
- Specification
- Information
- Motivational
- Legal and Ethical

2.2.1. Technical Feasibility

The system is developed in best of its approach and helps resolving the problems. The technology used for the system will be applicable to the project successfully. The team members of this project have knowledge about the technology used and also have technical expertise how to implement technology into the system.

2.2.2. Operational Feasibility

The problems arise in existing system is important to solve because it benefits the relevant people economically. This system digitalized the manual procedures and helpful in solution of many problems. End-users and project related people feel that this system will enable them to tackle the day-to-day problems easily.

2.2.3. Economic Feasibility

This system will benefit the users in terms of efficiency and financially. Efficient procedures of this system save time of end-users and helps in management of problems more effectively. This will also help in reduction of cost and increasing of revenues. The system will provide quality information about the relevant field. This system gives effective communication facility among users. People get easy access to relevant information. Our system will provide economical solutions to the users. This system is less expensive than traditional system.

2.2.4. Schedule Feasibility

Time is an important factor. We have completely made our time schedule for this project. All task activities have been divided into each person equally and also in specific time slots. We are confident to complete our project in given time according to our schedule.

2.2.5. Specification Feasibility

Requirements are gathered according to the needs of the users. We examined that smart phone is in common use. System requirements are collected on reality basis. Requirements are dependent upon the problems faced by people in manual system.

The scope of system is to improve and making the system online. This system covers all the aspects like efficiency, effectiveness and quality. It is beneficial for all relevant people.

2.2.6. Information Feasibility

Our proposed system is reliable and efficient. And provide all the features which are required by the user.

2.2.7. Motivational Feasibility

All the members of the team are completed their tasks effectively and efficiently. All the members are hard-workers. We have achieved our goal in time.

2.2.8. Legal & Ethical Feasibility

In terms of system security concerns, the system security feasibility is under taken. No infringement or liability arises from this project. The user’s privacy is completely ensured. There is a proper system for authentication and ethical standards are implemented.

The legal perspectives are also acknowledged.

2.3. Project/Product Scope

The scope of the project covers all areas relevant to area. The product not only gives the basic information as well as helps in management of the processes. The product is an android app so many of people uses it comfortably. System requirements are gathered according to the problems faced by the people in daily life. In context, it includes the users of the system i.e. service provider, clients, and others. This product also helps in efficiency and easiness of the users. it brings change in the manual procedures as well as hectic online procedures. The product has user friendly features and interfaces.

2.4. Project/Product Costing

A metric is some measurement we can make of a product or process in the overall development process. Metrics are split into two broad categories:

2.4.1. Project Cost Estimation by Function Point Analysis

Function type	Simple	Average	Complex	Count
Internal Logical File	7x25	10x20	15x10	525
External Interface File	5x2	7	10x8	90
External Input	3	4x4	6x3	34
External Output	4x2	5x3	7x4	51
External Inquiry	3x2	4x2	6	14
UFP				714

Table 2.1

Number	Complexity Weighting Factor	Value
1	Backup and recovery	4
2	Data communications	5
3	Distributed processing	0
4	Performance critical	5
5	Existing operating environment	2
6	On-line data entry	5
7	Input transaction over multiple screens	3
8	Master files updated online	3
9	Information domain values complex	4
10	Internal processing complex	4

11	Code designed for reuse	4
12	Conversion/installation in design	4
13	Multiple installations	3
14	Application designed for change	4
	Total complexity adjustment value	50

Table 2.2

Total Unadjusted Function Points (UFP) = 714
Product Complexity Adjustment (PC) = 0.65 + (0.01 *50) = 1.15

Total Adjusted Function Points (FP) = UFP * PC =714 * 1.15= 821.1

Language Factor (LF) for Java assumed as = 53

Source Lines of Code (SLOC) = FP * LF = 821.1*53 = 43,518.3

Productivity=FP/PM=821.1/18=45.61

The formulas are given as follows:

- Cost / FP = labor rate / productivity parameter
  Cost/FP = 20000/45.61
  =438.50
- Total Project Cost = FP est. * (cost / FP)
  = 821.1*438.50
  = 360,050
- Total Estimated Effort = FP est. / productivity parameter
  =821.1/45.61
  = 18
  TEF = 18.00 person/month

2.4.2. Project Cost Estimation by using COCOMO’81 (Constructive Cost Model)

Basic COCOMO

Type	**Effort**	**Schedule**
Organic	PM= 2.4 (KLOC)1.05	TD= 2.5(PM)0.38
Semi-Detached	PM= 3.0 (KLOC)1.12	TD= 2.5(PM)0.35
Embedded	PM= 2.4 (KLOC)1.20	TD= 2.5(PM)0.32

PM= person-month (effort)

KLOC= lines of code, in thousands

TD= number of months estimated for software development (duration)

Intermediate COCOMO

Type	**Effort**
Organic	PM= 2.4 (KLOC)1.05 x M
Semi-Detached	**PM= 3.0 (KLOC)1.12 x M**
Embedded	PM= 2.4 (KLOC)1.20 x M

KLOC= 43.518

Effort Applied (E) = $a_b(\text{KLOC})^{b}{}_{b}$**[man-months]**
3.0(43.518)$^{\mathbf{1.12}}$ **= 205.13**

Development Time (D) = c_b (Effort Applied) $^{d}{}_{b}$**[months]**
2.5(205.13)$^{\mathbf{0.35}}$ **= 16.11**

People required (P) = Effort Applied / Development Time **[count]**

205.13/16.11= 12.73

2.5. Task Dependency Table

Activity	Activity Name	Dependencies
A	Requirement gathering	None
B	Project proposal	A
C	Data Base Modeling	A
D	User Interface Design	A,B
E	Web Development	B, C
F	Mobile App Development	B, C, E
G	Software Testing	E, F
H	Final Documentation	G
I	Deployment	None

Table 2.3

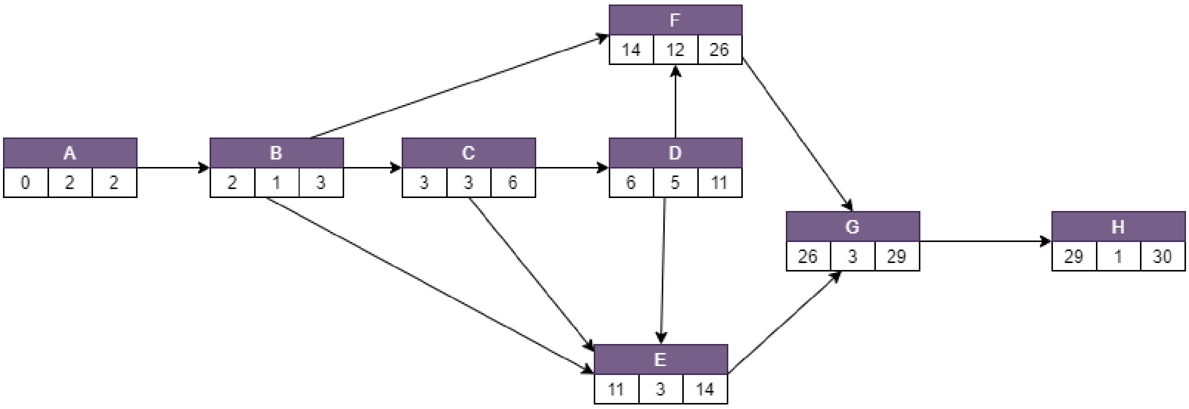
2.6. CPM - Critical Path Method

Activity	Activity Name	Predecessor	Duration (Week)
A	Requirement gathering	None	2
B	Project proposal	A	1
C	Data Base Modeling	C	3
D	User Interface Design	D	5
E	Web Development	B, C, D	3
F	Mobile App Development	B, C, E	12
G	Software Testing	E, F	3
H	Final Documentation	G	1

Table 2.4

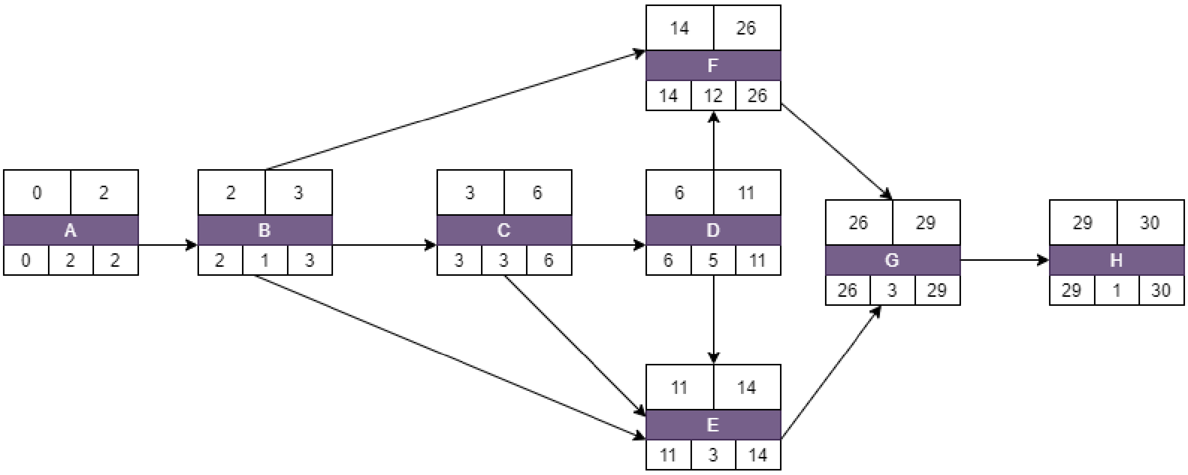
In 1957, DuPont developed a project management method designed to address the challenge of shutting down chemical plants for maintenance and then restarting the plants once the maintenance had been completed.

NETWORK:



Network Diagram for the above-mentioned activities

CPM:



Activity	Duration	ES	EF	LS	LF
A	2	0	2	0	2
B	1	2	3	2	1
C	3	3	6	3	3
D	5	5	8	5	13
E	3	8	14	8	13
F	12	14	22	14	17
G	3	22	25	22	13
H	1	26	27	26	

Table 2.5

The parameters and slacks are calculated as follows:

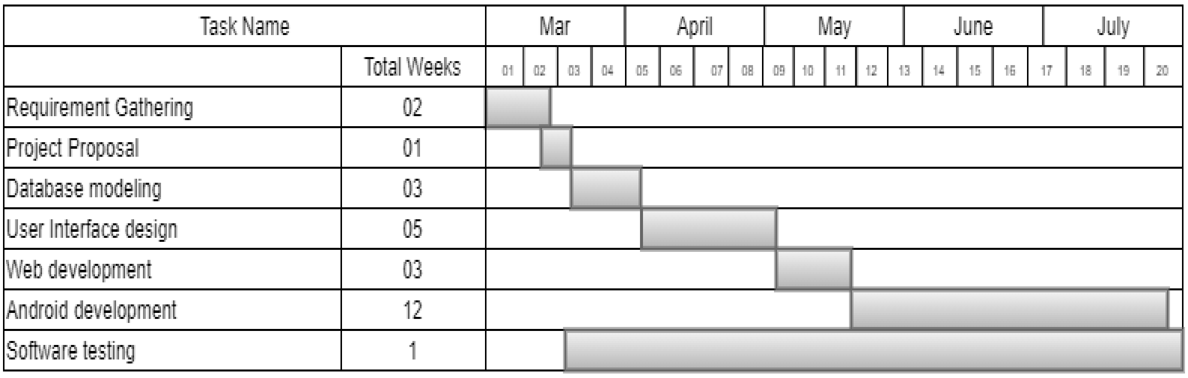
The critical path is:

A, B, C, D, E, F, G, H,

2.7. Gantt Chart

The Gantt chart enumerates the activities to be performed on the vertical axis and their corresponding duration on the horizontal axis. It is possible to schedule activities by either early start or late start logic. In the early start approach, each activity is initiated as early as possible without violating the precedence relations. In the late start approach, each activity is delayed as much as possible as long as the earliest finish time of the project is not compromised.

Based on the Work Breakdown Structure (WBS), a timeline or Gantt chart showing the allocation of time to the project phases or iterations should be developed. This Gantt chart would identify major milestones with their achievement criteria. It must contain duration estimation of all the necessary activities to be carried out during the project development along with the human resources responsible for the respective tasks. Activity dependencies are also required to be mentioned in it.



Task Name	Total Weeks
Requirement Gathering	02
Project Proposal	01
Database modeling	03
User Interface design	05
Web development	03
Android development	12
Software testing	1

2.8. Introduction to Team Member and their Skill-set

Table 2.6

Name	Skill Set	Responsibility
Muhammad Hamza Javed Talha Tarar Ahmed Jamal	Android Development with java support, UI Design	Android development java functionality + UI design
Muhammad Hamza Javed Talha Tarar Ahmed Jamal	Web developer, Database , Documentation	Developing Admin Panel + Database + Documentation
Muhammad Hamza Javed Talha Tarar Ahmed Jamal	Web developer, Database , Documentation	Developing Admin Panel + Database + Documentation

2.9. Task and Member Assignment Table

Table 2.7 Assignment Table

Task	Task Name	Members
T1	Requirement gathering	Muhammad Hamza Javed Talha Tarar Ahmed Jamal
T2	Project proposal	Muhammad Hamza Javed Talha Tarar Ahmed Jamal
T3	Data Base Modeling	Muhammad Hamza Javed Talha Tarar Ahmed Jamal
T4	User Interface Design	Muhammad Hamza Javed Talha Tarar Ahmed Jamal
T5	Web Development	Muhammad Hamza Javed Talha Tarar

		Ahmed Jamal
T6	Mobile App Development	Muhammad Hamza Javed Talha Tarar Ahmed Jamal
T7	Software Testing	Muhammad Hamza Javed Talha Tarar Ahmed Jamal
T8	Final Documentation	Muhammad Hamza Javed Talha Tarar Ahmed Jamal
T9	Deployment	Muhammad Hamza Javed Talha Tarar Ahmed Jamal

Task durations and dependencies

M1: Muhammad Hamza Javed
M2: Talha Tarar
M3: Ahmed Jamal

Table 2.8

Task	**Duration (days)**	**Dependencies**	**Members Assignment**
T1	20		**M1,M2, M3**
T2	12	T1	**M1, M2, M3**
T3	7	T2	**M1, M2,M3**
T4	25	T3	**M1, M2, M3**
T5	40	T2, T3, T4	**M1, M2,M3**
T6	55	T2, T3, T5	**M1.M2,M3**
T7	19	T5, T6	**M1, M2, M3**
T8	2	T7	**M1, M2,M3**

Allocation of People to Activities:

Table 2.9 Allocation of People to Activities

Task	**Engineer**
T1	Talha Tarar Ahmed Jamal
T2	Muhammad Hamza Javed Talha Tarar Ahmed Jamal
T3	Muhammad Hamza Javed Talha Tarar Ahmed Jamal
T4	Hamza Javed
	Ahmed Jamal

T5	Hamza Javed Talha Tarar
T6	Hamza Javed
T7	Muhammad Hamza Javed Talha Tarar Ahmed Jamal
T8	Talha Tarar Ahmed Jamal
T9	Muhammad Hamza Javed Talha Tarar Ahmed Jamal

2.10. Tools and Technology with reasoning

- **Android Studio**

In android studio which provide us xml front-end and back-end coding

- **Java**

  Language for android application development

- **XML**

  For application layout

- **Firebase Firestore**

  Main database platform

- **Firebase**

  Secondary database and for cloud messaging

- **HTML, CSS, Bootstrap**

  Construction and designing admin panel

- **JavaScript, jQuery, AJAX, PHP**

  Language for web-based admin panel

2.11. Vision Document

This is the vision and scope document of Event Scheduling System. This will cover the problem and vision statement including project background, list of users, stake holders, risks, vision statement, scope and list of features.

Project Background: Pakistan in the past time, event organizing was not much efficient. Finding appropriate event venue or service related to any kind of event was not that much easy. Even service providerlacks the resources of advertisement for finding clients related to their business. Both hands the clients and service providers were facing resistance. So, finding a service provider according to your best interest and location was task near to difficult without previous experience. Here arises a need of such system that ensures the quality of service with best service provider or event organizer and aid in organizing events or availing services related for their events by just one click.

Key Users:

- Service Provider.
- Clients.

Stake-Holders:

- Service Provider.
- Client.
- Admin
- Development Team

List of Risks:

- To ensure the quality of tasks in limited time
- Having limited development time
- To include extra functionalities that will extend the schedule
- Team members need to learn unfamiliar tools and technologies
- Detailed reporting could make more development time

Vision Statement: This is an android application that will facilitate the user to immediately get solution of organizing event problems in his own locality. This app will solve many of the regular event organizing, services needed for events for the common user. We will be digitalized the existing system. This will help the users in saving their time and make the procedures reasonable and easy. It will play important role in progress of this field due to easiness and convenient system.

Scope: Event SchedulingSystem is an android application, that will provide online assistance to clients and service providers in term of organizing any kind of event. Existing manual and online systems are not very efficient. They do not provide ease to users. Through this system, service provider or event organizer will be able to provide quality service. This app will help to get rid of conventional event organizing system. Clients do not need to travel a long distance searching for venue for their events.

List of Features:

- Overcome manual system and provide digital system.
- Get easy allocation of services.
- Time based events.

- Event Organizer of own choice .
- Client can organize any kind of event.
- Provide guidelines for using this application

2.12. Risk List

- Lost internet connection
- Connect failed from database
- As the project progresses more and more features that were not identified at the beginning of the project emerge that threaten estimates and timelines
- Mobile should be quad core with minimum 2GB of Ram
- Our App requires android version 7.0 or above to be installed

2.13. Product Features/ Product Decomposition

- To overcome manual system and provide more efficient digital system.
- To provide platform for both service providers and clients.
- To provide services related to event just on single click.
- Service Provider and client can communicate easily.
- Service Provider can easily manage different orders through our application.
- Provide Guidelines for using this system.
- It will make easier for client to get better services for their event.

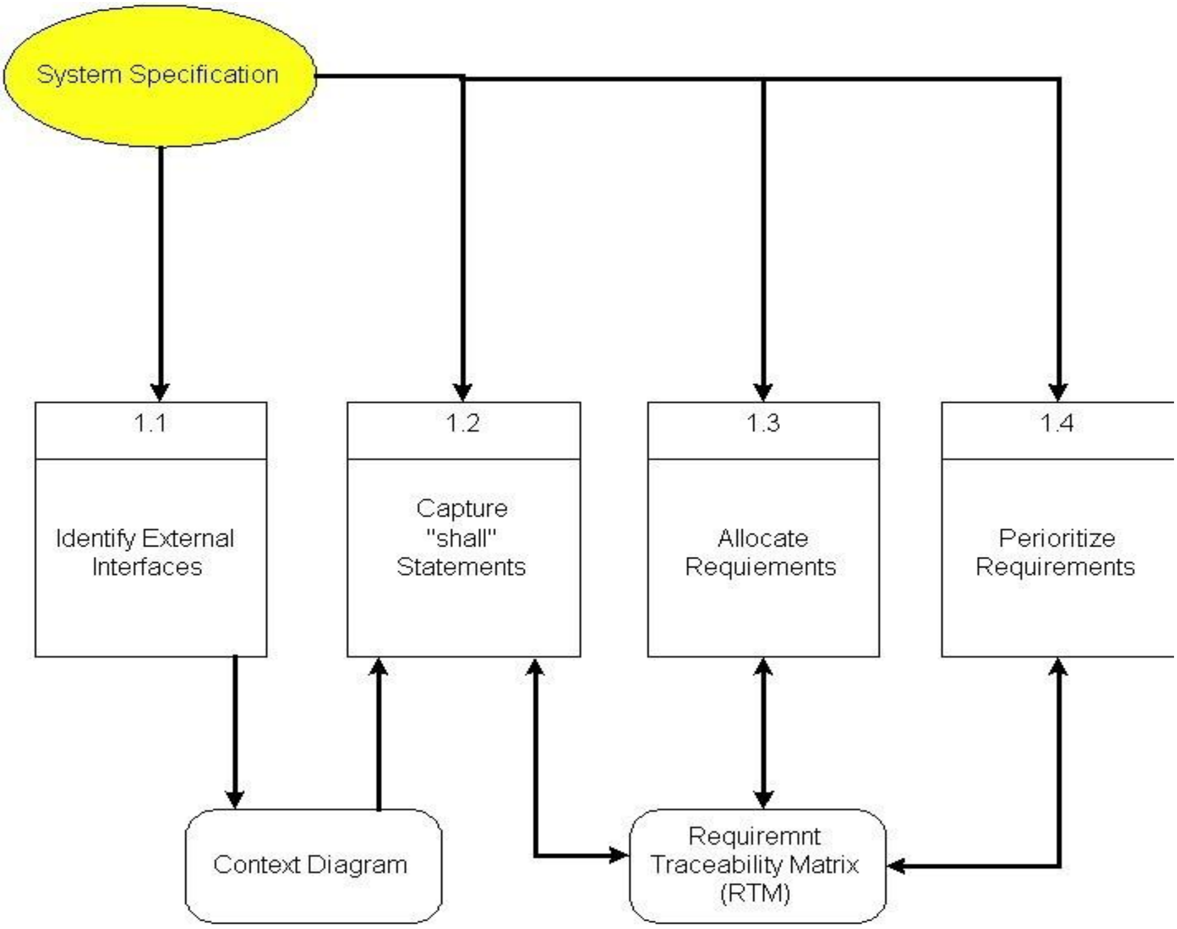
Chapter 3: Software Requirement Specification (For Object Oriented Approach)

3.1 Introduction:

Event Scheduling System is an android application, that will provide better event organizing system to service provider and clients in term of better online/offline service.

Requirements engineering process provides the appropriate mechanism for understanding what the customer wants, analyzing need, assessing feasibility, negotiating a reasonable solution, specifying the solution unambiguously, validating the specification and managing the requirements as they are transformed into an operational system. The task of capturing, structuring, and accurately representing the user's requirements so that they can be correctly embodied in systems which meet those requirements (i.e. are of good quality).

- Requirements elicitation
- Requirements analysis and negotiation
- Requirements specification
- System modeling
- Requirements validation
- Requirements management



Here, requirements specification is to be discussed. Requirements specification would lead to the following four steps:

- Identify external interfaces
- Development of context diagram
- Capture “shall statements
- Allocate requirements
- Prioritize requirements
- Development of requirements traceability matrix

3.1.1 Systems Specifications

The following are the clauses that must be included while describing the system specifications.

Introduction: According to survey in 2018, there are 3.1 billion active users of android in world. Android apps are the future as they provide more ease of use to common people than web-based systems.

"EVENT SCHEDULING SYSTEM" is an android application that will facilitate the service provider and client to interact and aid in event organizing process. Event Organizers can offer different packages while clients can avail packages of their choice. Event Organizer and Clientscan communicate with each other through this application.

Existing System: Existing manual and online systems are not very efficient. They do not provide ease to usersfew of them may have got benefits from online services. Although now a day, web-based event organizing platforms are available and working very efficiently. But the problem is that general public does not find ease of use of web system and also not aware of them. The existing android apps provide event organizing services in limited cities on small scale. But the online service provider platform does not match the standard of online platform.

Scope of the System: Event Organizing System is an android application that will facilitate the service provider/ event organizer and client to interact and aid event organizing process. Service Provider/ Event Organizer can provide services of their choice. Clientscan get theses services by just one click.

Summary of Requirements: (Initial Requirements): Event Organizing System is an online Android based mobile application which needs the following initial requirements:

- Android mobile with at least android 7.0 or higher
- Mobile should be quad core with minimum 2 GB of Ram
- Internet Connectivity

3.1.2. Identifying External Entities

The Identification of External Entities is done in two phases.

Phase I: Over Specify Entities from Abstract:

- Service Provider/ Event Organizer
- Client
- Admin
- Developer

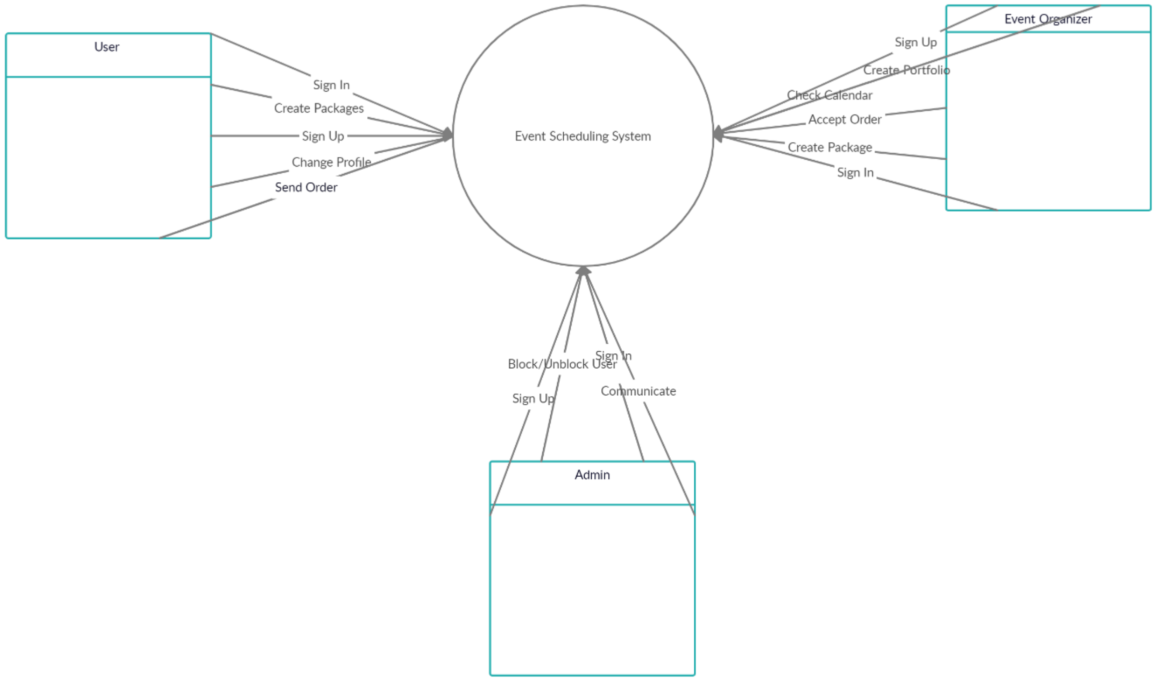
- Database
- Payment

On the basis of the Abstract, one might identify the entities from the problem.

Phase II: Perform Refinement:

- Service Provider/ Event Organizer
- Client
- Admin

3.1.3. Context Level Data Flow Diagram:



3.1.4. Allocate Requirements

Table: 3.1

Req #	Initial Requirements	Use Case Name
1	Admin, service provider, client, "shall" login.	UC_ Login
2	Admin "shall" block & unblock service provider and client.	UC_ Block/ Unblock
3	Admin "shall" view information about service provider and client	UC_ View Detail

4	User “shall” place order	UC_ Place Order
5	Service Provider, Client “shall” update profile	UC_ Profile Update
6	User “shall” view portfolio	UC_ View Portfolio
7	Service Provider “shall” check, accept and cancel order	UC_ Accept Order UC_ Cancel Order
8	Service Provider “shall” check calendar to check order dates.	UC_ Check Calendar
9	Service Provider “shall” update its portfolio.	UC_ Update Portfolio
10	User “shall” create custom packages	UC_ Create Custom Order
11	User “shall” communicate with event organizer and vice versa.	UC_ Communication
12	Service Provider, Admin, Client “shall” logout	UC_ Logout
13	User “shall” rate service provider and its packages	UC_ Rate Event Organizer
14	Service Provider, client “shall” rate application.	UC_ Rate App

3.1.5. Priorities Requirements:

Table: 3.2

Req #	Rank	Initial Requirements	Use Case ID	Use Case Name
1	Highest	Admin, service provider, user, “shall” login.	UC_ 1	UC_ Login

2	Highest	Admin “shall” block and unblock service provider and client.	UC_ 2	UC_ Block/Unblock
3	Highest	User “shall” place orders	UC_ 3	UC_ Place Orders
4	Highest	User, Service Provider and “shall” Rate app	UC_4	UC_ Rate app
5	Highest	User, Service Provider, Admin “shall” logout	UC_ 5	UC_ Logout
6	Medium	Service Provider “shall” check, accept, cancel order	UC_6 UC_7	UC_ Accept Order UC_ Cancel Order
7	Medium	Service Provider, Client, “shall” update profile	UC_8	UC_ Profile Update
8	Medium	Service Provider “shall” create and update its portfolio.	UC_9 UC_10	UC_ Create Portfolio. UC _ Update Portfolio.
9	Medium	Client “shall” create custom packages.	UC_12	UC_ Custom Package
10	Medium	Service Provider “shall” check calendar to view orders.	UC_ 13	UC_ Check Calendar

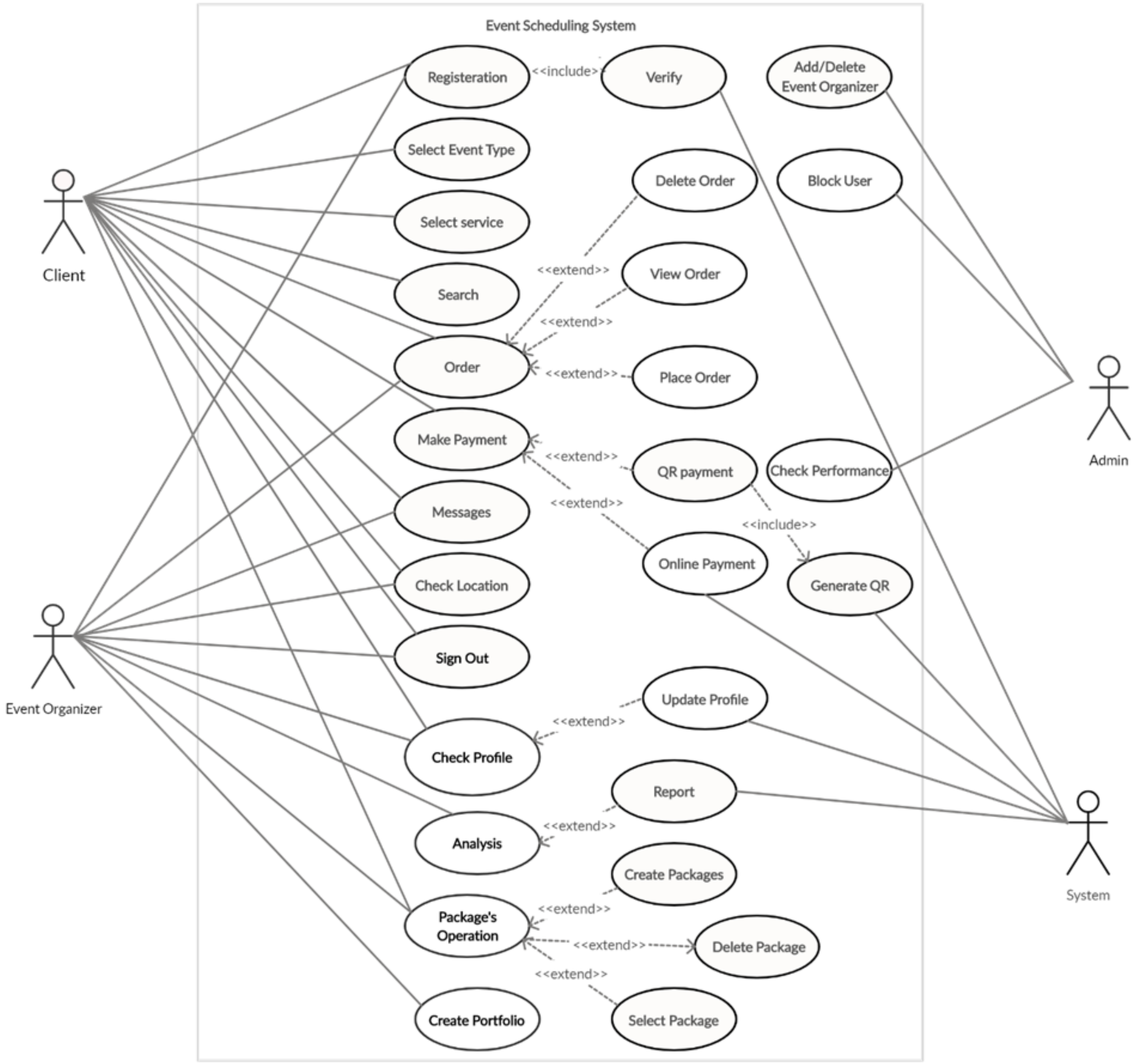
3.1.6. Requirements Traceability Matrix:

Table: 3.3

Sr#	Req#	System Specification Text	Build	Use Case Name	Category

1	1	Service Provider, client, “shall” login.	B1	UC_ Login	Business
2	2	Admin “shall” block/ unblock service provider and clients	B1	UC_ Block/Unblock	Business
3	3	User “shall” Place order	B2	UC_ Place Order	Business
4	4	Service Provider “shall” check , accept or cancel order	B2	UC_ Accept Order UC_ Cancel Order	Business
5	5	User, Service Provider “shall” Rate app	B3	UC_ Rate app	Business
6	6	Service Provider “shall” create and update portfolio	B4	UC_ Create Portfolio UC_ Update Portfolio	Business
7	7	Client “shall” create custom packages	B4	UC_ Create Custom Packages	Business
8	8	User, Service Provider “shall” update profile	B6	UC_ Profile Update	Business
9	9	User “shall” order different services	B7	UC_ Order Services	Business

3.1.7. High Level Use Case Diagram:



3.1.9. Use Case Description

Table 3.4

Use Case Template	
Use Case ID:	UC_1
Use Case Name:	UC_Login
Goal/Purpose:	To login into the application
Actors:	Service Provider, Client and Admin
Pre-Conditions	Install Application
Post-Conditions:	Provide username, contact number.etc
Basic Flow:	Main screen

Alternate Flow(s):	None
Exception Flow(s):	Display errors for Invalid password and email

Table: 3.5

Use Case Template	
Use Case ID:	UC_2
Use Case Name:	UC_ Block/Unblock
Goal/Purpose:	To block/ unblock service provider or client from application
Actors:	Admin
Pre-Conditions	None
Post-Conditions:	Provide username, contact number.etc
Basic Flow:	Screen
Alternate Flow(s):	None
Exception Flow(s):	None

Table: 3.6

Use Case Template	
Use Case ID:	UC_3
Use Case Name:	UC_ Place Order
Goal/Purpose:	To place order
Actors:	User/Client
Pre-Conditions	Install Application UC_ Login
Post-Conditions:	None
Basic Flow:	order screen
Alternate Flow(s):	None
Exception Flow(s):	None

Table: 3.7

Use Case Template	
Use Case ID:	UC_4
Use Case Name:	UC_ Profile Update
Goal/Purpose:	To update profile
Actors:	User, Admin and Service Provider

Pre-Conditions	UC_ Login
Post-Conditions:	None
Basic Flow:	Profile screen
Alternate Flow(s):	None
Exception Flow(s):	None

Table: 3.8

Use Case Template	
Use Case ID:	UC_5
Use Case Name:	UC_ Create Custom Packages
Goal/Purpose:	To login into the application
Actors:	User
Pre-Conditions	UC_ Login
Post-Conditions:	Send to service provider/ event organizer
Basic Flow:	Custom Packages screen
Alternate Flow(s):	None
Exception Flow(s):	None

Table: 3.9

Use Case Template	
Use Case ID:	UC_6
Use Case Name:	UC_ Check Calendar
Goal/Purpose:	To check calendar to check order schedule
Actors:	Service Provider/ Event Organizer
Pre-Conditions	UC_ Login UC_ Place Order
Post-Conditions:	None
Basic Flow:	Navigation screen
Alternate Flow(s):	None
Exception Flow(s):	None

Table 3.10

Use Case Template	
Use Case ID:	UC_7

Use Case Name:	UC_ Logout
Goal/Purpose:	To logout from application
Actors:	Service Provider, Client and Admin
Pre-Conditions	UC_ Login
Post-Conditions:	Close app
Basic Flow:	Logout Screen
Alternate Flow(s):	None
Exception Flow(s):	None

Table 3.11

Use Case Template	
Use Case ID:	UC_8
Use Case Name:	UC_ Accept Order
Goal/Purpose:	To accept the order of event from client
Actors:	Service Provider/ Event Organizer
Pre-Conditions	UC_ Login UC_ Place Order
Post-Conditions:	Order accepted notification
Basic Flow:	Order screen
Alternate Flow(s):	None
Exception Flow(s):	None

Table 3.12

Use Case Template	
Use Case ID:	UC_9
Use Case Name:	UC_ Cancel Order
Goal/Purpose:	To cancel the orders of client
Actors:	Store
Pre-Conditions	UC_ Login UC_ Place Order
Post-Conditions:	Order cancelled notification
Basic Flow:	Order screen
Alternate Flow(s):	None

Exception Flow(s):	None

Table 3.13

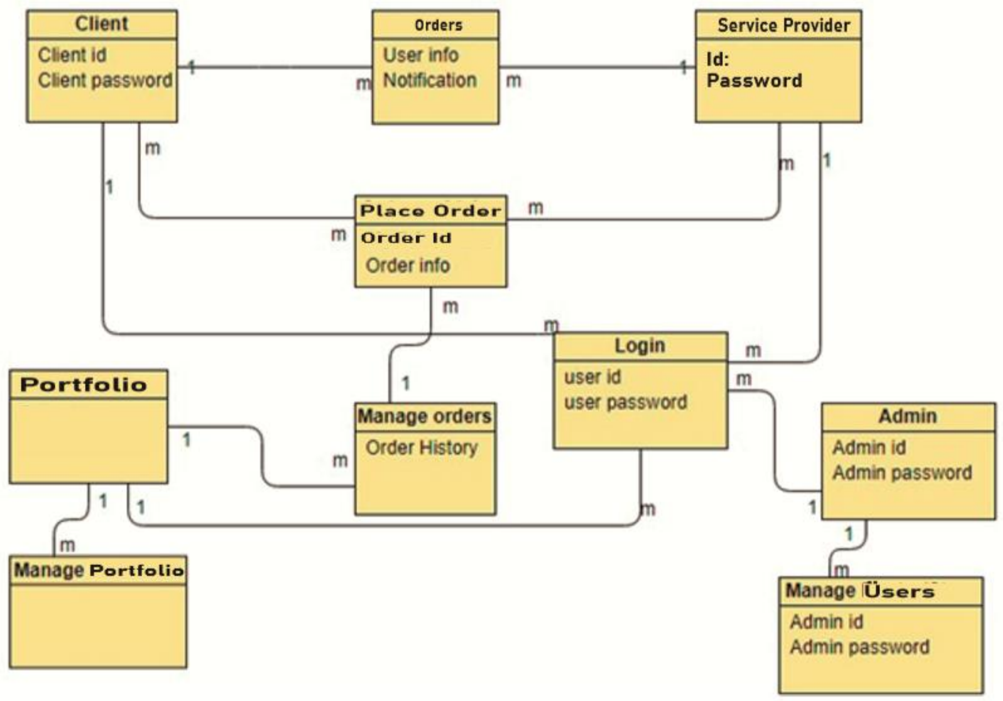
Use Case Template	
Use Case ID:	UC_10
Use Case Name:	UC_ Rate app
Goal/Purpose:	To rate the app
Actors:	Event Organizer/ Service Provider, Client
Pre-Conditions	UC_ Login UC_ Place Orders
Post-Conditions:	None
Basic Flow:	Main Screen
Alternate Flow(s):	None
Exception Flow(s):	None

Chapter 4 Design Document (For Object Oriented Approach)

4.1. Introduction:

Third deliverable is all about the software design.

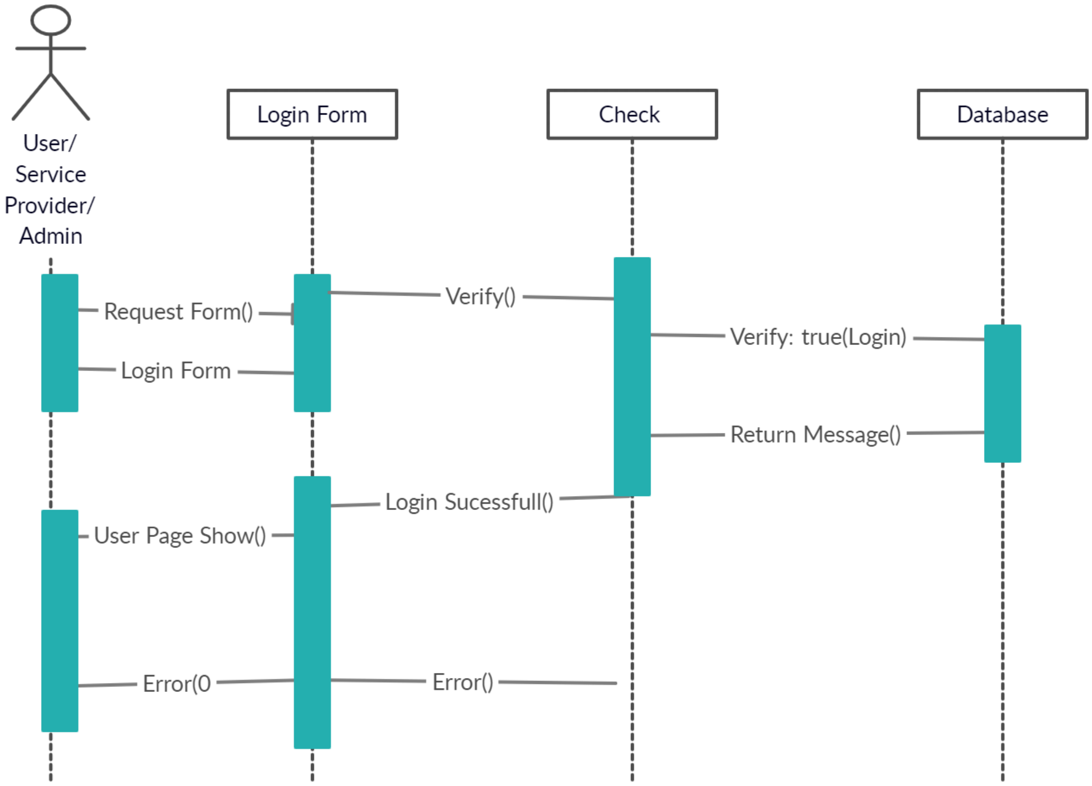
4.2. Domain Model



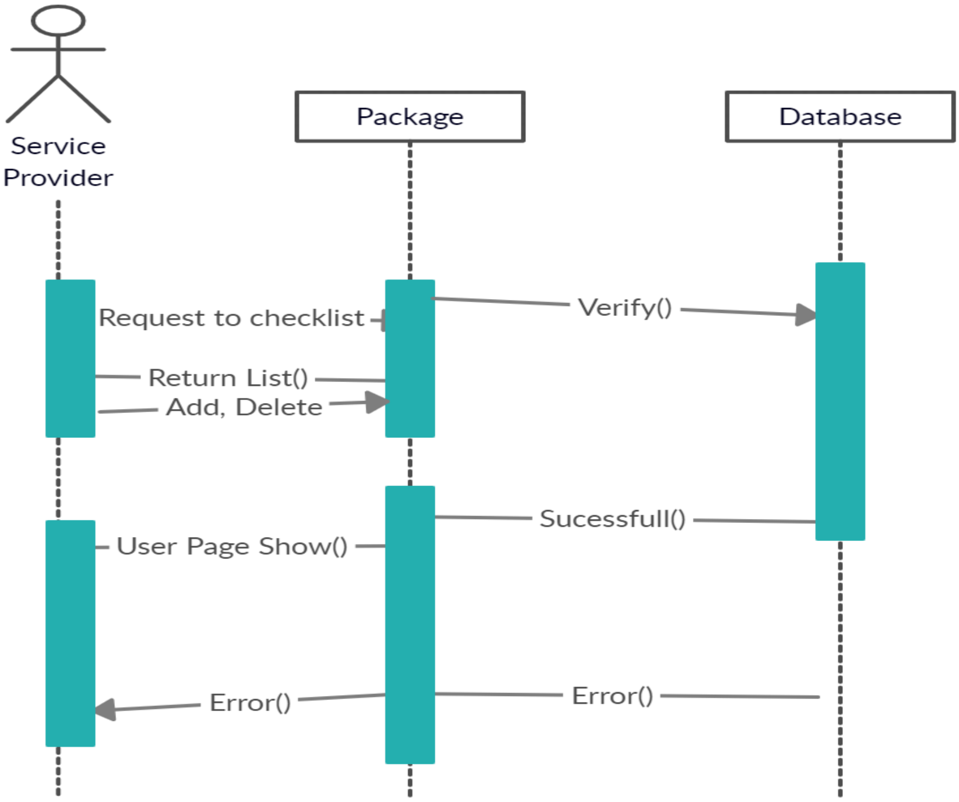
4.3. Sequence Diagram

A Sequence diagram depicts the sequence of actions that occur in a system. The invocation of methods in each object, and the order in which the invocation occurs is captured in a Sequence diagram.

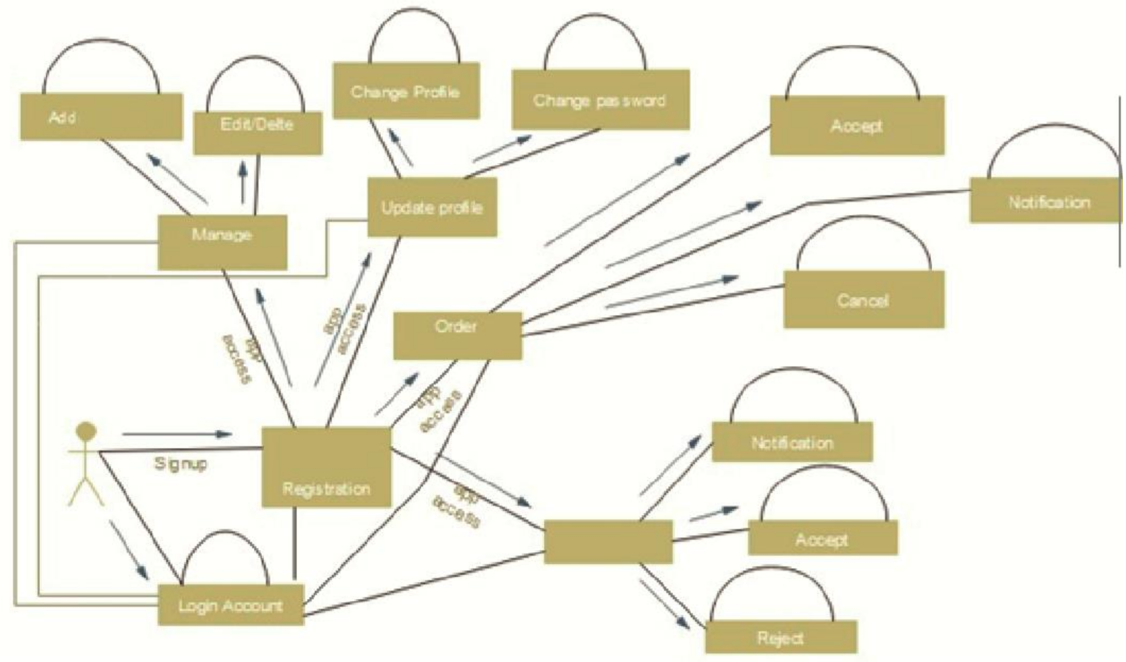
Login Sequence diagram



Add Package



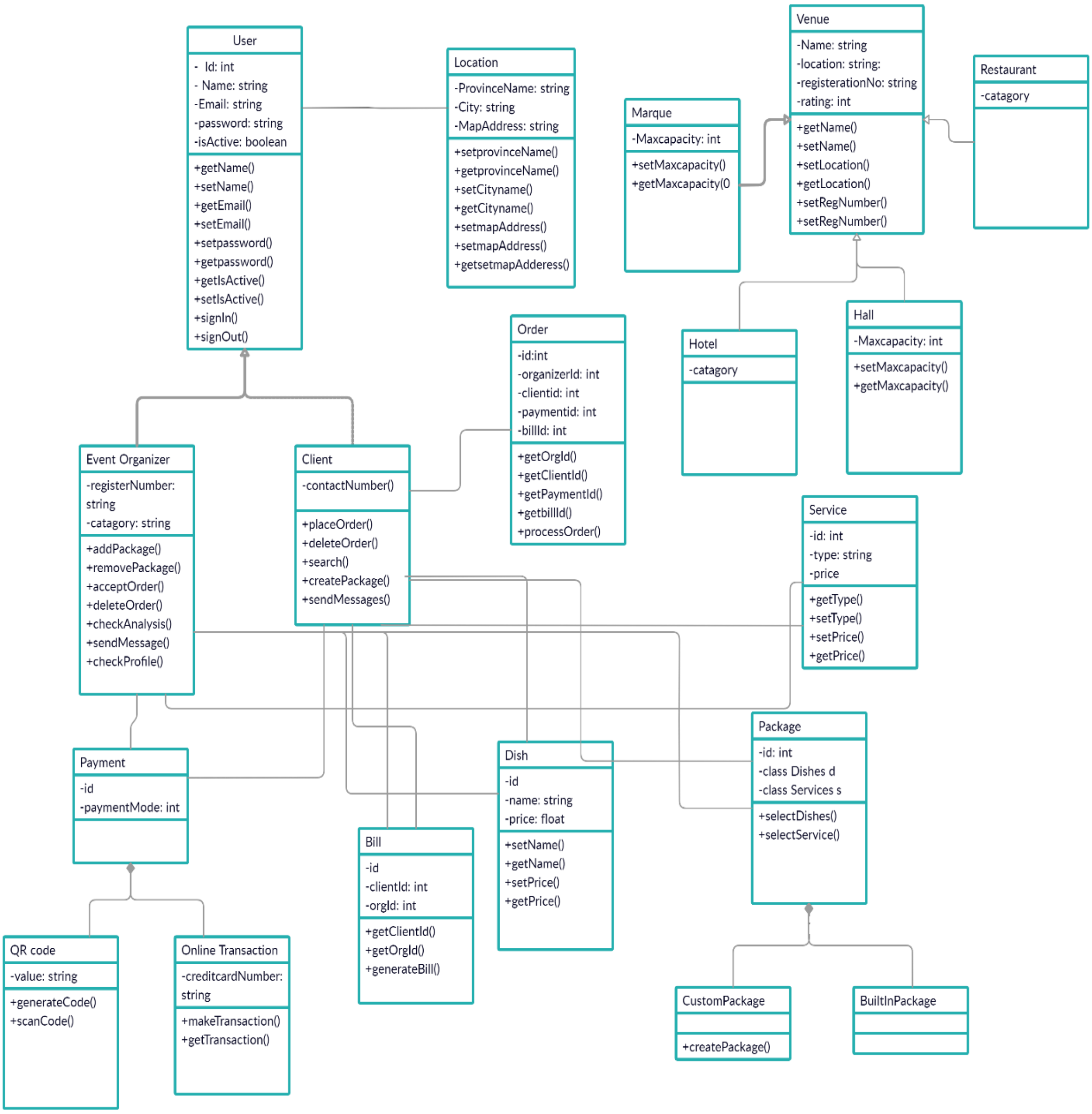
3.5. Collaboration Diagram



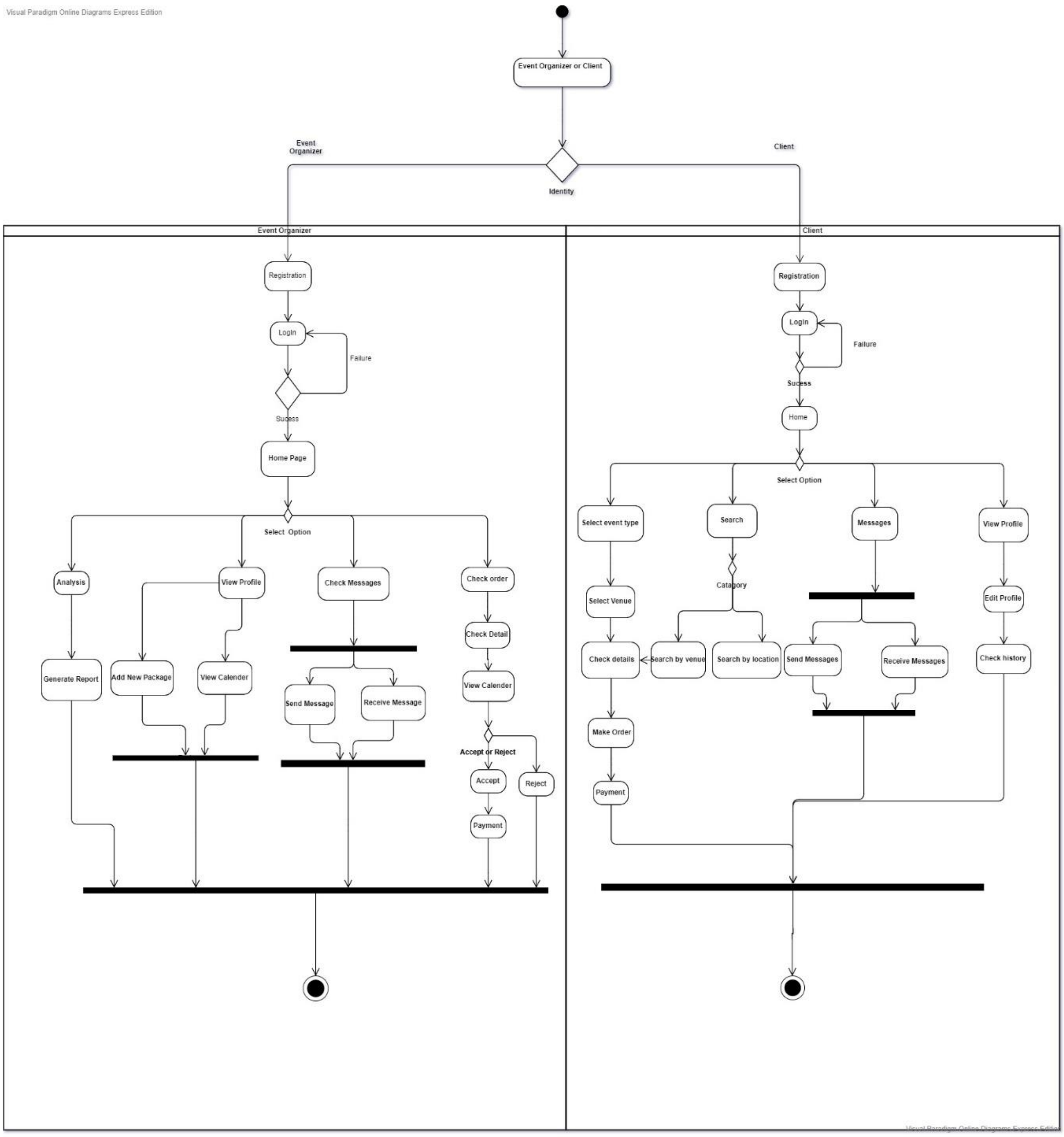
4.5. Design Class Diagram

Classes are the work-horses of the design effort—they actually perform the real work of the system. The other design elements—subsystems, packages and collaborations simply describe how classes are grouped or how they interoperate.

Active classes are design classes, which coordinate and drive the behavior of the passive classes - an active class is a class whose instances are active objects, owning their own thread of control.



4.6. State chart diagram


Visual Paradigm Online Diagrams Express Edition
Event Organizer or Client
Event Organizer
Client
Identity
Event Organizer
Client
Registration
Login
Failure
Sucess
Home Page
Select Option
Analysis
View Profile
Check Messages
Check order
Generate Report
Add New Package
View Calender
Send Message
Receive Message
Check Detail
View Calender
Accept or Reject
Accept
Reject
Payment
Registration
Login
Failure
Sucess
Home
Select Option
Select event type
Search
Messages
View Profile
Catagory
Select Venue
Check details
Search by venue
Search by location
Send Messages
Receive Messages
Edit Profile
Check history
Make Order
Payment
Visual Paradigm Online Diagrams Express Edition

Chapter 5 User Interface Design

Login Screen:

Profile Setting:

Doctor Appointment:

About us:

Animal List:

Change Language:

Feedback:

Add Problem:

Diseases List:

Chapter 6 Deliverable (Software Testing)

6.1 Introduction:

This deliverable is based on the IEEE standard of software testing i.e. IEEE SOFTWARE TEST DOCUMENTATION Std 829-1998. This standard describes a set of basic test documents that are associated with the dynamic aspects of software testing (i.e., the execution of procedures and code). The standard defines the purpose, outline, and content of each basic document. While the documents described in the standard focus on dynamic testing, several of them may be applicable to other testing activities (e.g., the test plan and test incident report may be used for design and code reviews). This standard may be applied to commercial, scientific, or military software that runs on any digital computer. Applicability is not restricted by the size, complexity, or criticality of the software. However, the standard does not specify any class of software to which it must be applied. The standard addresses the documentation of both initial development testing and the testing of subsequent software releases. For a particular software release, it may be applied to all phases of testing from module testing through user acceptance. However, since all of the basic test documents may not be useful in each test phase, the particular documents to be used in a phase are not specified. Each organization using the standard will need to specify the classes of software to which it applies and the specific documents required for a particular test phase.

The standard does not call for specific testing methodologies, approaches, techniques, facilities, or tools, and does not specify the documentation of their use. Additional test documentation may be required (e.g., code inspection checklists and reports). The standard also does not imply or impose specific methodologies for documentation control, configuration management, or quality assurance. Additional documentation (e.g., a quality assurance plan) may be needed depending on the particular methodologies used.

Following are standard artifacts, which must be included in this deliverable:

1. Test Plan
2. Test Design Specification
3. Test Case Specification
4. Test Procedure Specification
5. Test Item Transmittal Report
6. Test Log
7. Test Incident Report
8. Test Summary Report

6.2. Test plan

6.2.1. Purpose

Test planning, the most important activity to ensure that there is initially a list of tasks and milestones in a baseline plan to track the progress of the project.

To prescribe the scope, approach, resources, and schedule of the testing activities, to identify the items being tested, the features to be tested, the testing tasks to be performed, each tester is responsible for each task, and the risks associated with this plan.

6.2.2. Outline

A test plan shall have the following structure:

a. Test plan identifier
b. Introduction
c. Test items
d. Features to be tested
e. Features not to be tested
f. Approach
g. Item pass/fail criteria
h. Suspension criteria and resumption requirements
i. Test deliverable
j. Testing tasks
k. Environmental needs
l. Responsibilities
m. Staffing and training needs
n. Schedule
o. Risks and contingencies
p. Approvals

6.2.2.1. Test Plan Identifier

The Identifier for test plan is **Event Organizing System AFTP-1**The abbreviation for identifier is as:

Event Organizing System App

AFTP 1 All Feature Test Plan Version 1

6.2.2.2. Introduction

Software test planning is the most core part of the testing phase on which all the other phases depends in one or another way.

Test planning phases defines and elaborate following points:

What tester has to perform in testing, Standards of quality to use in testing, Resources to be employed for testing, Schedule and time scale for the testing phase

(5) Describes all the risk and contingencies which involved in testing and how they will be overcome. In multilevel test plans, each lower-level plan must reference the next higher-level plan

6.2.2.3. Test Items

Identify the test items including their version/revision level. Also specify characteristics of their transmittal media that impact hardware requirements or indicate the need for logical or physical transformations before testing can begin (e.g., programs must be transferred from tape to disk).

Supply references to the following test item documentation, if it exists:

a) Requirements specification

b) Design specification

c) Users guide

d) Operations guide

e) Installation guide

Reference any incident reports relating to the test items. Items that are to be specifically excluded from testing may be identified.

6.2.2.4. Features to be Tested

- User/Service Provider/Admin login
- Block/Unblock Service Provider/Client
- Place Order
- Accept and Cancel Order
- Create Custom Package
- Create and Update Portfolio
- Check Calendar
- View Profile
- Communicate
- Logout

6.2.2.5. Features not to be Tested

There will be no any feature or module that will not be tested.

.6.2.2.6. Approach

Test will be conducted as per documented test cases. Each member will test each feature and mark each case as Pass/Fail. Each tester will note the actual result and all relevant details. Once all tests will complete, the test manager will review the test report to the team accordingly.

6.2.2.7. Item Pass/Fail Criteria

There will be different criteria for the Unit testing and Instrumentation testing for the passing and failure of a testing. Moreover, we will be using test driven approach in our development of app. It means we are going to first write test and the execute them and after test will be failed, we will write the code required to pass the test. After that we will refactor and again test till, we pass all the test which are designed and written.

6.2.2.8. Suspension criteria and Resumption Requirements

Testing should be paused immediately if either system experiences the internet issue not performing the required tasks by web or by app.

6.2.2.9. Test Deliverable

Identify the deliverable documents. The following documents should be included:

a.Test plan;

b. Test design specifications;

c. Test case specifications;

d. Test procedure specifications;

e. Test item transmittal reports;

f. Test logs;

g. Test incident reports;

h. Test summary reports.

6.2.2.10. Testing Tasks

Following tasks should be completed before the testing:

Latest Android Studio Installed.

Add dependencies inside Android Studio needed for the test execution.

Design and plan test in order to write and execute them.

6.2.2.11. Environmental Needs

As mentioned earlier we are going to perform testing in android using unit test and instrumentation test. Both these types of tests need different platforms which are as:

Unit Testing

For execution of unit test, we need Java virtual machine to be installed in our computer.

Instrumented Testing

We need a physical android device or an android emulator installed at Android studio to smoothly run instrumented tests.

6.2.2.12. Responsibilities

Each member is responsible for designing, preparing documentation and executing the system properly. The test manager is responsible for testing, managing and resolving the issues of the project.

6.2.2.13 Staffing and Training Needs

To apply systematic testing methods, test team require a complete understanding of software testing. To apply test in android and to understand them a person requires complete knowledge of testing components provided by android. Training for the team members can be organized to get understanding of testing frameworks.

6.2.2.14. Schedule

The testing will take 4- 5 days. Some components and their features will be tested in a day.

6.2.2.15. Risks and Contingencies

If the first component testing is not completed within a day it can be delay. After bug fixes the testing will be performed. If the tester doesn’t have the basic understanding of IOT, testing will be delayed or not conducted properly.

6.2.2.16 Approvals

Name: Sabiqa Ashraf

Date:

Signature:

6.3. Test Design Specification

6.3.1. Purpose

Test-Case Specification document defines a test of ‘**Event Scheduling System**

6.3.2. Outline

A test case specification shall have the following structure:

a. Test case specification identifier
b. Test items
c. Input specifications
d. Output specifications
e. Environmental needs
f. Inter case dependencies

6.3.2.1. Test Case for Specification for Admin

Table 6.1

Test Engineer:	Talha Tarar
Test Case ID:	TC1
Test items:	Admin login to the Web App
Input Specifications:	Block/ Unblock User/ Service Provider
Output Specifications:	Successfully
Test Environment Needs	Home, Laptop or Computer and Any OS
Inter-case dependencies:	None

6.3.2.1. Test Case for Specification of Admin to Login

Table 6.2

Test Engineer:	Ahmed Jamal
Test Case ID:	TC2
Test items:	Admin login to the Web App
Input Specifications:	Admin can check all details
Output Specifications:	Successful
Test Environment Needs	Home, Android mobile
Inter-case dependencies:	None

6.3.2.1. Test Case Specification for Login to Android App

Table 6.3

Test Engineer:	Hamza Javed
Test Case ID:	TC3
Test items:	Service Provider/ Client login to the Android App

Input Specifications:	Name and Password stored in database
Output Specifications:	Authorized User/ Event Organizer will be logged in successfully
Test Environment Needs	Home, Android mobile
Inter-case dependencies:	TC1

6.3.2.1. Test Case for Specification to Place Order

Table 6.4

Test Engineer:	Hamza Javed
Test Case ID:	TC4
Test items:	Place Order
Input Specifications:	User click on Package and then place order
Output Specifications:	Successful
Test Environment Needs	Home, Android mobile
Inter-case dependencies:	TC3

6.3.2.1. Test Case for Specification for Event Organizer to Accept Order

Table 6.5

Test Engineer:	Talha Tarar
Test Case ID:	TC5

Test items:	Service Provider/ Event Organizer accept order
Input Specifications:	Service Provider/ Event Organizer click on Accept Order.
Output Specifications:	Successful
Test Environment Needs	Home, Android mobile
Inter-case dependencies:	TC3

6.3.2.1. Test Case for Specification of Client to create Custom Package

Table 6.6

	Ahmed Jamal
Test Case ID:	TC6
Test items:	Create Custom Package
Input Specifications:	User select different items and services to create custom package.
Output Specifications:	Successful
Test Environment Needs	Home, Android mobile
Inter-case dependencies:	TC5

6.3.2.1. Test Case for Specification for Event organizer to cancel order.

Table 6.7

Test Engineer:	Hamza Javed
Test Case ID:	TC7

Test items:	Event Organizer/ Service Provider Cancel Appointment
Input Specifications:	Service Provider/ Event Organizer click on Cancel Appointment
Output Specifications:	Successful
Test Environment Needs	Home, Android mobile
Inter-case dependencies:	TC3, TC4

6.3.2.1. Test Case for Specification for Service Provider to create Portfolio

Table 6.8

Test Engineer:	Talha Tarar	
	Test Case ID:	TC8

Test items:	Create Portfolio
Input Specifications:	Event Organizer click on create Portfolio
Output Specifications:	Successful
Test Environment Needs	Home, Android mobile
Inter-case dependencies:	TC3

6.3.2.1. Test Case for Specification for Event Organizer to create a package

Table 6.9

Test Engineer:	Ahmed Jamal

Test Case ID:	TC9
Test items:	Delete Package
Input Specifications:	Event Organizer select and delete package.
Output Specifications:	Successful
Test Environment Needs	Home, Android mobile
Inter-case dependencies:	TC3

6.3.2.1. Test Case for Specification for Check Calendar

Table 6.10

Hamza Javed, Talha Tarar

Test Case ID:	TC10
Test items:	Check calendar to check orders schedule
Input Specifications:	Event Organizer can check calendar to check orders on specific date
Output Specifications:	Successful
Test Environment Needs	Home, Android mobile
Inter-case dependencies:	TC3

6.3.2.1. Test Case for Specification for Event Organizer to accept order.

Table 6.11

Ahmed Jamal

Test Case ID:	TC11
Test items:	Event Organizer can accept order.
	Event Organizer can Click on Accept Order

Input Specifications:	
Output Specifications:	Successful
Test Environment Needs	Home, Android mobile
Inter-case dependencies:	TC3, TC10

6.3.2.1. Test Case for Specification for Event Organzier to Cancel order

Table 6.12

Test Engineer:	Hamza Javed	
	Test Case ID:	TC12
Test items:	Event Organizer can Cancel Order	
Input Specifications:	Event Organizer can click on Cancel Order	
Output Specifications:	Successful	
Test Environment Needs	Home, Android mobile	
Inter-case dependencies:	TC3, TC10	

6.3.2.1. Test Case for Specification for App Users to Change Profile

Table 6.13

Test Engineer:	Hamza Javed, Talha Tarar, Ahmed Jamal
Test Case ID:	TC13
Test items:	Event Organizer and Client can change their Profile
	Event Organizer and Client can click on Settings to change profile

Input Specifications:	
Output Specifications:	Successful
Test Environment Needs	Home, Android mobile
Inter-case dependencies:	TC3

6.3.2.1. Test Case for Specification to Check History

Table 6.14

Test Engineer:	Hamza Javed, Talha Tarar, Ahmed Jamal
Test Case ID:	TC14
Test items:	Client can check history
Input Specifications:	Client can check history about previous orders and rating.
Output Specifications:	Successful
Test Environment Needs	Home, Android mobile
Inter-case dependencies:	TC3

6.3.2.1. Test Case for Specification to Add Packages

Table 6.15

Test Engineer:	Talha Tarar, Ahmed Jamal
Test Case ID:	TC15
Test items:	Event Organizer can add new packages.
Input Specifications:	Event Organizer can add packages by click on Packages button
Output Specifications:	Successful

Test Environment Needs	Home, Android mobile

Inter-case dependencies:	TC3

6.3.2.1. Test Case for Specification for Admin to view detail about users

Table 6.16

Test Case ID:	TC16	Talha Tarar
Test items:	Admin can view user detail.	
Input Specifications:	Admin can view user detail from web application	
Output Specifications:	Successful	

Test Environment Needs	Home, Android mobile

Inter-case dependencies:	TC2

6.3.2.1. Test Case for Specification to Logout from App

Table 6.17

Test Case ID:	TC17	Talha Tarar, Ahmed Jamal, Hamza Javed
Test items:	Service Provider, Client and Admin can logout from app	
Input Specifications:	Service Provider, Client and Admin can click on Logout	
Output Specifications:	Successful	

Test Environment Needs	Home, Android mobile , web application

Inter-case dependencies:	TC3

Appendixes:

Appendix 1: Final Documentation Format Guidelines

Typographical Format and Binding

Page Format:

Page size:	A4
Top margin:	1.00 inch
Bottom margin:	1.00 inch
Left margin:	1.5 inch
Right margin:	1.00 inch

Page numbering: Bottom right - part of the footnote

Title page not numbered

All other pages before the page of chapter one numbered in lower roman numerals (i, ii, iii, …)

All other pages starting from first page of chapter one to last page of the report numbered in integers (1, 2, 3, …)

Footer: Each page shall have a footnote "University of Gujrat"

Left aligned

In case of long titles shorter versions should be used.

There shall be a line over the footnote.

Header: Each page shall have a header “Project Name”

Left aligned

In case of long titles shorter versions should be used.

There shall be a line under the footnote.

Chapter Startup: Each chapter shall be numbered as Chapter 1, Chapter 2, etc. The name of the chapter shall be written immediately below.

Both shall be centered horizontally as well as vertically.

The actual chapter content shall start from the next page.7

Text: Only one side of the paper shall be used.
The other side shall be blank.

When a report is opened the right side would contain text, figures, or tables and the left side would be blank.

Tables and Figures: Tables and figures shall be placed on one side only

Separate pages shall be used for figures and tables.

One page may contain more than one figure or table but text will not be combined or interlaced with figure or table.

Each table / figure shall be numbered.

For example "Table 1.2: Population distribution in Asia" or "Figure 3.2: Temperature distribution"

The table number or figure number shall be placed as normal text centered at the bottom of the table or figure or sideways with table / figure title coming on the opening side of the paper and note on the binding side.

Paragraph:

Single-spaced.

Line entered paragraph.

DONOT put indents at the beginning of the paragraph.

Left aligned or justified.

Text Format

Normal and plane text:

Font Type:	Times New Roman
Font Size:	12

Headings:

Chapter Heading:	Times New Roman Bold Size 16 Title Case normal
Heading 1:	Times New Roman Bold Size 14 Title Case normal
Heading 2:	Times New Roman Bold Size 12 Title Case normal
Heading 3:	Times New Roman Bold Size 12 Title Case italic

Sections and Subsections

In case of sections and subsections follow this format:

1 Section

1.1 Sub Section

1.1.1 Nested Sub Section

a

b

i

ii

The subsequent reference to a any section shall be made using the section and its number. For example **section 2.1.3** means chapter 2 section 1 subsection 3.

Mathematical Equations

The following numbering scheme should be used to number the equations:

$$f(x) = x+3 \qquad (XX:YY)$$

Where XX is the chapter number and YY is the sequence number of that equation in that chapter.

If an equation is previously quoted in an earlier chapter, say as equation 4:5 and need to be re-quoted in chapter 5, its number will remain as equation 4:5.

References

References are to be placed in square brackets and interlaced in the text. For example "A comprehensive detail of how to prevent accidents and losses caused by technology can be found in the literature [1]. A project report / thesis cannot be accepted without proper references. The references shall be quoted in the following format:

The articles from journals, books, and magazines are written as:

[1] Abe, M., S. Nakamura, K. Shikano, and H. Kuwabara. Voice conversion through vector quantization. *Journal of the Acoustical Society of Japan*, April 1990, E-11 pp 71-76.

[2] Hermansky, H. Perceptual linear predictive (PLP) analysis for speech. *Journal of the Acoustical Society of America*, January 1990, pp 1738-1752. The books are written as:

[1] Nancy G. Leveson, Safeware System Safety and Computers, A guide to preventing accidents and losses caused by technology, Addison-Wesley Publishing Company, Inc. America, 1995.

[2] Richard R. Brooks, S. S. Iyengar, *Multi-Sensor Fusion Fundamentals and Applications with Software*, The Prentice-Hall Inc. London, 1998.

Binding

All reports shall be bounded with an appropriate print on the backbone.
Two copies should be submitted.

Color of the binding:

- BSc project / thesis reports: black
- MSc project / thesis reports: blue

Contents of the CD Attached

All reports / theses must accompany a CD whose contents will have the following:

Top-level directories:

Doc All documents related to the project

Instructions how to access the CD to the point to running the project

All reports already submitted

The final project report in thesis form

Installation instructions

Trouble shooting instructions in case of problems

User manual

Research material including URLs

Papers consulted / referred to

Slides of the presentations

Source All source files that will be needed to compile the project.

Further subdirectories can be used.

This must include sample data files as well.

Project The running project including sample data files as well as sample output.

This should be in a form that if copied to a machine runs without errors.

This may an exe file of an entire project, an installer depending on the project or simply a running project.

You can have sub directories with appropriate names.

Length

The length of your dissertation depends on the type of project you have selected. An excellent dissertation will often be brief but effective (its author will have said a lot in a small amount of space). Voluminous data can be submitted electronically on CD.