I-TOWNSHIP

An effective society management web web application

A PROJECT REPORT

Submitted by

Datt Patel [120770107056]

Ronak Janani [120770107062]

Mayur Kanojiya [120770107086]

In fulfilment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER ENGINEERING

Silver Oak College of Engineering & Technology, Ahmedabad

Gujarat Technological University, Ahmedabad

2015-16

CANDIDATE'S DECLARATION

We hereby declare that project report titled **"I-Township"** submitted towards the completion of Project in 8th semester of Bachelor of Computer Engineering in Silver Oak College of Engineering & Technology, Ahmedabad is an authentic record of our work carried out.

First Candidate's Name : **Datt Patel**

Branch : **CE**

Enrollment No : **120770107056**

Candidate's Signature :

Second Candidate's Name :**Ronak Janani**

Branch : **CE**

Enrollment No : **120770107062**

Candidate's Signature :

Third Candidate's Name : **Mayur Kanojiya**

Branch : **CE**

Enrollment No : **120770107086**

Candidate's Signature :

Submitted To: **Silver Oak College of Engineering & Technology,**

Ahmadabad,

Affiliated to,

Gujarat Technological University.

ACKNOWLEDGEMENT

We would like to extend our heartily thanks with a deep sense of gratitude and respect to all those who provides us immense help and guidance during our project.

We would like to thank **Prof.Satvik Khara**(H.O.D - CE)for providing a vision about the system. We have been greatly benefited from their regular critical reviews and inspiration throughout my work.

We would like to express our sincere thanks to our internal guide **Miss. Snehi Patel** respectively and who gave us an opportunity to undertake such a great challenging and innovative work. We are grateful to them for their guidance, encouragement, understanding and insightful support in the development process.

Last but not the least we would like to mention here that we are greatly indebted to each and everybody who has been associated with our project at any stage but whose name does not find a place in this acknowledgement.

Yours Sincerely,

Datt Patel
(120770107056)

Ronak Janani
(120770107062)

Mayur Kanojiya
(120770107086)

ABSTRACT

I-Township is a java based web web application which is built using Spring MVC Framework. It provides platform for residents of society to manage their time and work by using features of this web web application. There is also an android web application which is built using Android Studio which gives necessary information about society.

The main aim of the webweb application is to help members of society to effectively manage their society related work and establish communication with each other remotely. Purpose of this web web application is to provide a solution for management, communication, transparent finance policy and efficient maintenance of running day to day operation of society.

LIST OF FIGURES

LIST OF TABLES

TABLE OF CONTENTS

CHAPTER

1

INTRODUCTION

1.1 Project Summary

1.2 Purpose

1.3 Scope

1.4 Technology & Literature Review

1.4.1 Technology & Tools

1.4.2 Literature Review

CHAPTER 1 :INTRODUCTION

1.1PROJECT SUMMARY

- In this project we are going to develop the java based software which involves management of a society. We will develop the web portal for this Society management via "Spring MVC" and an android web application to receive push notifications sent by chairman.
- There are mainly four modules Management, Forum, Notifications and Payment Gateway which will help members of society to manage society related work, to resolve issues or give their view on some ongoing topics, to be in touch with chairman directly, to make fast transaction of bill and expenses by using above modules respectively.
- In this portal, the chairman suggests the management for completing such tasks. This task involves management of society from different areas of interest. And management orders the members of society for doing that task. Task process will be tracked by task manager which will ensure that task must complete on time.
- It includes various functionalities such as perfect UI interactions, complain box, notification box. At last user can suggest or give complains or give thanks to management according to their requirements.
- There is such functionality like notifications, security, VR map integration etc. To get a notification about important work of society,We have developed and android web application where all these notifications willbe received.

1.2PURPOSE

- The whole purpose of the web application is to help members of society to utilize their time and work by using features of this project.
- The problems of society can be reduced by using this system. It will be beneficial to citizens because they can communicate to the chairman through this portal.

- The task of chairman becomes easy by using this portal because they can manage in proper manner.

1.3SCOPE

- Based on its advanced computing capabilities and transparent financial policy, It will used by most society and township.
- With a growing number of users and a wide variety of web applications emerging, the Smartphones is fundamentally altering our current use and understanding of the society management.
- The vision of nomadic users having seamless, worldwide access to a range of i-township services is expected to become a reality within only a few years from now.
- Hence, the concept of i-township has recently emerged wherein users manage their society related work remotely.
- I-township is also provide VR-Dashboard which will give 360 virtual view of society using Oculus, Cardboard etc. Hence, It this feature will be more helpful and much used by members of society in near future.

1.4TECHNOLOGY & LITERATURE REVIEW

1.4.1TECHNOLOGY AND TOOLS

- **Technology :** Spring MVCFramework
- **Front End :** HTML, CSS, JavaScript
- **Back End:** Java, MySQL, JSON, Google Cloud Messaging API
- **Development Tool :** Eclipse
- **Development Language :** Java
- **Simulation Tool :** Google Chrome
- **Simulation OS :** Windows/Linux/Mac
- **Versions support :** Internet Explorer Latest

1.4.2LITERATURE REVIEW

➢A brief introduction to Spring MVC Framework :

The Spring Web model-view-controller (MVC) framework is designed around a DispatcherServlet that dispatches requests to handlers, with configurable handler mweb appings, view resolution, locale and theme resolution as well as support for uploading files. The default handler is based on the @Controller and @RequestMweb appingannotations, offering a wide range of flexible handling methods. With the introduction of Spring 3.0, the @Controller mechanism also allows you to create RESTful Web sites and web applications, through the @PathVariable annotation and other features.

In Spring Web MVC you can use any object as a command or form-backing object; you do not need to implement a framework-specific interface or base class. Spring's data binding is highly flexible: for example, it treats type mismatches as validation errors that can be evaluated by the web application, not as system errors. Thus you need not duplicate your business objects' properties as simple, untyped strings in your form objects simply to handle invalid submissions, or to convert the Strings properly. Instead, it is often preferable to bind directly to your business objects.

Spring's view resolution is extremely flexible. A Controller is typically responsible for preparing a model Map with data and selecting a view name but it can also write directly to the response stream and complete the request. View name resolution is highly configurable through file extension or Accept header content type negotiation, through bean names, a properties file, or even a customViewResolver implementation. The model (the M in MVC) is a Map interface, which allows for the complete abstraction of the view technology. You can integrate directly with template based rendering technologies such as JSP, Velocity and Freemarker, or directly generate XML, JSON, Atom, and many other types of content. The model Map is simply transformed into an web appropriate format, such as JSP request attributes, a Velocity template model.

➢**Features of Spring Web MVC:**

Spring's web module includes many unique web support features:

- Clear separation of roles. Each role — controller, validator, command object, form object, model object, Dispatcher Servlet, handler mweb apping, view resolver, and so on — can be fulfilled by a specialized object.
- Powerful and straightforward configuration of both framework and web application classes as JavaBeans. This configuration capability includes easy referencing across contexts, such as from web controllers to business objects and validators.
- Adaptability, non-intrusiveness, and flexibility. Define any controller method signature you need, possibly using one of the parameter annotations (such as @RequestParam, @RequestHeader, @PathVariable, and more) for a given scenario.
- Reusable business code, no need for duplication. Use existing business objects as command or form objects instead of mirroring them to extend a particular framework base class.
- Customizable binding and validation. Type mismatches as web application-level validation errors that keep the offending value, localized date and number binding, and so on instead of String-only form objects with manual parsing and conversion to business objects.
- Customizable handler mweb apping and view resolution. Handler mweb apping and view resolution strategies range from simple URL-based configuration, to sophisticated, purpose-built resolution strategies. Spring is more flexible than web MVC frameworks that mandate a particular technique.
- Flexible model transfer. Model transfer with a name/value Map supports easy integration with any view technology.
- Customizable locale and theme resolution, support for JSPs with or without Spring tag library, support for JSTL, support for Velocity without the need for extra bridges, and so on.
- A simple yet powerful JSP tag library known as the Spring tag library that provides support for features such as data binding and themes. The custom tags allow for maximum flexibility in terms of markup code.

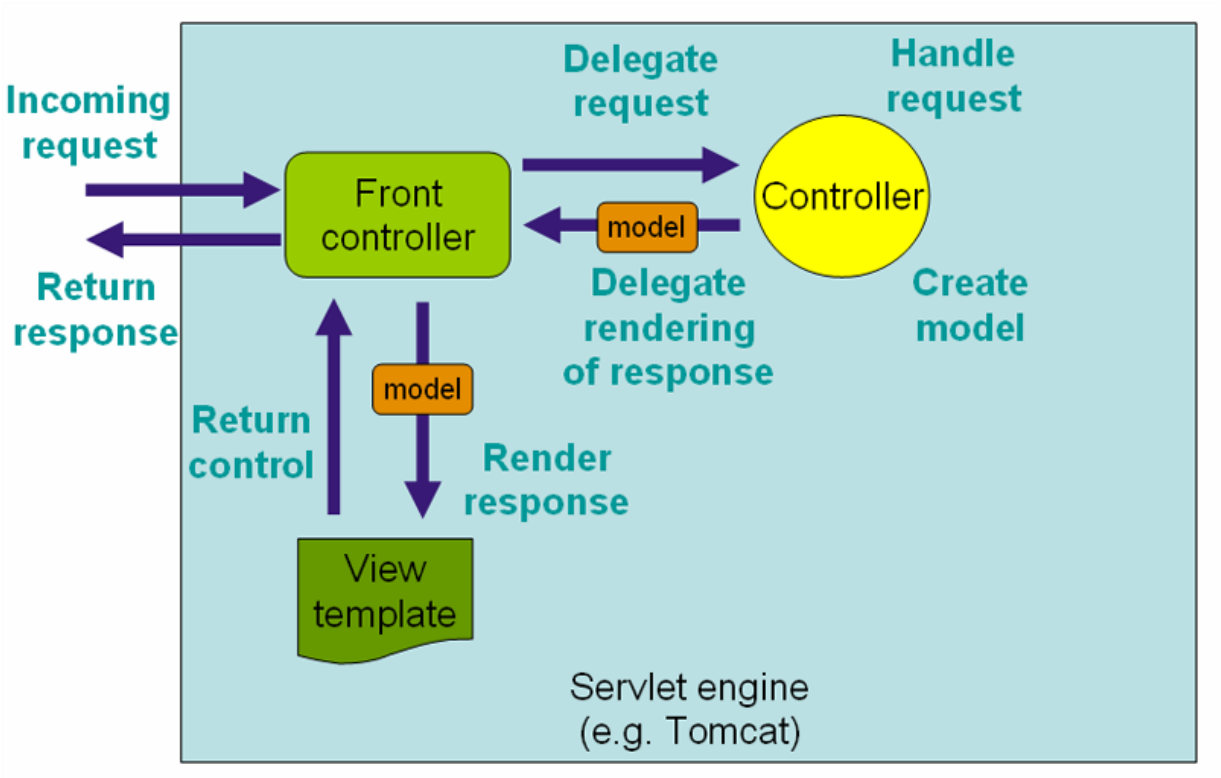


Figure 1.2 The request processing workflow in Spring Web MVC (high level)

➢**Components of Android Web application Framework :**

The DispatcherServlet is an actual Servlet (it inherits from the HttpServlet base class), and as such is declared in the web.xml of your web web application. You need to map requests that you want the DispatcherServlet to handle, by using a URL mweb apping in the same web.xml file. This is standard Java EE Servlet configuration; the following example shows such a DispatcherServlet declaration and mweb apping:

<web-web app>

<servlet>

<servlet-name>example</servlet-name>

<servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>

<load-on-startup>1</load-on-startup>

</servlet>

<servlet-mweb apping>
<servlet-name>example</servlet-name>
<url-pattern>/example/*</url-pattern>
</servlet-mweb apping>

</web-web app>

In the preceding example, all requests starting with /example will be handled by the DispatcherServlet instance named example. In a Servlet 3.0+ environment, you also have the option of configuring the Servlet container programmatically. Below is the code based equivalent of the above web.xml example:

publicclassMyWebWeb applicationInitializer**implements**WebWeb applicationInitializer {

@Override
publicvoidonStartup(ServletContext container) {
ServletRegistration.Dynamic registration = container.addServlet("dispatcher", **new**DispatcherServlet());
registration.setLoadOnStartup(1);
registration.addMweb apping("/example/*");
}
}

WebWeb applicationInitializer is an interface provided by Spring MVC that ensures your code-based configuration is detected and automatically used to initialize any Servlet 3 container. An abstract base class implementation of this interace named AbstractDispatcherServletInitializer makes it even easier to register theDispatcherServlet by simply specifying its servlet mweb apping.The above is only the first step in setting up Spring Web MVC. You now need to configure the various beans used by the Spring Web MVC framework, Web applicationContext instances in Spring can be scoped. In the Web MVC framework, each DispatcherServlet has its own WebWeb applicationContext, which inherits all the beans already defined in the root WebWeb applicationContext. These inherited beans can be overridden in the

servlet-specific scope, and you can define new scope-specific beans local to a given Servlet instance.

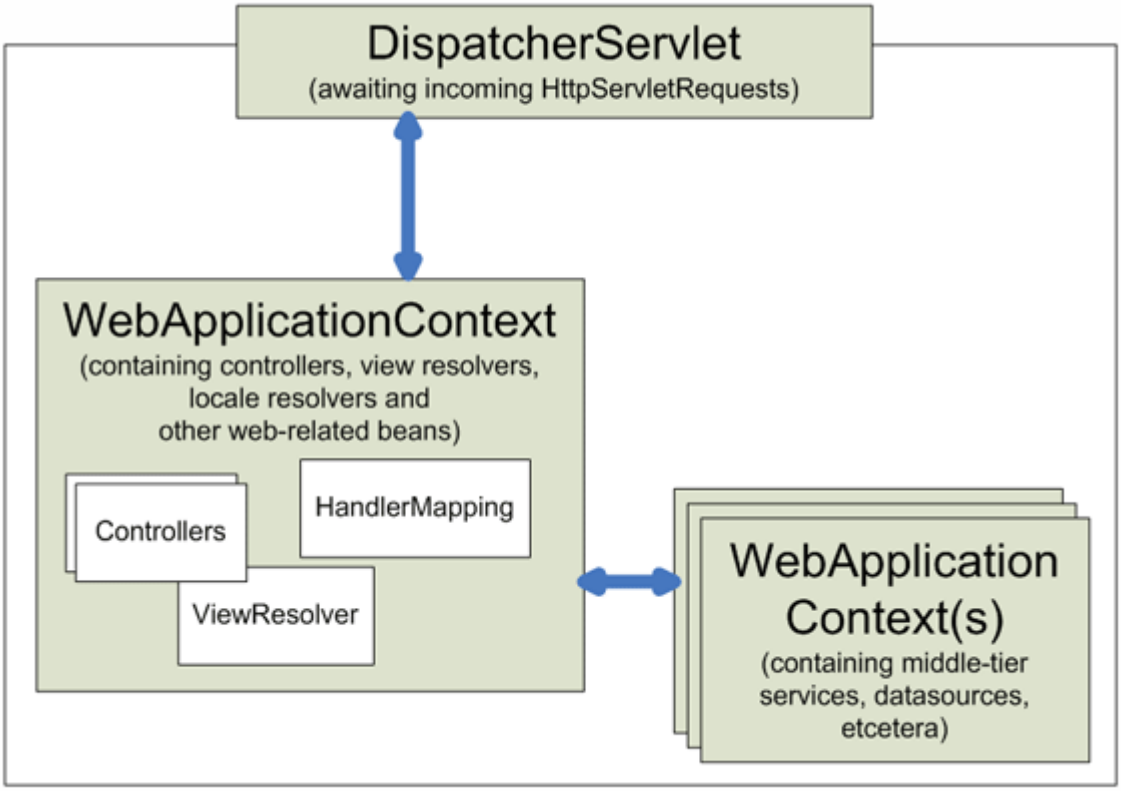


Figure 1.3Context hierarchy in Spring Web MVC

Upon initialization of a DispatcherServlet, Spring MVC looks for a file named *[servlet-name]-servlet.xml* in the WEB-INF directory of your web web application and creates the beans defined there, overriding the definitions of any beans defined with the same name in the global scope.

Consider the following DispatcherServlet Servlet configuration (in the web.xml file):

<web-web app>

<servlet>

<servlet-name>**golfing**</servlet-name>

<servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>

<load-on-startup>1</load-on-startup>

</servlet>

<servlet-mweb apping>

<servlet-name>**golfing**</servlet-name>

<url-pattern>/golfing/*</url-pattern>

</servlet-mweb apping>

</web-web app>

With the above Servlet configuration in place, you will need to have a file called /WEB-INF/**golfing**-servlet.xml in your web application; this file will contain all of your Spring Web MVC-specific components (beans). You can change the exact location of this configuration file through a Servlet initialization parameter

- **Google Cloud Messaging API :**

Google Cloud Messaging (GCM) is a free service that enables developers to send messages between servers and client web apps. This includes downstream messages from servers to client web apps, and upstream messages from client web apps to servers.

For example, a lightweight downstream message could inform a client web app that there is new data to be fetched from the server, as in the case of a "new email" notification. For use cases such as instant messaging, a GCM message can transfer up to 4kb of payload to the client web app. The GCM service handles all aspects of queueing of messages and delivery to and from the target client web app.

A GCM implementation includes a Google connection server, an web app server in your environment that interacts with the connection server via HTTP or XMPP protocol, and a client web app.

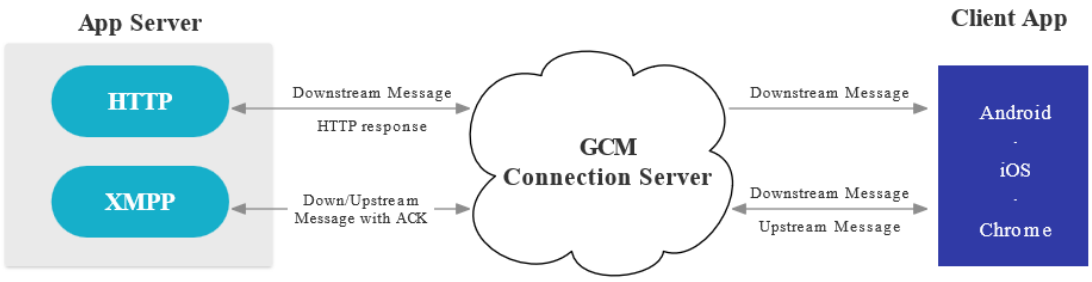


Figure 1.4 GCM Architecture.

Here's how these components interact:

- Google **GCM Connection Servers** accept downstream messages from your web app server and send them to a client web app. The XMPP connection server can also accept messages sent upstream from the client web app and forward them to your web app server. For more information, see About GCM Connection Server.
- On your **Web app Server**, you implement the HTTP and/or XMPP protocol to communicate with the GCM connection server(s). Web app servers send downstream messages to a GCM connection server; the connection server enqueues and stores the message, and then sends it to the client web app. If you implement XMPP, your web app server can receive messages sent from the client web app.
- The **Client Web app** is a GCM-enabled client web app. To receive and send GCM messages, this web app must register with GCM and get a unique identifier called a registration token. For more information on how to implement the client web app, see the documentation for your platform.

CHAPTER

2

SOFTWARE PROJECT MANAGEMENT &PLAN

2.1 Project Planning & Scheduling

2.1.1 Project Development Web approach

2.1.2 Project Plan

2.1.3 Scheduling

2.2 Risk Management

2.2.1 Risk Identification

2.2.2 Project Analysis

2.2.3 Risk Planning

2.3 Estimation

2.3.1 Effort & Cost Estimation

2.3.2 Cost Analysis

CHAPTER:2

SOFTWARE PROJECT MANAGEMENT AND PLAN

2.1 PROJECT PLANNING & SCHEDULING

2.1.1 PROJECT DEVELOPMENT WEB APPROACH

To developed this project we will follow the prototype model because we need to make a _quick design‘ of the web application first to understand the overall flow and behaviour of the web application and accordingly to refine our requirements. Also, a developed prototype can help us to critically examine the technical issues associated with the product development.

Another reason for choosing prototype as a process model is that it is not possible that we can get the perfect product in the first attempt. The experience gained in developing the prototype can be used further to develop the final product.

➢**Prototype model:**

A prototype is a toy implementation of the system. A prototype usually exhibits limited functional capabilities, low reliability, and inefficient performance compared to the actual software. A prototype is usually built using several shortcuts. The shortcuts might involve using inefficient, inaccurate, or dummy functions. The shortcut implementation of a function, for example, may produce the desired results by using a table look-up instead of performing the actual computations.

A prototype is a toy implementation of the system. A prototype usually exhibits limited functional capabilities, low reliability, and inefficient performance compared to the actual software. A prototype is usually built using several shortcuts. The shortcuts might involve using inefficient, inaccurate, or dummy functions. The shortcut implementation of a function, for example, may produce the desired results

by using a table look-up instead of performing the actual computations.

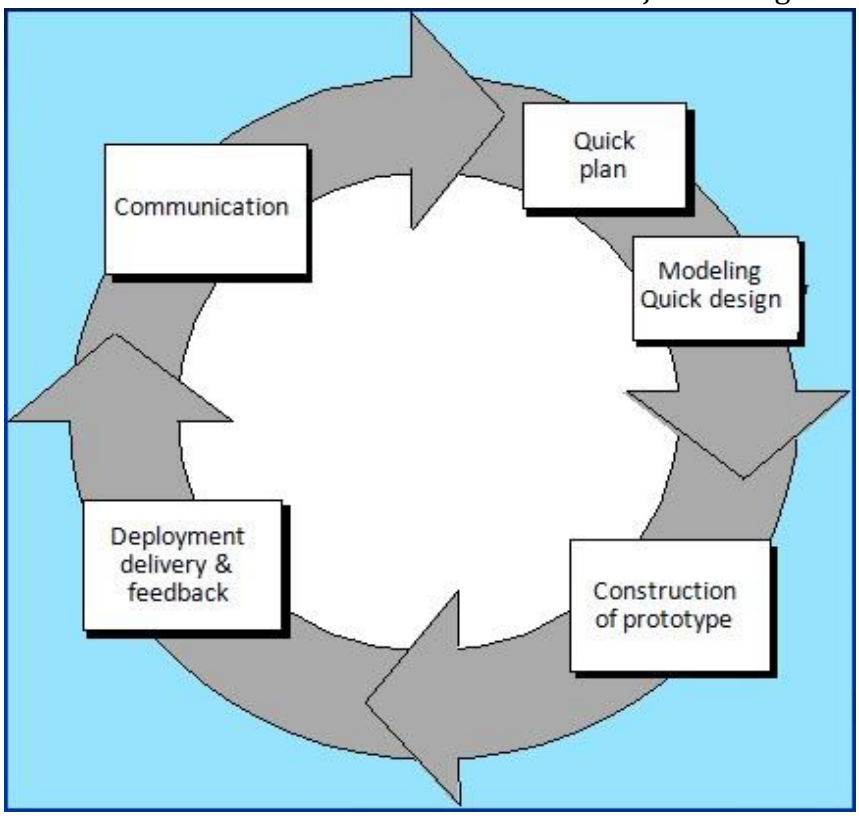


Figure 2.1 Prototype Model

This is something similar to what the architectural designers of a building do; they show a prototype of the building to their customer. The customer can evaluate whether he likes it or not and the changes that he would need in the actual product. A similar thing hweb appens in the case of a software product and its prototyping model.

The prototyping paradigm begins with requirements gathering. Developer and customer meet and define the overall objectives for the software, identify whatever requirements are known, and outline areas where further definition is mandatory. A "quick design" then occurs. The quick design focuses on a representation of those aspects of the software that will be visible to the customer/user. The quick design leads to the construction of a prototype. The prototype is evaluated by the customer/user and used to refine requirements for the software to be developed. Iteration occurs as the prototype is tuned to satisfy the needs of the customer, while at the same time enabling the developer to better understand what needs to be done.

2.1.2PROJECT PLAN

- Project planning is the discipline of planning, organizing and managing resources to bring about the successful completion of specific project goals and objectives.
- The primary challenge of project planning is to achieve all of the project goals and objectives while honouring the project constraints. Typical constraints are scope, time and budget. The secondary and more ambitious challenge is to optimize the allocation and integration of inputs necessary to meet pre-defined objectives.

➢ **Milestones & Deliverables:**

Milestone is an end-point of the software process activity. At each milestone there should be formal output, such as report, that can be represented to the management.

Milestones	Deliverables
Studying various modules of I-township web web application & its Problems. Started collecting the Requirements.	SPMP
Study of existing System and its flow; Understand the situation and its solution.	System Requirement Study
Gathering the requirements of the project using different fact-finding techniques.	
Understanding the project in depth and doing the requirement analysis by studying process of Tourist Guide modules and database creation.	System Analysis
On the basis of detailed study and specification of the requirements Data dictionary was initiated.	
Creating UML design based on analysis.	
Started Web application Flow Diagram.	Detailed Design Document
The functional Design of the web application.	

Started making a quick design or ‗prototype' of modules to understand and refine requirements accordingly.	First Prototype
Completed the prototype design.	
Started Module coding. Short-term Coding objectives like Form design (Input, Output, Interface, and Implementation)	Progress Report
Started Module Integration.	
Implementation almost finish, so to start unit testing for the same.	
Involved in Module Testing by input characters, validations, flow of contents.	Final Prototype
Completed Web application Testing and deployed on a Web Server.	Deployed Web application

Table 2-1 Milestones & Deliverables

- **Roles and Responsibilities:**

Role	Responsibility	Team/Member
Project Guide	Defining scope	Miss. Snehi Patel
	Providing required resources	
	Project planning, tracking and monitoring.	
	Analysis and Effort Estimation.	
Team	Designing & Documentation	Datt Patel Ronak Janani
Leader	Execution and implementation of project as per defined schedule.	Mayur Kanojiya

Team Member	Software development as per the design and Documentation and defined scope	Datt Patel Ronak Janani Mayur Kanojiya
QA	Testing and Quality Check as per Defined Requirement.	Datt Patel Ronak Janani Mayur Kanojiya

Table 2-2 Roles & Responsibilities

2.1.3 SCHEDULING

➢Project-task scheduling is an important project planning activity. It involves deciding which tasks would be taken up when. In project management, a schedule consists of a list of a project's terminal elements with intended start and finish dates. Those items are often estimated in terms of resource requirements, budget and duration, linked by dependencies and scheduled events. In order to schedule the project activities, a software project manager needs to do the following:

- Identify all the tasks needed to complete the project.
- Break down large tasks into small activities.
- Establish estimates for the time durations necessary to complete the activities.
- Allocate resources to activities

Task Name	Start Date	Finish Date
Feasibility Study	01/07/15	28/07/15
Requirement Analysis	29/07/15	05/09/15
System Design	08/09/15	17/10/15
Prototyping	11/11/15	12/05/15
Implementation	09/01/16	25/02/16
Testing	26/02/16	09/04/16
Deployment	10/04/16	15/04/16

Table 2-3 Project Schedule

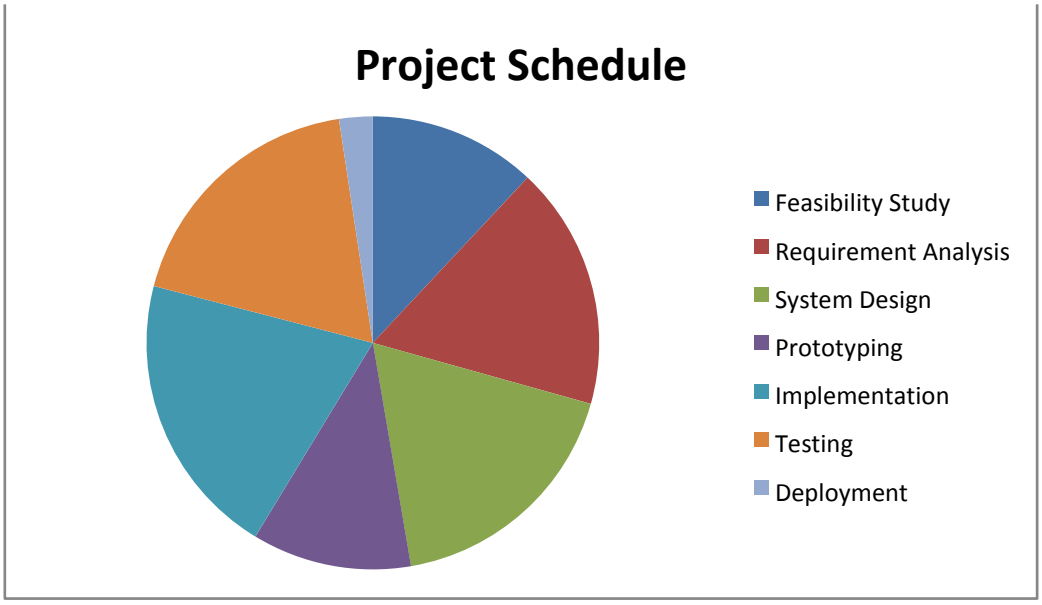


Figure 2.2 Pie-chart Schedule Representation

➢**Gantt Charts :**

Gantt Charts are useful tools for analysing and planning more complex projects. When a project is under way, Gantt Charts help you to monitor whether the project is on schedule.

They are used to:

- Help you to plan out the tasks that need to be completed.
- Give you a basis for scheduling when these tasks will be carried out.
- Allow you to plan the allocation of resources needed to complete the project, and help you to work out the critical path for a project where you must complete it by a particular date.

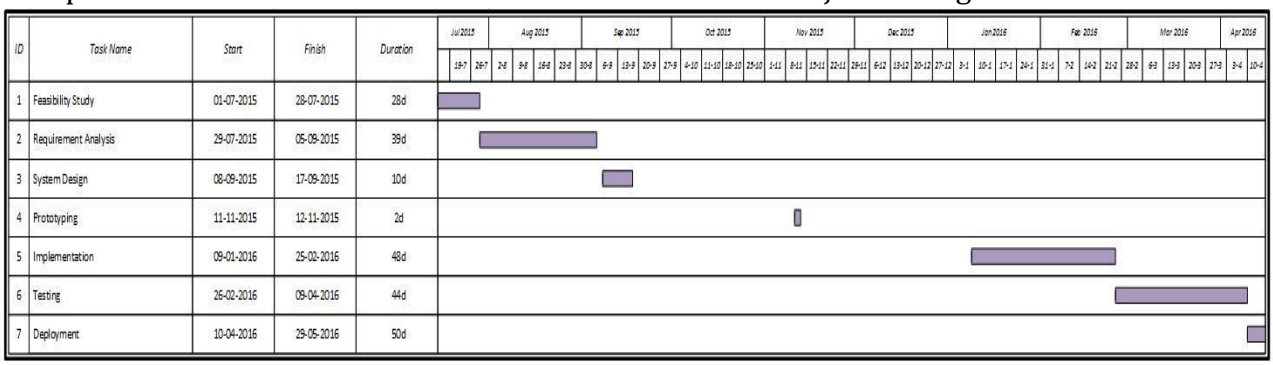


Figure 2.3 Gantt Chart

2.2RISK MANAGEMENT

Risk management can consider as dealing with the possibility and actual occurrence of those events that are not regular or commonly expected that is they are probabilistic. So risk management begins where a normal project management ends.

Risks can be categorized as follows:

Components / Category	Performance	Support	Cost	Schedule
Catastrophic	Failure to meet the requirement	Non responsive or unsupportable software.	Financial shortage and lack of	Failure results in schedule delays. Unachievable line
	would result in mission failure.		resources.	of code.
Critical	Database server crashes.	A backup system with the server installed to be ready for operation.	Cheaper system that does not meet sufficient requirement.	System maintenance after every fixed time period.

Marginal	Overload on server system due to large increase in no. of parallel connections.	Either the bit-rate of the transmission of the video has to be reduced or to decrease the no. of parallel connections.	Sufficient financial support	Reduce the no. of multicasting and also provide some constraints

Table 2-4 Types of Risk

2.2.1RISK IDENTIFICATION

Risk identifications a systematic attempt to specify threats to the project plan (estimates, schedule, resource loading, etc.). By identifying known and predictable risks, the project manager takes a first step toward avoiding them when possible and controlling them when necessary.

The objective of risk assessment is to rank the risks in terms of their damage causing potential. In order to be able to systematically identify the important risks which might affect a software project, it is necessary to categorize risks into different classes.

Risk Type	Description
Technology	The underlying technology on which the system is built is superseded by new technology.
Performance	The server may not handle large no. of parallel requests made by users.
Viruses	Viruses may corrupt software files.
System Failure	In case of system crash, development software and files may get lost.

Hacking	The hackers may spam or flood server with large no. of requests or data.

Table 2-5 Risk Identification

2.2.2 RISK ANALYSIS

While a test plan is created the following risk are identified, these risks are involved in testing the product are to be taken in to consideration along with possibilities of their occurrence and the damage they may cause along with the solution.

Risk	Description	Probability	Impact
Technological Changes	The underlying technology on which the system is built is superseded by new technology.	Low	Medium
Database server crash	Big amount of data can cause database server crash.	Medium	High
Viruses	Viruses may corrupt the software file.	Low	Medium
System Failure	In case of system crash, development software and files may get lost.	Low	High
Flooding/Spamming	The hackers may spam or flood server with requests or data.	Medium	High

Table 2-6 Risk Analysis

2.2.3RISK PLANNING

Risk planning the identification and scheduling of actions needed to reduce the level of risk within a project.

Risk	Remedies/Plan
Database server crash	Store data on distributed servers or on cloud as per requirements.
Viruses	Install good antivirus software on the system.
System Failure	Backup should be kept on another system or on cloud storage services like Google drive, Dropbox etc.
Flooding/Spamming	Also, encrypt personal data of users with the use of hash based encryption algorithms.

Table 2-7 Risk Planning

2.3ESTIMATION

EFFORT AND COST ESTIMATION:

Effort Estimation is the process of prediction the most realistic use of effort required to develop or maintain software based on incomplete input.

(A) Function Point:

Function point metric was proposed by Albrecht. This metric overcomes many of the shortcomings of the LOC metric. One of the important advantages of using the function point metric is that it can be used to easily estimate the size of a software product directly from the problem specification. This is in contrast to the LOC metric, where the size can be accurately determined only after the product has fully been developed.

The size of a product in function points (FP) can be expressed as the weighted sum of these five problem characteristics. The weights associated with the five characteristics were proposed empirically and validated by the observations over many projects. Function point is computed in two steps. The first step is to compute the unadjusted function point (UFP).

Once the unadjusted function point (UFP) is computed, the technical complexity factor (TCF) is computed next. TCF refines the UFP measure by considering fourteen other factors such as high transaction rates, throughput, and response time requirements, etc. Each of these 14 factors is assigned from 0 (not present or no influence) to 6 (strong influence). The resulting numbers are summed, yielding the total degree of influence (DI). Now, TCF is computed as (0.65+0.01*DI). As DI can vary from 0 to 70, TCF can vary from 0.65 to 1.35.

Finally, FP = UFP * TCF.

➢**Function Point equations :**

Unadjusted Function Point (UFP) = (Number of inputs)*4 + (Number of outputs)*5 + (Number of inquiries)*4 + (Number of files)*10 + (Number of interfaces)*10.

Technical Complexity Factor (TCF) = 0.65 + 0.01*Degree of Influence

Function Point (FP) = Unadjusted Function Point * Technical Complexity Factor

No. of inputs: 180

No. of outputs: 120

No. of Inquiries: 80

No. of files: 480

No. of interfaces: 220

UFP = 180*4 + 120*5 + 80*4 + 480*10 + 220*10

= 8,640.

TCF = 0.65 + 0.01* DI

= 0.65 + 0.01*20

= 0.85

FP = UFP * TCF =

8640 * 0.85

∴ FP = 7,344

(B)COCOMO Model:

The Constructive Cost Model (COCOMO) is an algorithmic software cost estimation model developed by Barry W. Boehm. The model uses a basic regression formula with parameters that are derived from historical project data and current as well as future project characteristics.

According to Boehm, software cost estimation should be done through three stages: Basic COCOMO, Intermediate COCOMO, and Complete COCOMO. The model also assumes following three types of projects.

➢**Types of Projects:**

- **Organic:** A development project can be considered of organic type, if the project deals with developing a well understood web application program, the size of the development team is reasonably small, and the team members are experienced in developing similar types of projects.

- **Semi-detached:** A development project can be considered of semidetached type, if the development consists of a mixture of experienced and inexperienced staff. Team members may have limited experience on related systems but may be unfamiliar with some aspects of the system being developed.
- **Embedded:** A development project is considered to be of embedded type, if the software being developed is strongly coupled to complex hardware, or if the stringent regulations on the operational procedures exist.

The four basic steps in software project estimation using basic COCOMO model are:

1. Estimate the size of development product. The units of measure are lines of code (LOC) and function point (FP).
2. Estimate the effort in person-months or person-hours.
3. Estimate the schedule in calendar months.
4. Estimate the project cost.

Project Modes	Constants			
	a_b	b_b	c_b	d_b
Organic project mode	2.4	1.05	2.5	0.38
Semi-detached project mode	3.0	1.12	2.5	0.35
Embedded project mode	3.6	1.20	2.5	0.32

Table 2-8 Constants for Basic COCOMO

Mode Used for our project: Semi-detached

We prefer semi-detached type project because we have limited knowledge and experience on Android development and related systems.

Hence, the values are as follows:

$a_b = 3.0$, $b_b = 1.12$, $c_b = 2.5$, $d_b = 0.35$.

- **COCOMO equations:**

  Effort Estimation (E): $E = a_b * (KLOC)^{b_b}$

  Duration Estimation (D): $D = c_b * (E)^{d_b}$

  Person Estimation (P): $P = E / D$

  Cost Estimation (C): $C = D * \text{Labour Charge}$

Modules	Line of Code (LOC)
Login/sign up and profile module	860
Travel recommendations module	1400
Location tracking and maps module	2100
Trip data management module	1700
Blog module	3800
Total Line of Code	12,535

Table 2-9 Calculation for Line of Code

∴ Total KLOC: 12.54

- **Effort Web applied (E):**

  $E = a_b*(KLOC) \wedge b_b$

  E = 3.0*9.86 ^ 1.12 E = 38.928

  ~ 39 person-months

- **Development Time (D):**

  $D = c_b*(E) \wedge d_b$

  D = 2.5 * (33.30) ^ 0.38

  D = 9.5 months

- **People Required (P):**

  P = E / D

  P = 38.928/ 9.006

  P = 4.3 ~ 4 people

- **Project Cost (C):**

  C = Development time (D) * Labour charge

  C = 33.30*12000

∴ Cost = 3, 99,600 **Rs.**

2.3.4 COST ANALYSIS:

- A cost-benefit analysis is necessary to determine economic feasibility. The Primary objective of the cost-benefit analysis is to find out whether it is economically worthwhile to invest in the project. If the return on the investments is good, then the project is considered economically worthwhile.
- Cost-benefit analysis is performed by first listing all the costs associated with the project. Costs consist of direct costs and indirect costs. Benefits can be broadly classified as tangible benefit and intangible benefits. Tangible benefits are directly measurable and intangible are not. Thus, the associated cost can be decreased .

CHAPTER

3

SYSTEM REQUIREMENT STUDY

3.1 User Characteristics

3.2 Hardware and Software Requirements

3.2.1 Hardware Requirements

3.2.2 Software Requirements

3.3 Constraints

3.4 Assumptions & Dependencies

CHAPTER:3 SYSTEM REQUIREMENT STUDY

3.1 USER CHARACTERISTICS:

User characteristic describes the type of users, which deals with the system i.e. a person or any other entity that uses the system. In the investment planning System there are much kind of user dealing and use this useful web application.

This web application is basically used by end users and moderated by admin. The user can perform following operations:

- **End User :**

  - User can discover meeting which were recently held.
  - User can add expenses occurs during completing task of society.
  - User can pay bills and maintenance online through payment gateway.
  - User can receive importance notice through push notifications which are sent by chairman of society.
  - User can discuss, share and write a view on certain ongoing topic by using Forum.
  - User can also give feedback to chairman concerning improvement of society.
  - User can flag content if it is inweb appropriate.

- **Admin :**

  - Admin has authority to insert, update and delete data.
  - Admin can add/remove user accounts in the web application.
  - Admin can grant/revoke privileges on users.
  - Admin can check flagged posts or inweb appropriate content and delete them.

3.2 HARDWARE AND SOFTWARE REQUIREMENTS

3.2.1 HARDWARE REQUIREMENTS

Requirement Specification	For Simulation Platform	For Server	For Android Device
Processor	Intel Dual Core or higher	Intel Dual Core or higher	1 GHz
Disk space	160 GB	500 GB	50 MB
RAM	1 GB	2 GB	512 MB

Table 3-1 Hardware Requirements

3.2.2SOFTWARE REQUIREMENTS

Requirement Specification	For Simulation Platform	For Server	For Android Device
Operating System	Windows XP or higher	Windows XP or higher	Android 2.3 Gingerbread and higher
Development Kits	JDK, Android SDK	-	-
IDE/ Workbench	Eclipse with ADT pug-in	APACHE	-
Web Server	-	Apache HTTP Server	-
Database Server	-	MySQL	-

Table 3-2 Software Requirements

3.3 CONSTRAINTS:

- **Regularity policies:**

  - Look and feel as well as the user interface of the Web application Development Metrics Management should be easy with user-friendly navigation.
  - Commonly used icons should be used for easy understanding of users.
  - Proper level of abstraction should be maintained for each module in the system.

- **Hardware limitations:**

  - On Android platform the web application itself requires 1.0 GHz processor with minimum 512 MB of RAM and minimum 50 MB of disk space.
  - The devices on which the web application is to be installed must have GPS supported hardware.

- **Interface to other web application:**

  - The web application requires an interface to Google maps for rendering places on the map.
  - So, Google play services must be installed on the Android device to run services which are based on Google maps.

- **Higher order language requirements:**

  - Now at this time there is no need to require the higher level language. Because it already works with the JAVA language environment.
  - JAVA is the most viable language for Web application development measurement metrics system.

- **Performance requirements:**

  - The web application must be efficient & fast. Its response time should be minimal. The output should be responded within a maximum of 10 seconds depending on the internet connection speed and handset performance.

- It should also work in slow 2G (less than 15 kbits/second) internet connection. For this, some kind of catching mechanism should be implemented.

➢ **Safety and security considerations:**

- The web application should have a high level of security and safety consideration above any of the feature so that an unauthorized user can not log into the system.
- The web application must utilize some hash based encryption algorithm to protect the critical databases from hackers.

➢ **Reliability requirements:**

- The web application must implement error checking and crash recovery features.
- The sever must be available all the time and it must be capable for maintaining high load of concurrent database operations.
- The web application should be protected against failure by dynamically resolving the use of system resources at runtime.

3.4 ASSUMPTIONS & DEPENDANCIES:

- It is assumed that the user is well familiar with the Android device.
- User has basic knowledge of navigation and gestures in tough based devices.
- User is aware about rules and regulations.

CHAPTER

4

SYSTEM ANALYSIS

4.1 Study of Current System

4.2 Problem and Weakness of Current System

4.3 Requirement of New System

4.4 Feasibility Study

4.5 Requirements Validation

4.6 Functions of System

4.6.1 Use Case Diagram

4.7 Data Modelling

4.7.1 Class & E-R Diagram

4.7.2 Activity Diagram

4.7.3 Sequence Diagram

4.7.4 Data Dictionary

4.8 Functional and BehaviouralModelling

4.8.1 Data Flow Diagram

4.9 Main Modules of New System

4.10 Selection of Hardware & Software & Justification

CHAPTER:4 SYSTEM ANALYSIS

4.1STUDY OF CURRENT SYSTEM:

- Chairman of Society direct residents to do task which leads society to its peaceful and interactive state.
- Chairman manually manages maintenance department, meetings, events, problems, forum, notifications, tasks etc.
- Chairman is a responsible person who direct other residents with one to one communication.
- So, these systems address an important aspect of personalization and hence reduce the memory burden for the user.

4.2 PROBLEMS IN EXISTING SYSTEMS

- The problems associated with communication problems and advising residentsleads to miss communication, i.e. to People don't know what to or not.
- The existing system fails in building trust on financial policy of chairman due to their no transparent policy on payments.
- Residents cannot give their time for particular event or meeting on specific date.
- Sometimes in high end township living, even neighbours don't know each other name or any contact details.
- The same data gets repeated over and over since the residents find it hard to Keep Track Of the Documents, Information and Transactions.
- Since everything and every detail are written down manually in paper there will be too much paper work and it's frustrating.
- There will be unavailability for future use, since data might get misplaced during manual filing. So data won't be preserved properly for future use.
- When entering data users might have accidentally switched details and data since it is hand written.
- The existing systems also fail to build an informative network for residents in which they can get collaborative society information from other resident'sexperiences, stories and posts.

4.3 REQUIREMENT OF NEW SYSTEM

After we have collected all the required information regarding the web application to be developed, we have to remove all incompleteness, inconsistencies and anomalies from the specification. Following are some remedies and unique features that will be added in the new system:

4.3.1 FUNCTIONAL REQUIREMENTS:

- The web application will give recommendations to user based on his preferences and others‘ ratings.
- The web application will help user to search for travel information in collaborative social environment where user can discover other users‘ trips and reviews.
- The web application will provide a blog like system for tourists where they can post their travel stories and read other tourist‘s posts etc. to gain travel information
- The trip management facilities will be provided to reduce memory burden of users.

4.3.2 NON-FUNCTIONAL REQUIREMENTS:

- **Security:**

  - Web application will be accessible only after login successfully i.e. login credentials must be correct.
  - Web application should use some hash based encryption algorithm to protect critical databases like user account details etc.

- **Reliability:**

  - The database of projects, project tasks and client maintained by the system should be correct and maintained up to date.

- **Usability:**

  - The web application will provide user friendly environment to its users and must be easy to access its various features.

-

- **Scalability:**

    - The system will be scalable enough to be able to add any additional functionality even after the project is developed once.

4.4 FEASIBILITY STUDY

- **Technical Feasibility:**

Technical feasibility of a project determines whether a project can be developed using the technology on hand. The system is technically feasible as the front-end and the back-end required for it is available and already installed.

- **Operational Feasibility:**

In the system operational feasibility, checks are made whether the user who is going to use the system is able to work with the system. If the user does not understand the functionality of the system or is not able to work on the system further development is of waste. Modular web approach can be adapted in this type of system where in as and when one module is done that can be published on the website for user access.

- **Economical Feasibility:**

This system takes a less effort for data access; provide the right information in right time. This is the mobile web application so it can be accessed in any portable device which runs on Android as operating system.

This mobile web application can also provide secrecy of data and right to use this it can also make the web app secure from the outside world. This web app is updated by the experienced faculties so the user can trust on the performance and efficiency of the web application so popularity of the organization will be increased.

Also, no manpower of the company needs to actively participate in the project, so no extra cost of non-productive employee is incurred to the company. Thus, the project is economically feasible for the client also.

4.5 REQUIREMENT VALIDATION:

It means that the created android web app is as per requirement or not? Simply starting whatever we are doing is right or wrong as per requirement? Here we check each & every requirement & compare with our website & that it satisfies the user need.

Requirements validation is concerned with showing that the requirements actually define the system that user wants. If this validation is inadequate, errors in the requirements will be propagated to the system design and implementation. Requirements are checked to discover if they are complete, consistent and in accordance with what users want from the projected system.

4.6FUNCTIONS OF SYSTEM

4.6.1USE CASES:

Use case diagrams used to describe a set of actions (use cases) that some system or systems (subject) should or can perform in collaboration with one or more external users of the system (actors). Each use case should provide some observable and valuable result to the actors or other stakeholders of the system.

Followings are the components of use case diagram:

- **Use Case:** A Use Case describes a sequence of actions that provide something of measurable value to the actor and is drawn as horizontal ellipse.
- **Actors:** An Actor is a Person, Organization, or external system that plays a role in one or more interactions with your systems. Actors are drawn as stick figures.
- **Associations:** Association between actors and use cases are indicated in use case diagrams using solid lines. An association exists whenever an actor is involved with an interaction described by the use case.

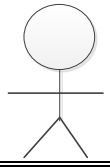
Symbol	Meaning
	Actor
	Use Case
	Association
	<<Include>>/<<Extend>>

Table 4-1 Symbols for Use case diagram

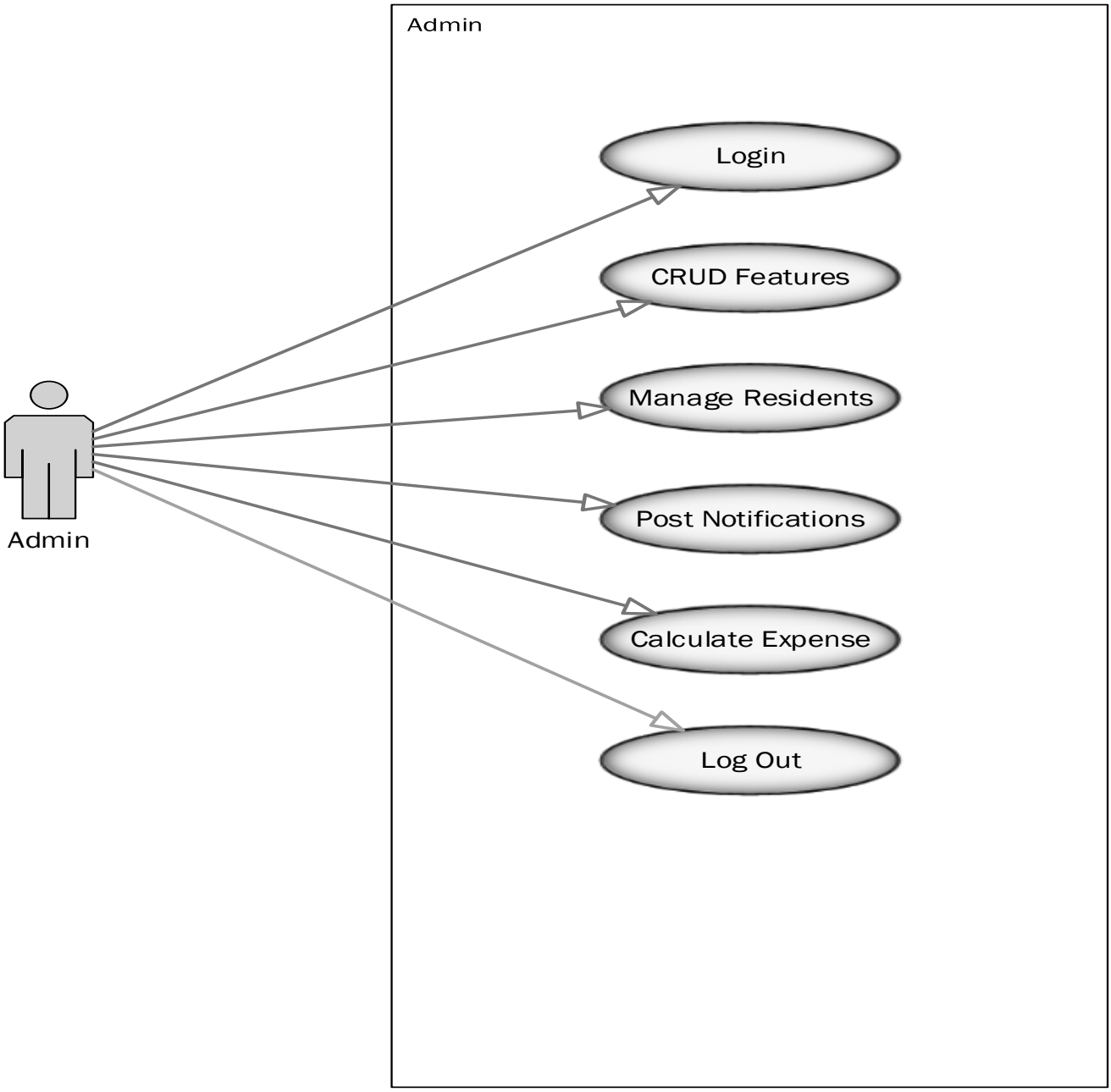


Figure 4.1 Use Case Diagram - User Login/ Register

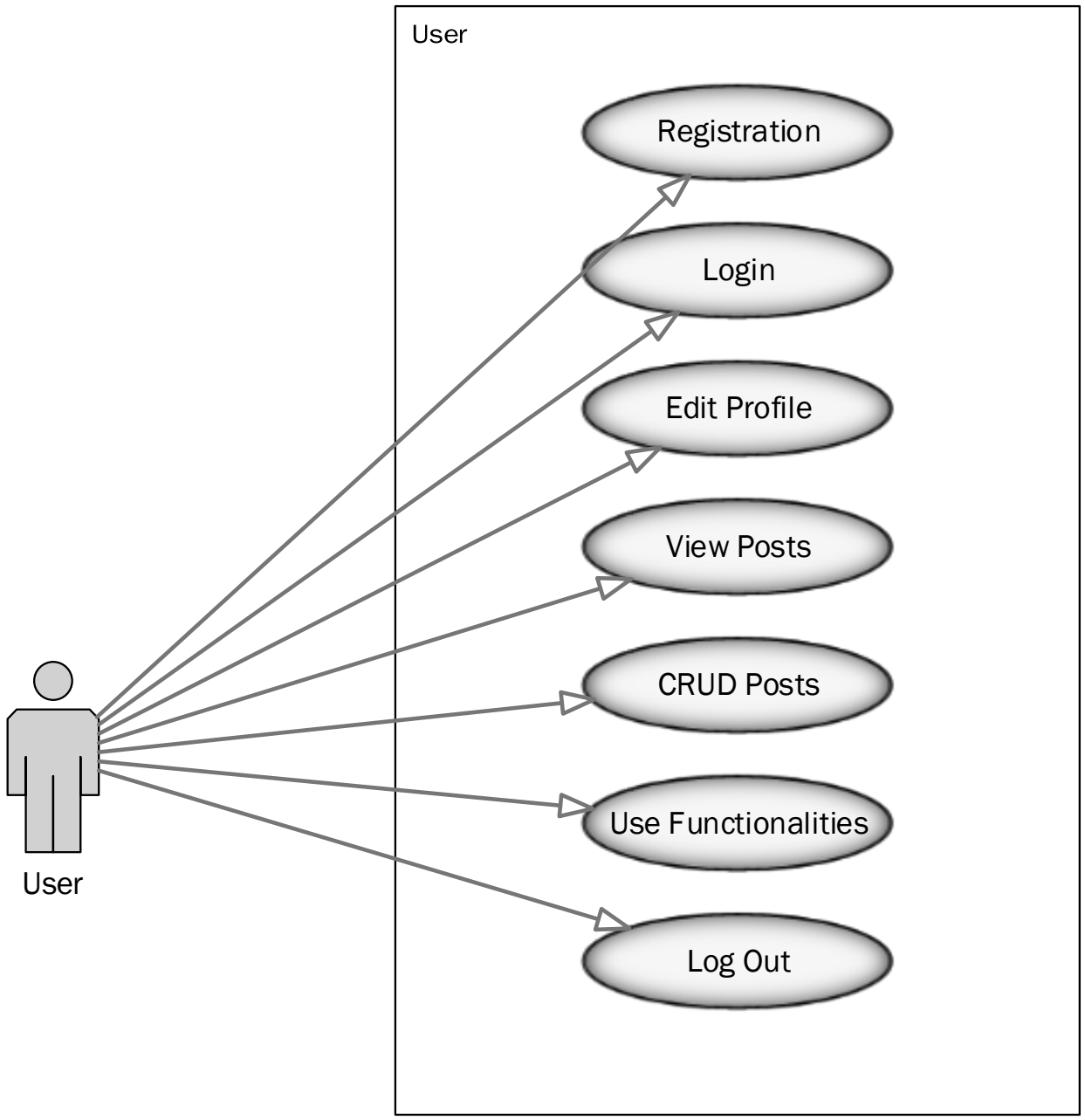


Figure 4.2 Use Case Diagram - For User

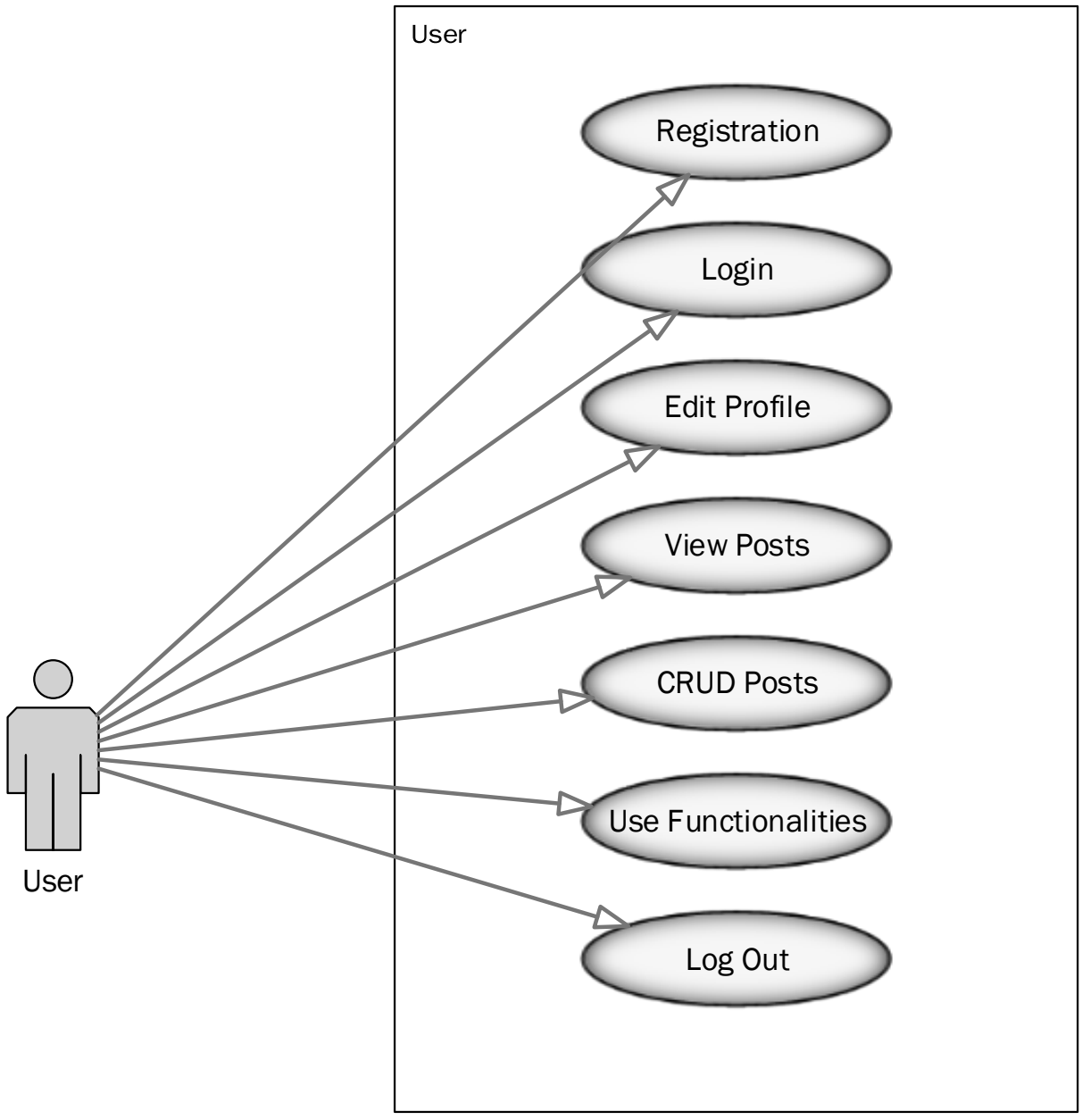


Figure 4.3 Use Case Diagram – For Guest User

4.7DATA MODELLING

4.7.1CLASS DIAGRAM:

It is used for describing structure and behaviour in the use cases. It provides a conceptual model of the system in terms of entities and their relationships.

Following Modifiers are used to indicate visibility of attributes and operations.

- _+‘ is used to denote *Public* visibility (everyone)
- _#‘ is used to denote *Protected* visibility (friends and derived)
- _-‘ is used to denote *Private* visibility (no one)

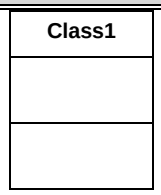
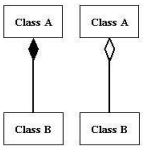
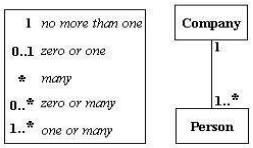
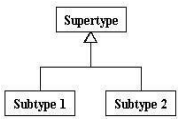
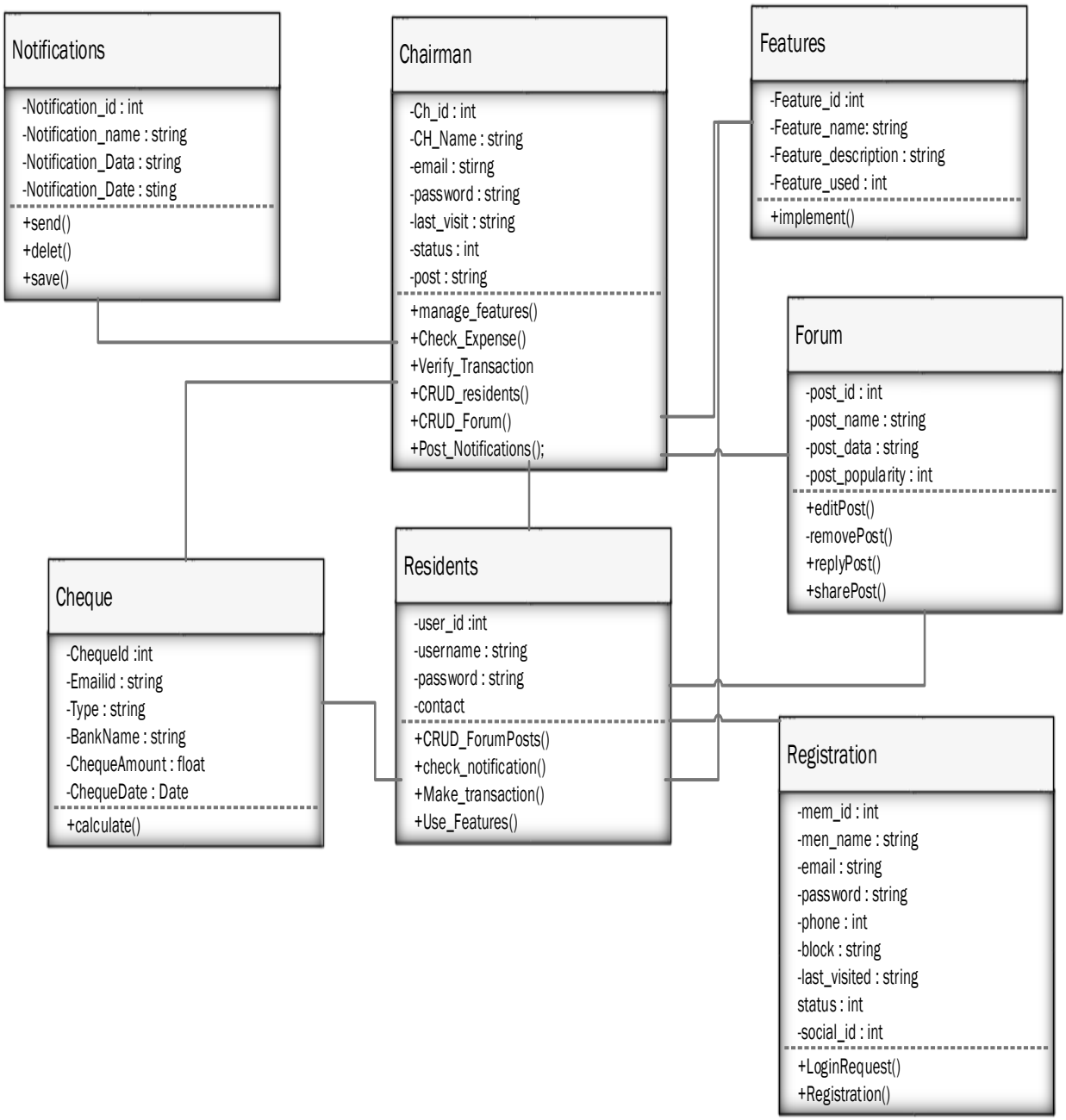
Symbol	Meaning
	Represent The class
	Aggregation-Has or part of relationship
	Multiplicity
	Generalization-Type of relationship

Table 4-2 Symbols For Class Diagram

Figure 4.8 Class Diagram



4.7.2ENTITY RELTIONSHIP DIAGRAM:

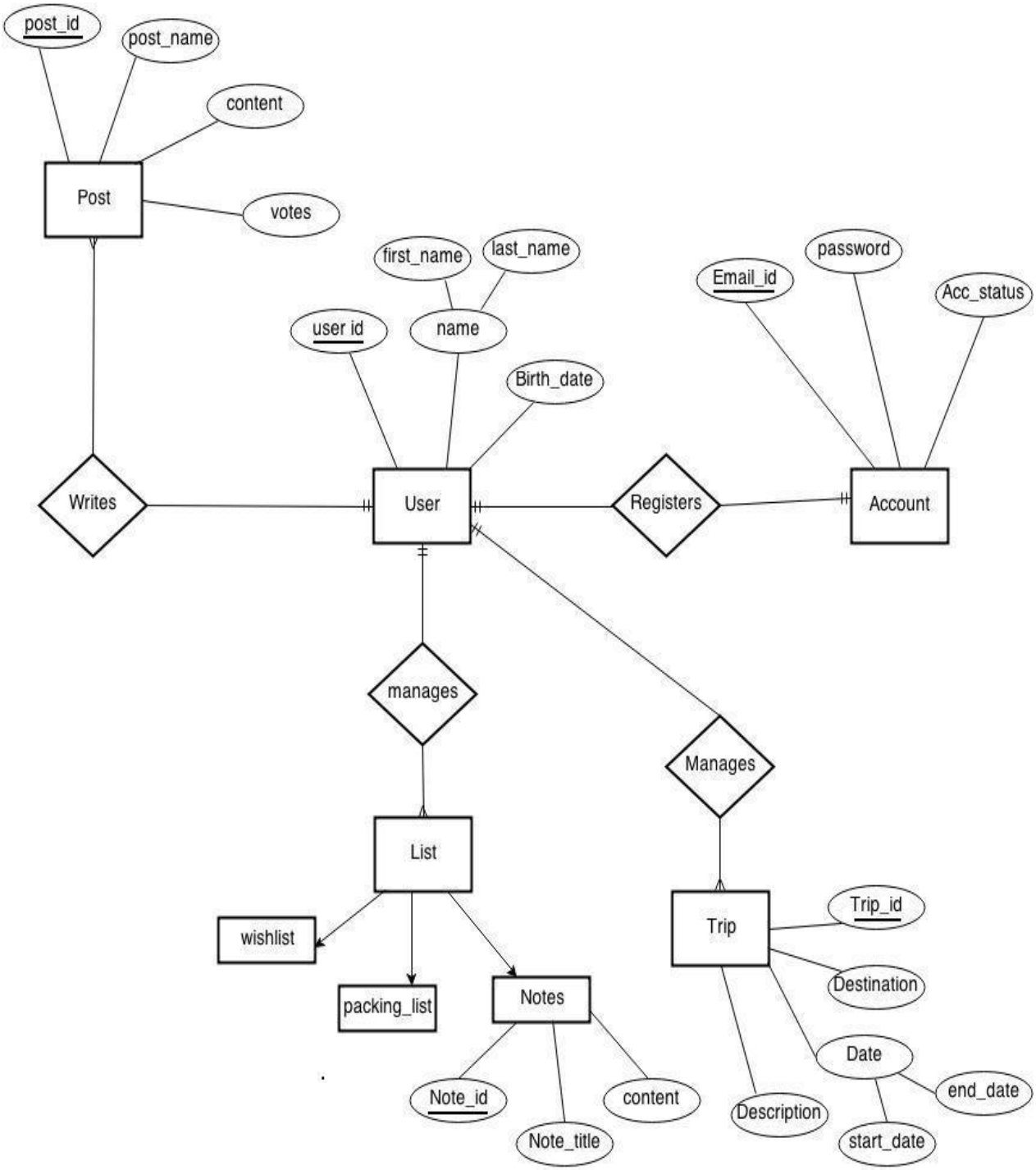


Figure 4.9 E-R Diagram

4.7.3ACTIVITY DIAGRAM:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency.

In the Unified Modelling Language, activity diagrams are intended to model both computational and organizational processes (i.e. workflows). Activity diagrams show the overall flow of control.

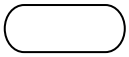
Symbol	Meaning
	Start
	Activity
	Join
	Fork
	Decision
	End

Table 4-3 Symbols for Activity Diagram

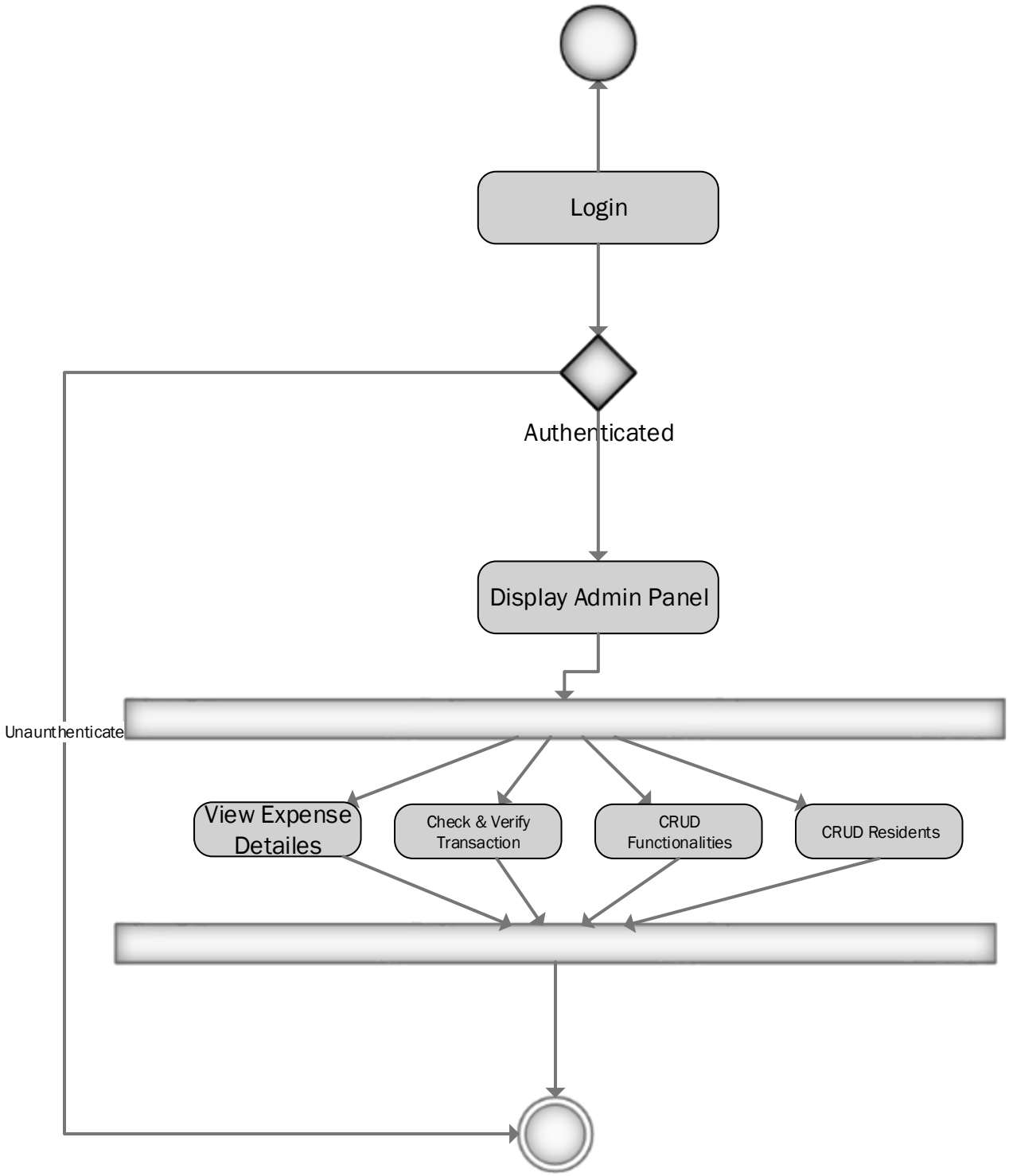


Figure 4.10 Activity Diagram – For Admin

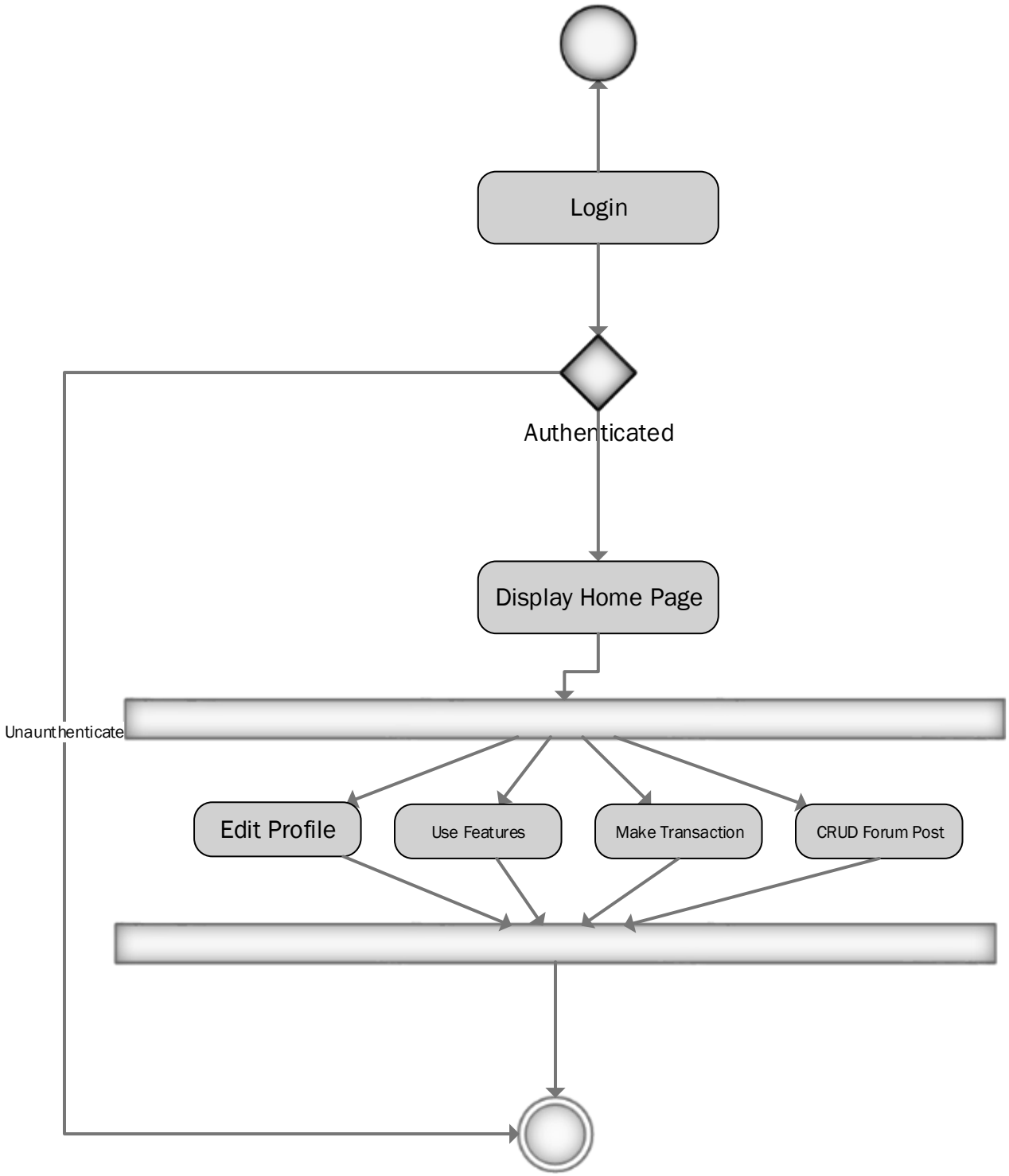


Figure 4.11 Activity Diagram – For User

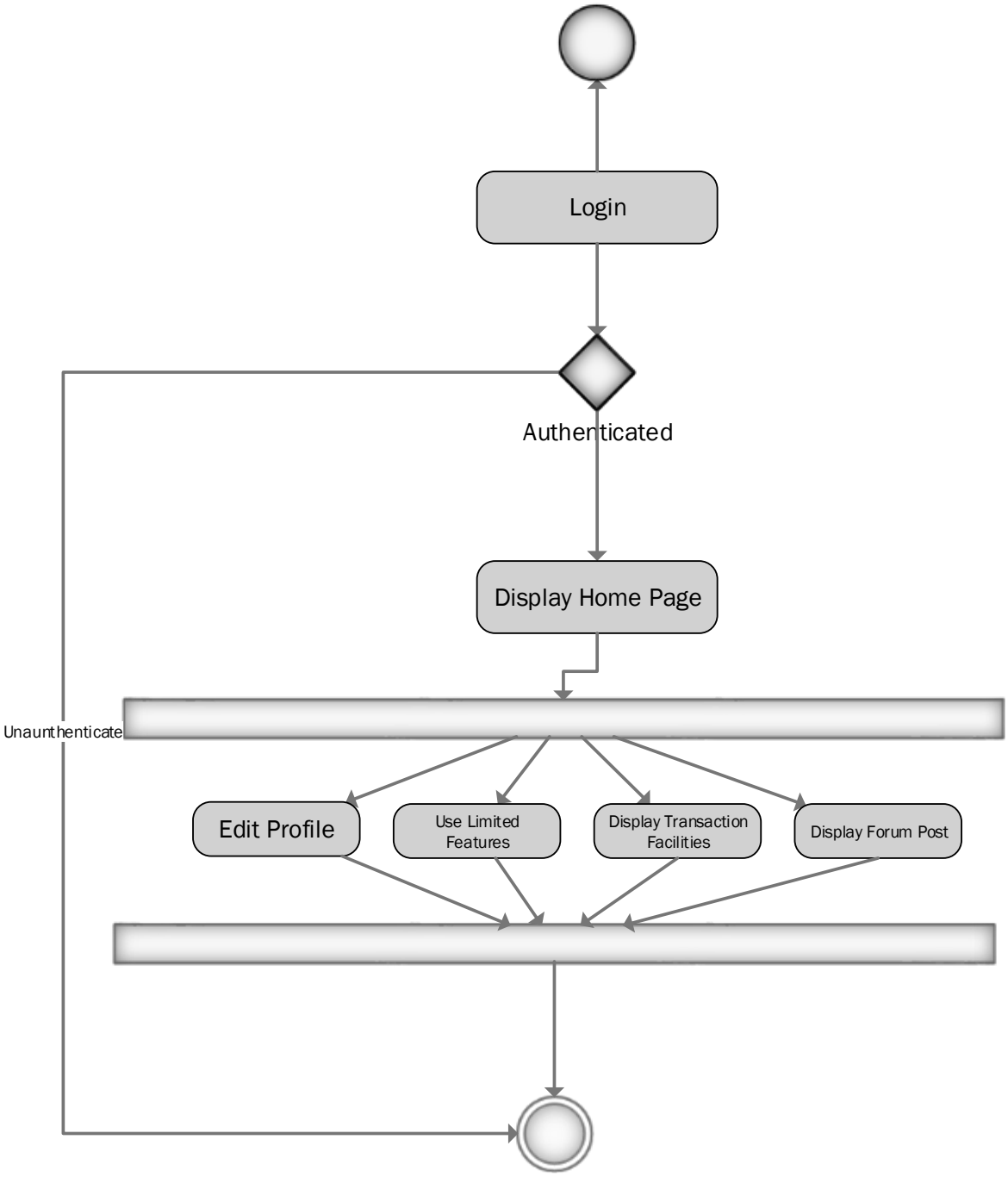


Figure 4.12 Activity diagram – For Guest User

4.7.4 SEQUENCE DIAGRAM

A sequence diagram is a kind of interaction diagramthat shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart.

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development.

Followings are the symbols for sequence diagrams:

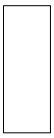
Symbol	Meaning
	Activation
	Entity
	Object Life Line

Table 4-4 Symbols For Sequence Diagram

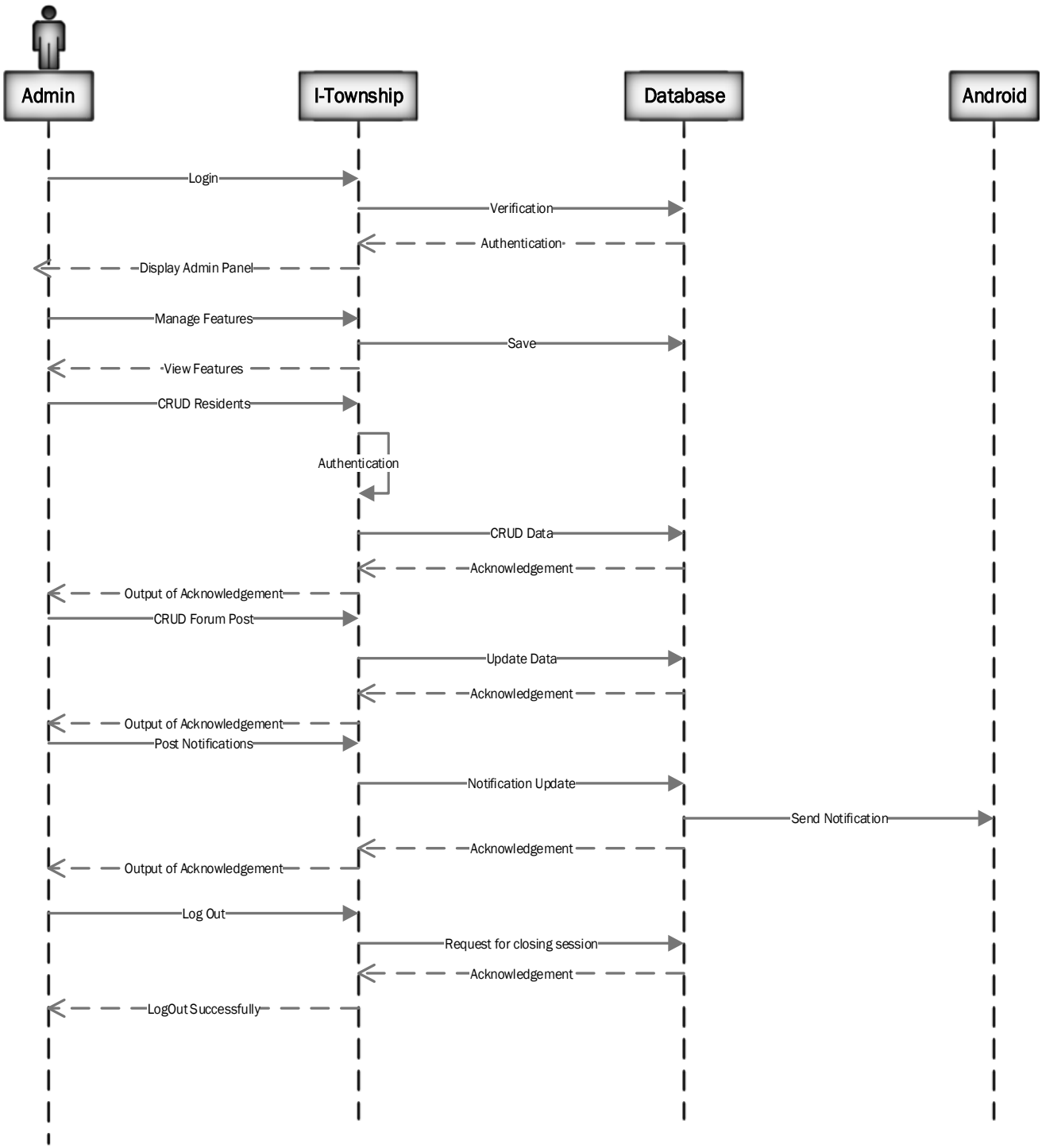


Figure 4.16 Sequence Diagram – For Admin

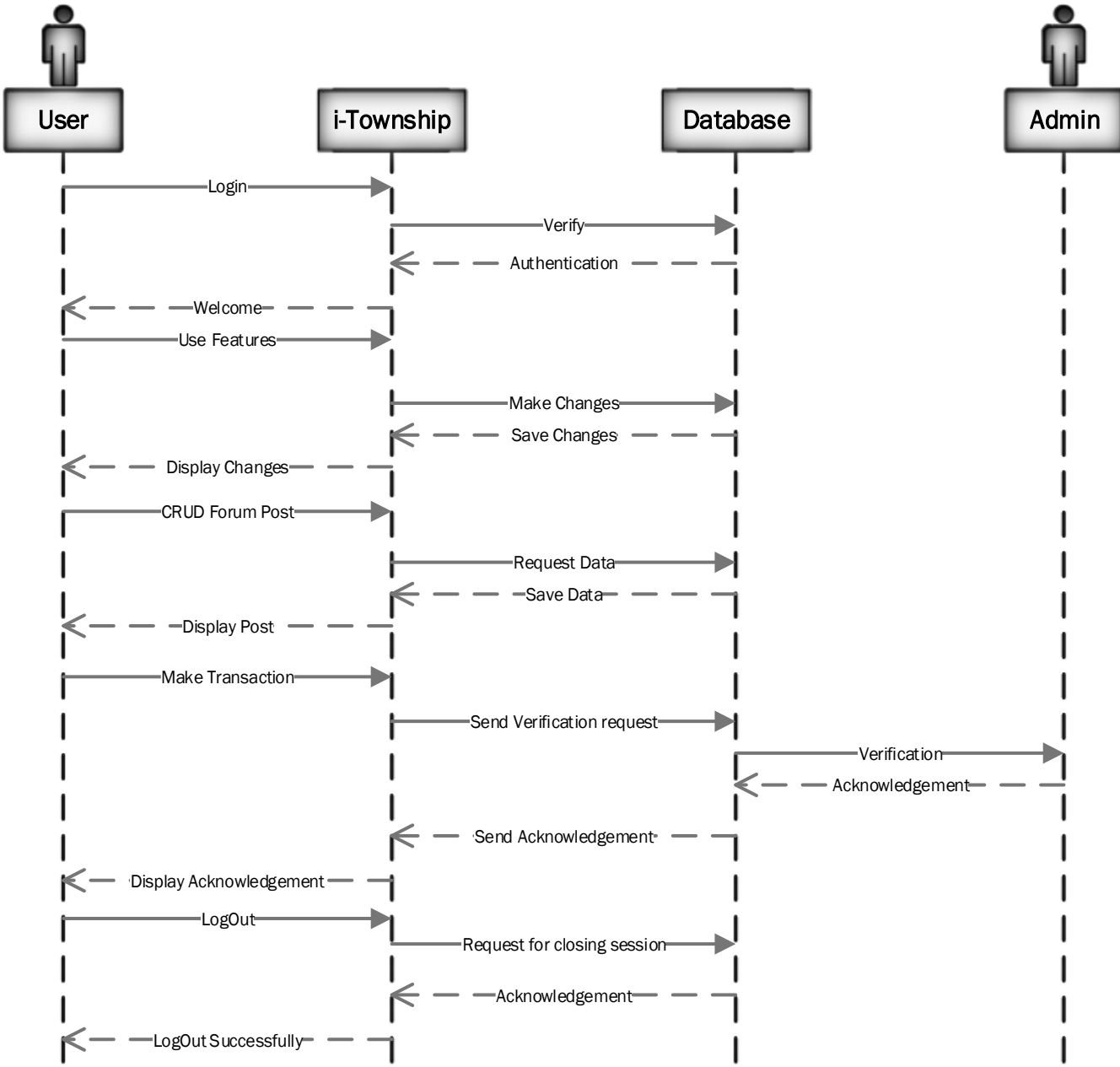


Figure 4.17 Sequence Diagram – For User

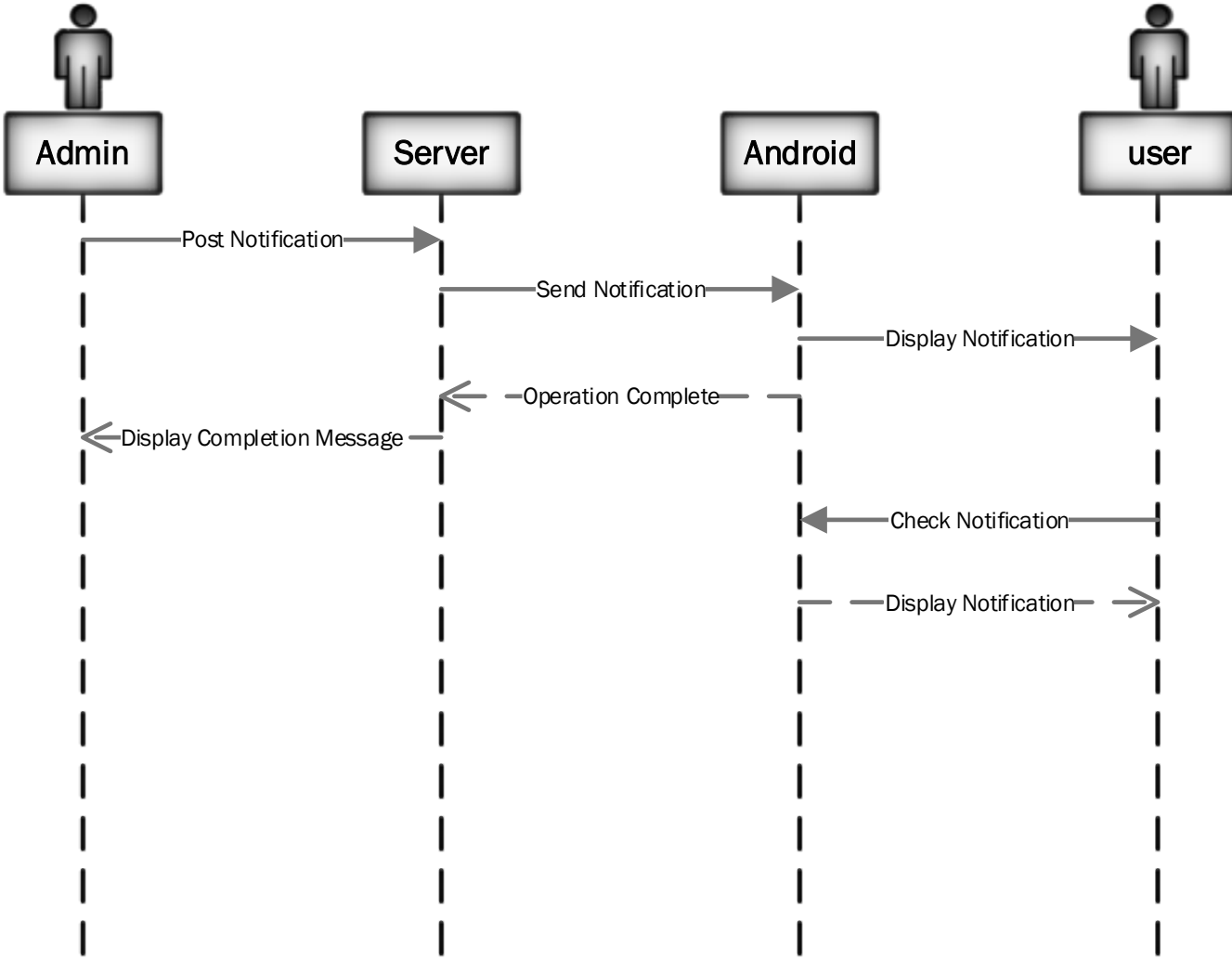


Figure 4.18 Sequence Diagram – For Android

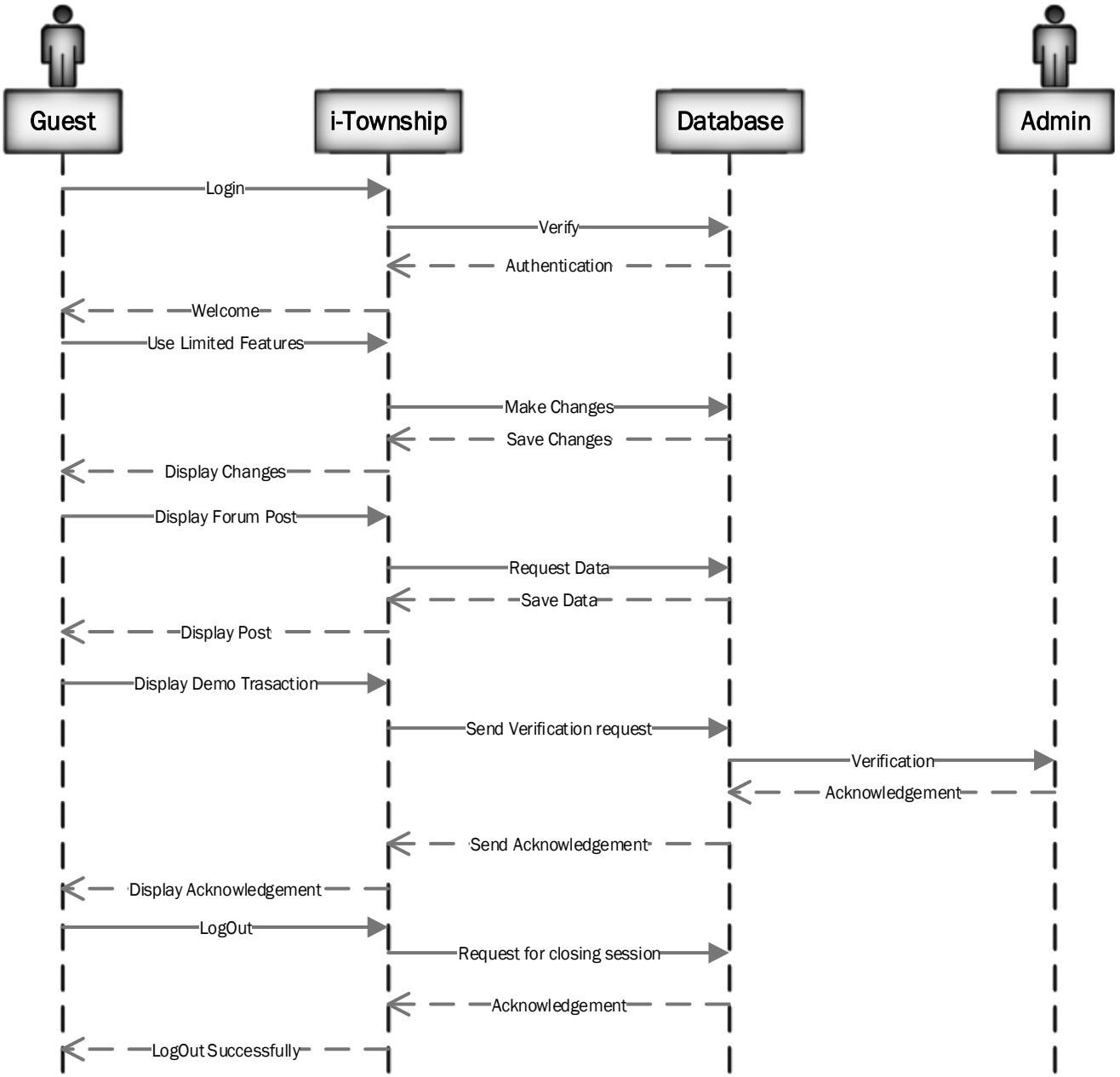


Figure 4.19 Sequence Diagram – For Guest User

4.7.5 DATA DICTIONARY:

Data Dictionary is catalogue –a repository of the elements in the systems. This element centre data and way they are structure to meet user requirements and organization needs.

Need for Data Dictionary:

- To manage details in a large system.
- To communicate a common meaning for all system elements.
- To document the futures of the system.
- To facilitate analysis of the details in order to evaluate characteristics and determine where system should be made.

Table Name :Members			
Table Description: This table describes members account information.			
Primary key:mem_id		**Foreign keys:** None	
Table Structure			
Field Name	**Data Type**	**Size**	**Constraints**
mem_id	INTEGER	11	Primary Key, Auto Increment
mem_name	VARCHAR	50	Not Null
email	VARCHAR	50	Not Null
password	VARCHAR	255	Not Null
phone	BIGINT	10	Not Null
block	VARCHAR	100	Not Null

last_visited	VARCHAR	200	Not Null
Status	TINYINT	1	Not Null
Social_id	INT	100	Null
Icard	VARCHAR	200	Not Null
Mem_type	VARCHAR	10	Not Null

Table 4-5 Members

Table Name :Expense			
Table Description: This table describes all expenses of society.			
Primary key: e_id		**Foreign keys:** member_id	
Table Structure			
Field Name	**Data Type**	**Size**	**Constraints**
E_id	INTEGER	10	Primary Key, Auto Increment
Member_id	INT	10	Not Null
E_amount	VARCHAR	20	Not Null
E_type	VARCHAR	20	Not Null
e-receiver	VARCHAR	20	Not Null
E_category	VARCHAR	20	Not Null

E_date	VARCHAR	40	Not Null
E_Receipt	VARCHAR	100	Not Null
E_notes	VARCHAR	200	Not Null

Table 4-6 Expense

Table Name :Forum			
Table Description: This table describes forum information.			
Primary key: post_id		**Foreign keys:** member_id	
Table Structure			
Field Name	**Data Type**	**Size**	**Constraints**
Post_id	INT	10	Primary Key, Auto Increment
Post_title	VARCHAR	20	Not Null
Post_desc	VARCHAR	500	Not Null
Lastpost	VARCHAR	15	Not Null
Block_name	VARCHAR	10	Not Null
Member_id	INT	10	Not Null
Post_views	INT	10	Not Null

Table 4-7 Countries

Table Name :forum_cmnts			
Table Description: This table describes replies given by resident to forum post.			
Primary key: fc_id		**Foreign keys:** post_id, mem_id	
Table Structure			
Field Name	**Data Type**	**Size**	**Constraints**
Fc_id	INTEGER	10	Primary Key, Auto Increment
Post_id	INT	10	Foreign Key
Mem_id	INT	10	Foreign Key
Cmnt	VARCHAR	500	Not Null
Cmnt_likes	INT	10	Null
Cmnt_dislikes	INT	10	Null

Table 4-8 forum_cmnts

Table Name :meetings			
Table Description: This table describes information about meetings events of society.			
Primary key: m_id		**Foreign keys:** m_presentid, m_agendaid	
Table Structure			
Field Name	**Data Type**	**Size**	**Constraints**
M_id	INT	10	Primary Key, Auto Increment

M_name	VARCHAR	20	Not Null
M_about	VARCHAR	100	Not Null
M_place	VARCHAR	30	Not Null
M_presentid	INT	10	Foreign Key
M_agendaid	INT	10	Foreign Key
M_sign	VARCHAR	200	Not Null
M_date	VARCHAR	10	Not Null

Table 4-9 Trips

Table Name :notification			
Table Description: This table describes information about notifications.			
Primary key: noty_id		**Foreign keys:** None	
Table Structure			
Field Name	**Data Type**	**Size**	**Constraints**

Noty_id	INTEGER	10	Primary Key, Auto Increment
EmailId	VARCHAR	100	Not Null
regId	VARCHAR	200	Not Null
Message	VARCHAR	100	Not Null

Table 4-10 Notification Table

Table Name :present_members			
Table Description: This table describes information about present members during meeting of society.			
Primary key:p_presentid		**Foreign keys:** meeting_id, p_presentIcard	
Table Structure			
Field Name	**Data Type**	**Size**	**Constraints**
P_presentid	INTEGER	10	Primary Key, Auto Increment
Meeting_id	INTEGER	20	Foreign Key
P_presenticard	INTEGER	20	Foreign Key

Table 4-11 Present_Members

Table Name :agenda	
Table Description: This describes information about main agenda of meeting.	
Primary key:a_id	**Foreign keys:** meeting_id
Table Structure	

Field Name	Data Type	Size	Constraints
A_id	INTEGER	10	Primary Key, Auto Increment
Meeting_id	INTEGER	10	Foreign Key
A_title	VARCHAR	100	Not Null
A_discussion	VARCHAR	500	Not Null
A_conclusion	VARCHAR	500	Not Null

Table 4-12 Agenda

Table Name :tasks			
Table Description: This table describes information about task information.			
Primary key:tsk_id		**Foreign keys: mem_id, chr_id**	
Table Structure			
Field Name	**Data Type**	**Size**	**Constraints**
Tsk_id	INTEGER	10	Primary Key, Auto Increment
Mem_id	INTEGER	10	Foreign Key
Chr_id	INTEGER	10	Foreign Key
Task_desc	VARCHAR	100	Not Null
Priority	VARCHAR	10	Not Null
Progress	INT	10	Null

Deadline	VARCHAR	10	Not Null

Table 4-13 Tasks

Table Name :payments			
Table Description: This table describes all transaction occurred using payment gateway.			
Primary key:txn_id		**Foreign keys:** None	
Table Structure			
Field Name	**Data Type**	**Size**	**Constraints**
Txn_id	INTEGER	11	Primary Key, AUTO INCREMENT
First_name	VARCHAR	200	Not Null
Last_name	VARCHAR	200	Not Null
Payer_email	VARCHAR	200	Not Null
Payment_gross	VARCHAR	200	Not Null
Payment_fee	Longblob		

Table 4-14 Payments

4.8FUNCTIONAL & BEHAVIOURAL MODELING:

4.8.1DATA FLOW DIAGRAM:

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modelling its process aspects. A DFD is often used as a preliminary step to create an overview of the system, which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design).

A DFD shows what kind of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored.

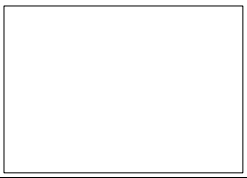
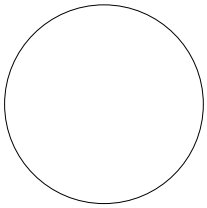
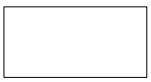
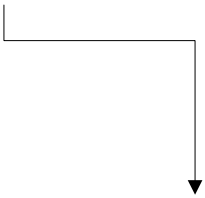
Symbol	Meaning
	Entity
	Data Process
	External Integrator
	Dynamic Connector
	Data store

Table 4-21 Symbols for Data Flow Diagram

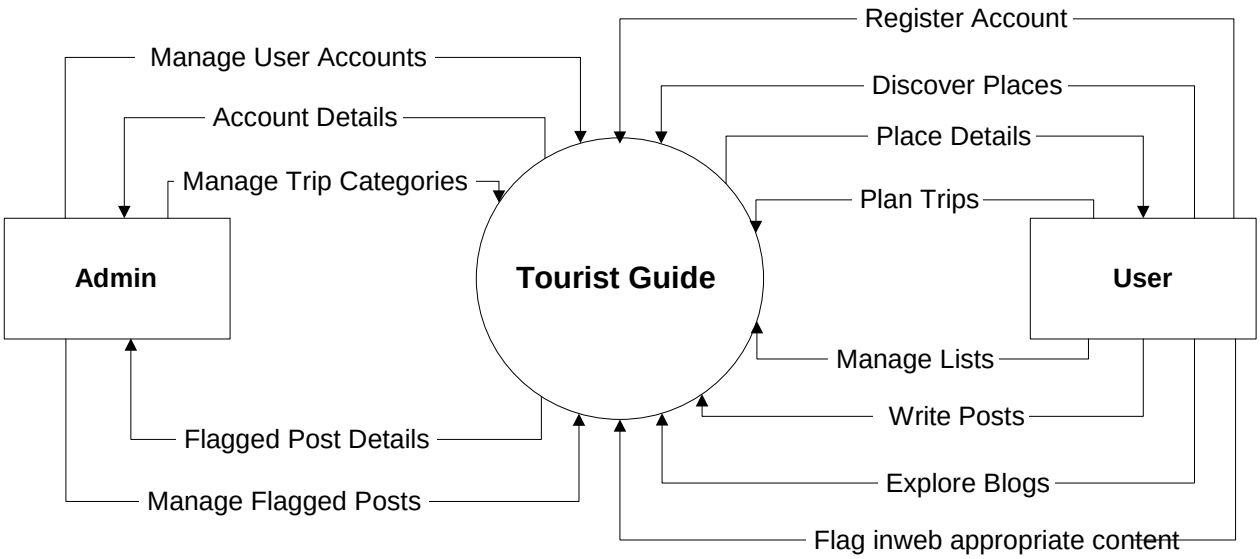


Figure 4.21 DFD Level 0 Context Level Diagram

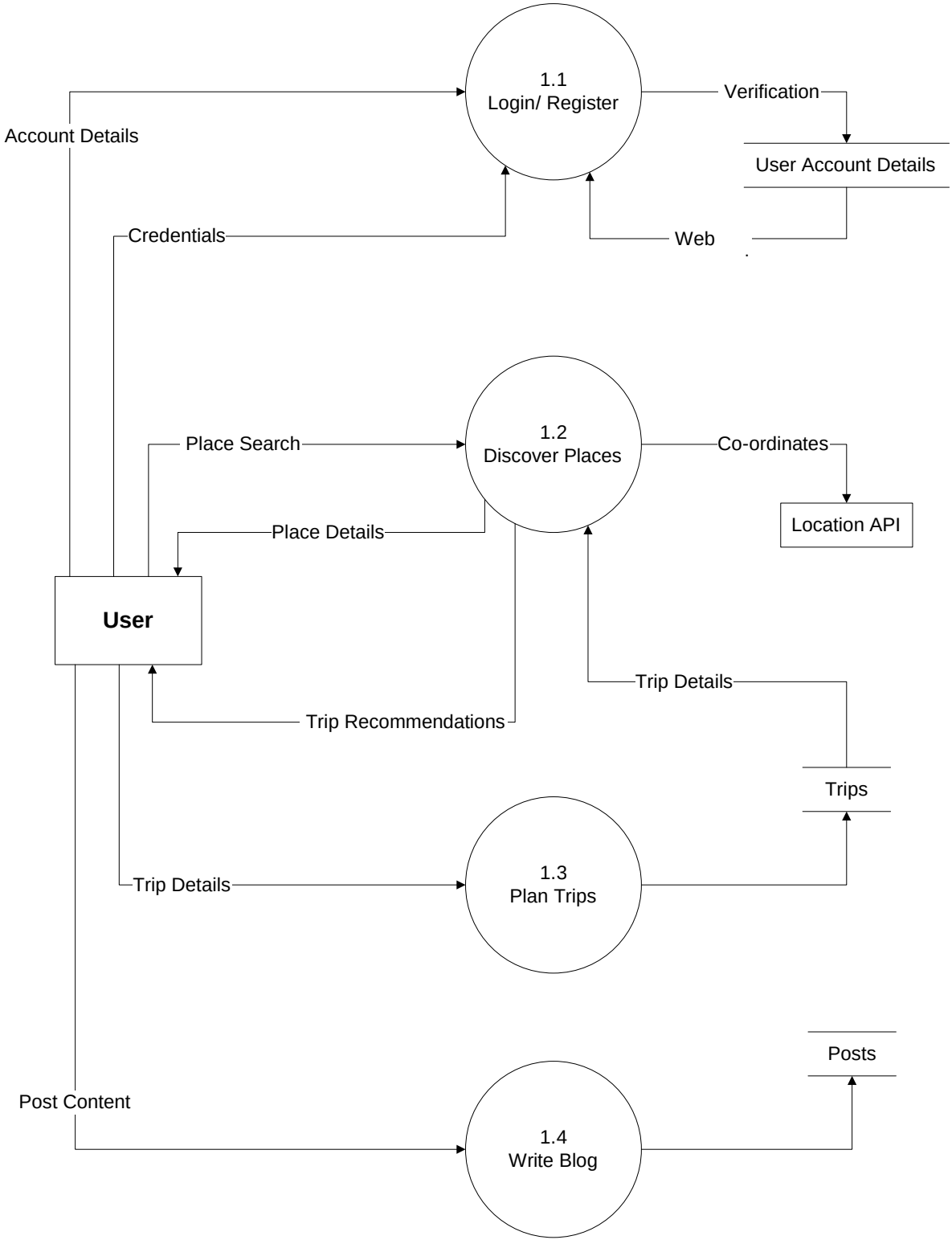


Figure 4.22 DFD Level 1 For User

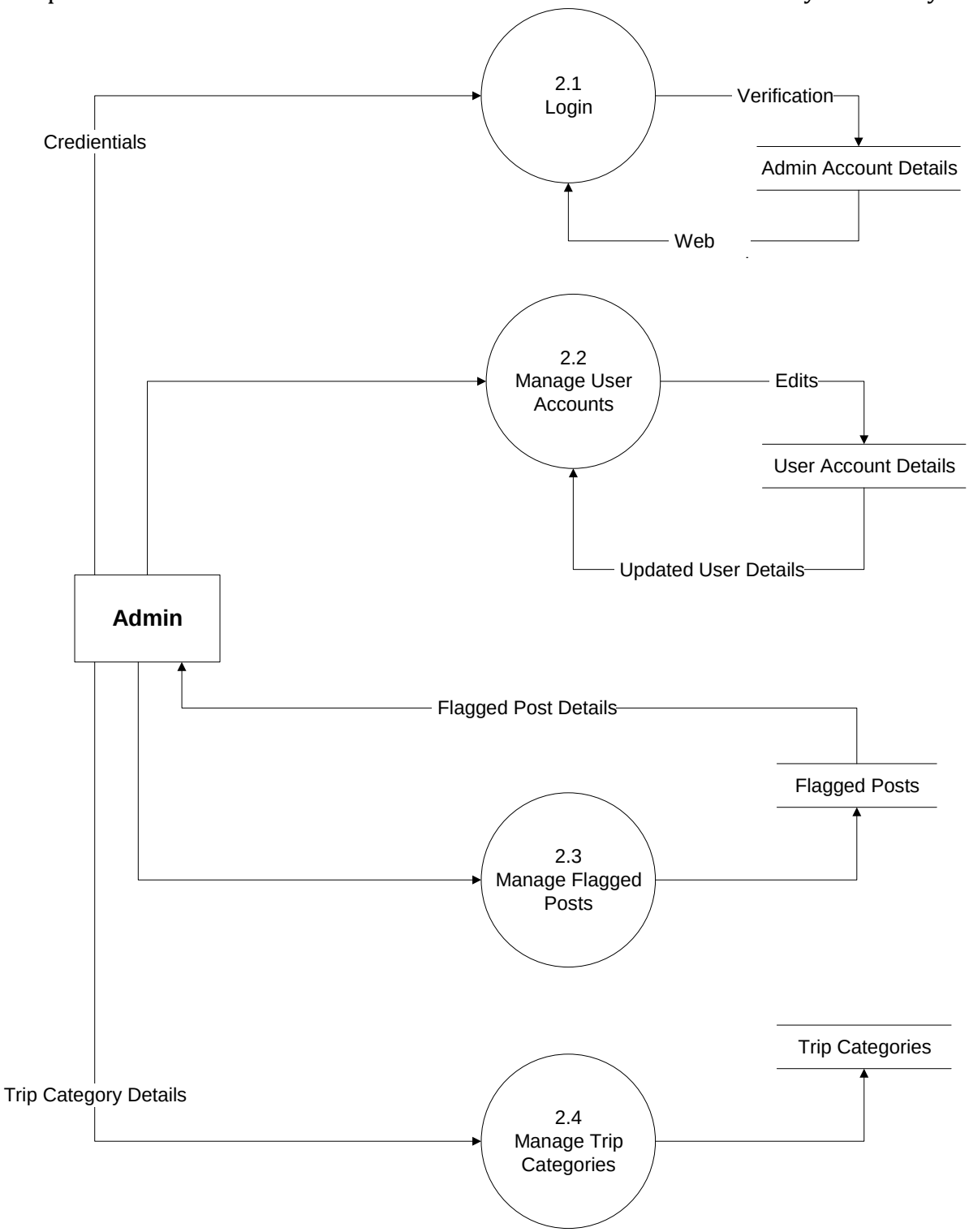


Figure 4.23 DFD Level 1 for Admin

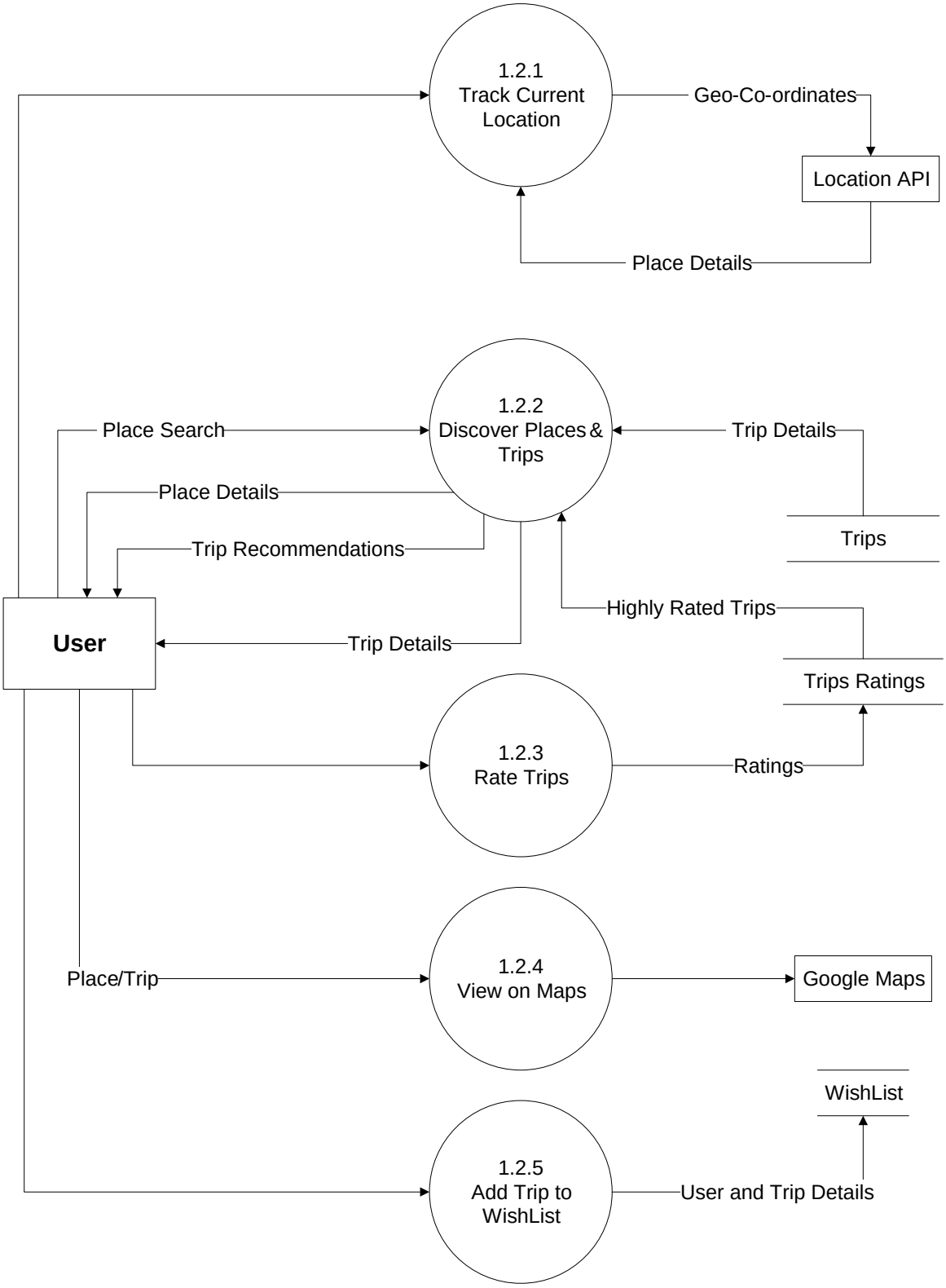


Figure 4.24 DFD Level 2 - Discover Places Module

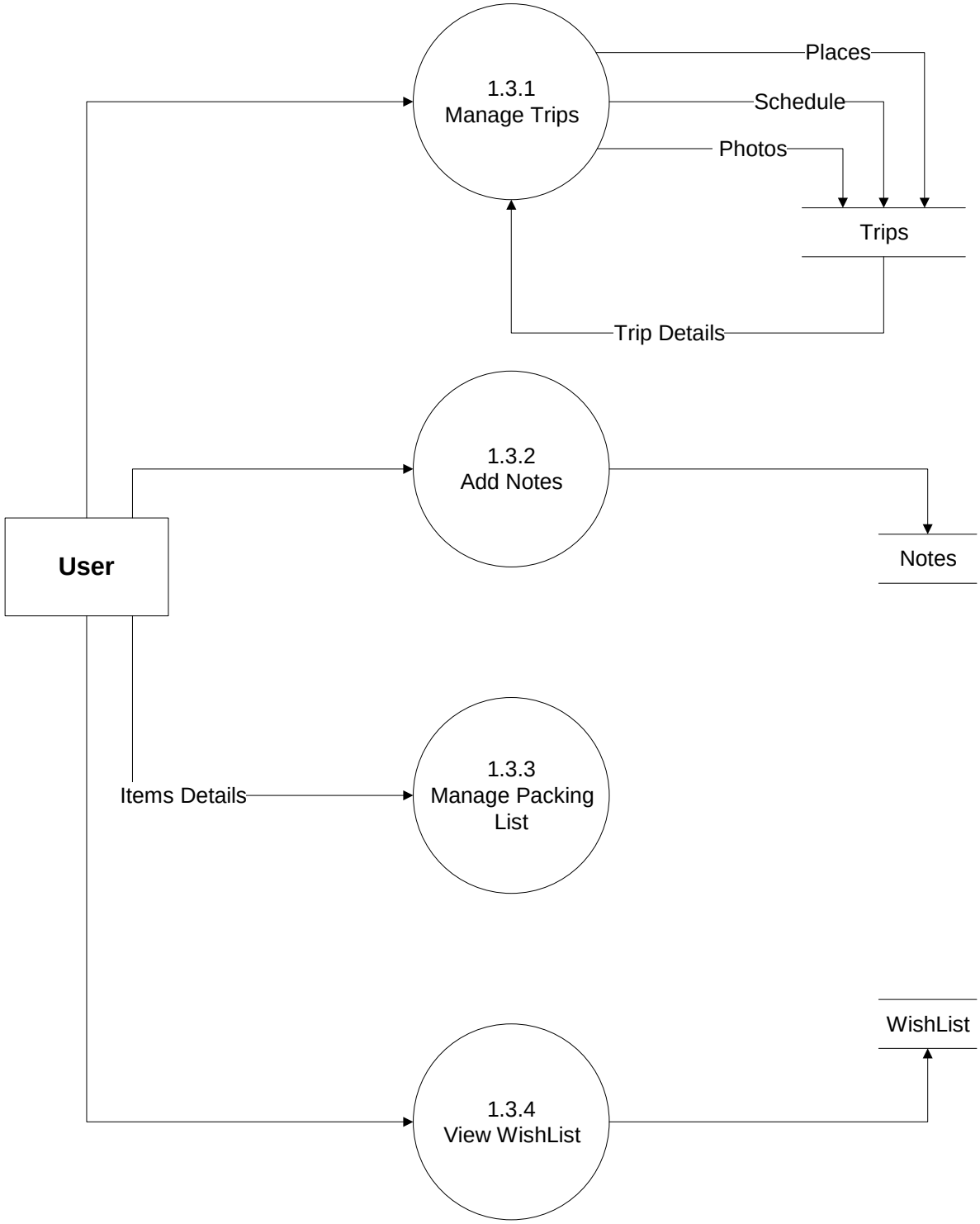


Figure 4.25 DFD Level 2 - Trip Planning Module

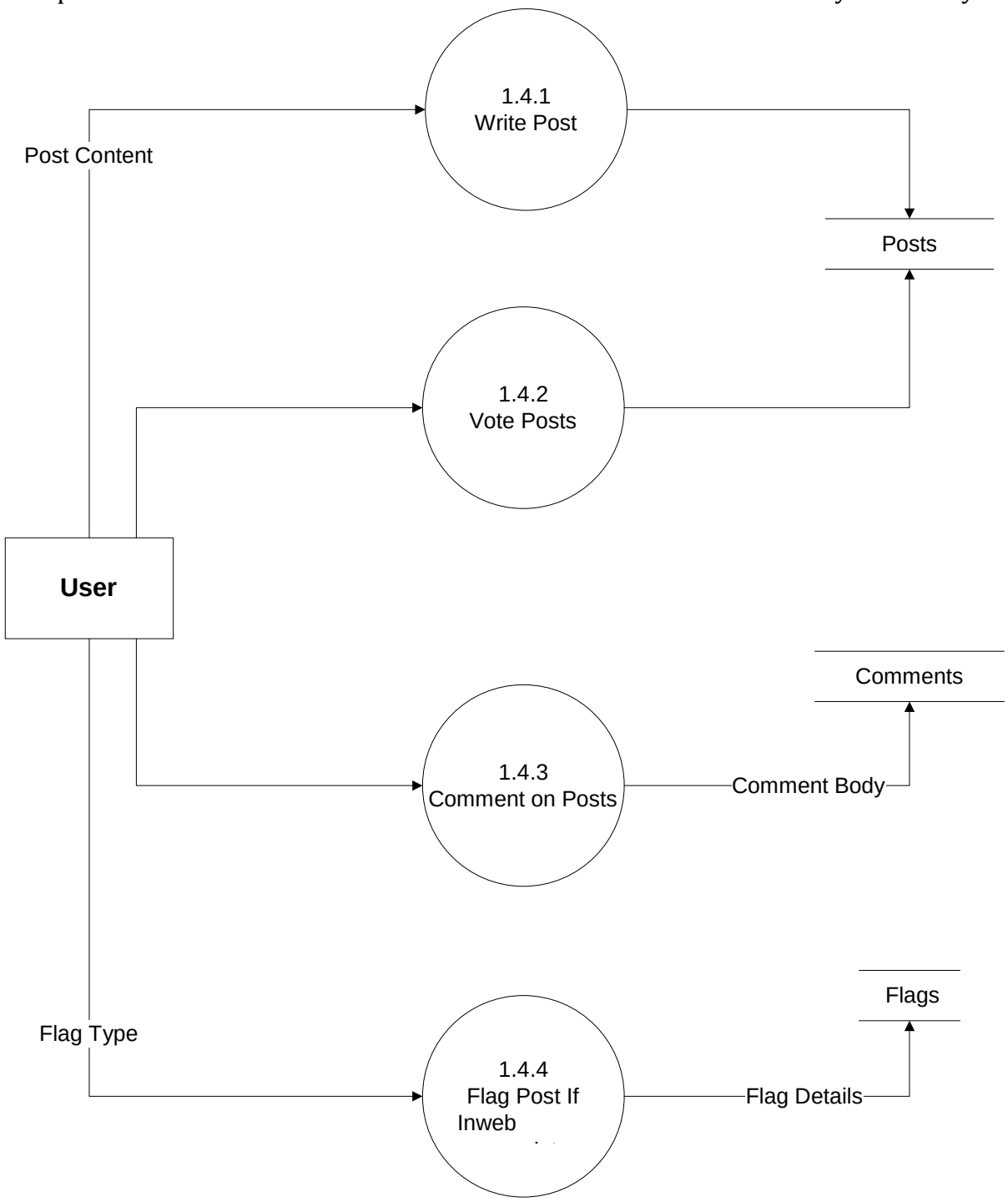


Figure 4.26 DFD Level 2 - Blog Module

4.9MAIN MODULES OF NEW SYSTEM

The modules involved in the Project are:

- **Admin**

- Calculate expense and verify transactions
- Update Forum post
- Add and Manage Functionalities
- Add Users
- Post Notifications

- **User**

- Edit profile
- Add Expense
- Insert, Update and delete posts
- View Expense Records
- Use features
- Make Transactions

4.10 SELECTION OF HARDWARE &SOFTWARE & JUSTIFICATION

❖ **Hardware**

- Pentium processor
- Windows OS
- 1GB RAM
- 4GB Hard Disk

❖ **Software**

- Windows Operating System
- Eclipse with Android SDK API-19
- SQL yog: - As a Back End

Justification:

- Eclipse IDE is best for developing web based web application and Android Studio is best for Android web applications and it is completely open source. Since Android web applications use Java as programming language.

CHAPTER

5

SYSTEM DESIGN

5.1 Database/Data Structure Design

5.1.1 Mweb apping Object/Classes to Object

5.1.2 Logical Description of Data

5.2 System Procedural Design

5.2.1 Design Pseudo Code or Algorithm for Method or Operation

5.2.2 Flow Chart Diagram

5.3 Input/output & Interface Design

5.3.1 Samples of Forms, Reports & Interface

CHAPTER:5 SYSTEM DESIGN

5.1 DATABASE/ DATA STRUCTURE DESIGN

5.1.1 MWEB APPING OBJECT/CLASSES TO TABLES

members

Fields

Field	Type	Collation	Null	Key	Default	Extra	Privileges	Comment
mem_id	int(200)	(NULL)	NO	PRI	(NULL)	auto_increment	select,insert,update,references	
mem_name	varchar(200)	utf8_general_ci	NO		(NULL)		select,insert,update,references	
email	varchar(300)	utf8_general_ci	NO		(NULL)		select,insert,update,references	
password	varchar(200)	utf8_general_ci	NO		(NULL)		select,insert,update,references	
phone	bigint(10)	(NULL)	YES		(NULL)		select,insert,update,references	
block	varchar(100)	utf8_general_ci	YES		(NULL)		select,insert,update,references	
last_visited	varchar(200)	utf8_general_ci	YES		(NULL)		select,insert,update,references	
status	tinyint(1)	(NULL)	YES		(NULL)		select,insert,update,references	
social_id	int(100)	(NULL)	YES		(NULL)		select,insert,update,references	
icard	varchar(200)	utf8_general_ci	YES		(NULL)		select,insert,update,references	
mem_type	varchar(10)	utf8_general_ci	YES		(NULL)		select,insert,update,references	

Indexes

Table	Non unique	Key name	Seq in index	Column name	Collation	Cardinality	Sub part	Packed	Null	Index type	Comment	Index comment
members	0	PRIMARY	1	mem_id	A	6	(NULL)	(NULL)		BTREE		

Figure 5.1 members table

notification

Fields

Field	Type	Collation	Null	Key	Default	Extra	Privileges	Comment
noty_id	int(10)	(NULL)	NO	PRI	(NULL)		select,insert,update,references	
emailId	varchar(100)	utf8_general_ci	YES		(NULL)		select,insert,update,references	
regId	varchar(200)	utf8_general_ci	YES		(NULL)		select,insert,update,references	

Indexes

Table	Non unique	Key name	Seq in index	Column name	Collation	Cardinality	Sub part	Packed	Null	Index type	Comment	Index comment
notification	0	PRIMARY	1	noty_id	A	0	(NULL)	(NULL)		BTREE		

Figure 5.2 notification table

payments

Fields

Field	Type	Collation	Null	Key	Default	Extra	Privileges	Comment
txn_id	int(10)	(NULL)	YES		(NULL)		select,insert,update,references	
first_name	varchar(200)	utf8_general_ci	YES		(NULL)		select,insert,update,references	
last_name	varchar(200)	utf8_general_ci	YES		(NULL)		select,insert,update,references	
payer_email	varchar(200)	utf8_general_ci	YES		(NULL)		select,insert,update,references	
payment_gross	varchar(200)	utf8_general_ci	YES		(NULL)		select,insert,update,references	
payment_fee	longblob	(NULL)	YES		(NULL)		select,insert,update,references	

Indexes

Figure 5.3 payments table

tasks

Fields

Field	Type	Collation	Null	Key	Default	Extra	Privileges	Comment
tsk_id	int(10)	(NULL)	NO	PRI	(NULL)	auto_increment	select,insert,update,references	
mem_id	int(10)	(NULL)	NO	MUL	(NULL)		select,insert,update,references	
chr_id	int(10)	(NULL)	NO	MUL	(NULL)		select,insert,update,references	
task_desc	varchar(100)	utf8_general_ci	NO		(NULL)		select,insert,update,references	
priority	varchar(10)	utf8_general_ci	YES		(NULL)		select,insert,update,references	
progress	int(10)	(NULL)	YES		(NULL)		select,insert,update,references	
deadline	varchar(10)	utf8_general_ci	YES		(NULL)		select,insert,update,references	

Indexes

Table	Non unique	Key name	Seq in index	Column name	Collation	Cardinality	Sub part	Packed	Null	Index type	Comment	Index comment
tasks	0	PRIMARY	1	tsk_id	A	4	(NULL)	(NULL)		BTREE		
tasks	1	mem_id	1	mem_id	A	4	(NULL)	(NULL)		BTREE		
tasks	1	chr_id	1	chr_id	A	2	(NULL)	(NULL)		BTREE		

Foreign Key Relationships

FK Id	Reference Table	Source Column	Target Column	Extra Info
tasks_ibfk_1	members	`mem_id`	`mem_id`	
tasks_ibfk_2	members	`chr_id`	`mem_id`	

Figure 5.4 tasks table

meetings

Fields

Field	Type	Collation	Null	Key	Default	Extra	Privileges	Comment
m_id	int(10)	(NULL)	NO	PRI	(NULL)	auto_increment	select,insert,update,references	
m_name	varchar(20)	utf8_general_ci	NO		(NULL)		select,insert,update,references	
m_about	varchar(100)	utf8_general_ci	NO		(NULL)		select,insert,update,references	
m_place	varchar(30)	utf8_general_ci	NO		(NULL)		select,insert,update,references	
m_presentid	int(10)	(NULL)	YES	MUL	(NULL)		select,insert,update,references	
m_agendaid	int(10)	(NULL)	YES	MUL	(NULL)		select,insert,update,references	
m_sign	varchar(200)	utf8_general_ci	YES		(NULL)		select,insert,update,references	
m_date	varchar(10)	utf8_general_ci	YES		(NULL)		select,insert,update,references	

Indexes

Table	Non unique	Key name	Seq in index	Column name	Collation	Cardinality	Sub part	Packed	Null	Index type	Comment	Index comment
meetings	0	PRIMARY	1	m_id	A	17	(NULL)	(NULL)		BTREE		
meetings	1	meetings_ibfk_1	1	m_presentid	A	17	(NULL)	(NULL)	YES	BTREE		
meetings	1	meetings_ibfk_2	1	m_agendaid	A	17	(NULL)	(NULL)	YES	BTREE		

Foreign Key Relationships

FK Id	Reference Table	Source Column	Target Column	Extra Info
meetings_ibfk_1	present_members	`m_presentid`	`meeting_id`	ON DELETE SET NULL ON UPDATE SET NULL;
meetings_ibfk_2	agenda	`m_agendaid`	`meeting_id`	ON DELETE SET NULL ON UPDATE SET NULL

Figure 5.5 meetings table

agenda

Fields

Field	Type	Collation	Null	Key	Default	Extra	Privileges	Comment
meeting_id	int(10)	(NULL)	YES	MUL	(NULL)		select,insert,update,references	
a_id	int(10)	(NULL)	NO	PRI	(NULL)	auto_increment	select,insert,update,references	
a_title	varchar(100)	utf8_general_ci	NO		(NULL)		select,insert,update,references	
a_discussion	varchar(500)	utf8_general_ci	YES		(NULL)		select,insert,update,references	
a_conclusion	varchar(500)	utf8_general_ci	YES		(NULL)		select,insert,update,references	

Indexes

Table	Non unique	Key name	Seq in index	Column name	Collation	Cardinality	Sub part	Packed	Null	Index type	Comment	Index comment
agenda	0	PRIMARY	1	a_id	A	14	(NULL)	(NULL)		BTREE		
agenda	1	agenda_ibfk_1	1	meeting_id	A	14	(NULL)	(NULL)	YES	BTREE		

Foreign Key Relationships

FK Id	Reference Table	Source Column	Target Column	Extra Info
agenda_ibfk_1	meetings	`meeting_id`	`m_id`	ON DELETE CASCADE ON UPDATE CASCADE

Figure 5.6 agenda table

present_members

Fields

Field	Type	Collation	Null	Key	Default	Extra	Privileges	Comment
meeting_id	int(20)	(NULL)	NO	MUL	(NULL)		select,insert,update,references	
p_presentid	int(10)	(NULL)	NO	PRI	(NULL)	auto_increment	select,insert,update,references	
p_presenticard	varchar(20)	utf8_general_ci	NO		(NULL)		select,insert,update,references	

Indexes

Table	Non unique	Key name	Seq in index	Column name	Collation	Cardinality	Sub part	Packed	Null	Index type	Comment	Index comment
present_members	0	PRIMARY	1	p_presentid	A	25	(NULL)	(NULL)		BTREE		
present_members	1	meeting_id	1	meeting_id	A	25	(NULL)	(NULL)		BTREE		

Foreign Key Relationships

FK Id	Reference Table	Source Column	Target Column	Extra Info
present_members_ibfk_1	meetings	`meeting_id`	`m_id`	

Figure 5.7 present_members table

forum

Fields

Field	Type	Collation	Null	Key	Default	Extra	Privileges	Comment
post_id	int(10)	(NULL)	NO	PRI	(NULL)	auto_increment	select,insert,update,references	
post_title	varchar(20)	utf8_general_ci	NO		(NULL)		select,insert,update,references	
post_desc	varchar(500)	utf8_general_ci	YES		(NULL)		select,insert,update,references	
lastpost	varchar(15)	utf8_general_ci	YES		(NULL)		select,insert,update,references	
block_name	varchar(10)	utf8_general_ci	YES		(NULL)		select,insert,update,references	
member_id	int(10)	(NULL)	YES	MUL	(NULL)		select,insert,update,references	
post_views	int(10)	(NULL)	YES		(NULL)		select,insert,update,references	

Indexes

Table	Non unique	Key name	Seq in index	Column name	Collation	Cardinality	Sub part	Packed	Null	Index type	Comment	Index comment
forum	0	PRIMARY	1	post_id	A	6	(NULL)	(NULL)		BTREE		
forum	1	member_id	1	member_id	A	6	(NULL)	(NULL)	YES	BTREE		

Foreign Key Relationships

FK Id	Reference Table	Source Column	Target Column	Extra Info
forum_ibfk_1	members	`member_id`	`mem_id`	

Figure 5.8 forum table

forum_cmnts

Fields

Field	Type	Collation	Null	Key	Default	Extra	Privileges	Comment
fc_id	int(10)	(NULL)	NO	PRI	(NULL)	auto_increment	select,insert,update,references	
post_id	int(10)	(NULL)	YES	MUL	(NULL)		select,insert,update,references	
mem_id	int(10)	(NULL)	YES	MUL	(NULL)		select,insert,update,references	
cmnt	varchar(500)	utf8_general_ci	YES		(NULL)		select,insert,update,references	
cmnt_likes	int(10)	(NULL)	YES		(NULL)		select,insert,update,references	
cmnt_dislike	int(10)	(NULL)	YES		(NULL)		select,insert,update,references	

Indexes

Table	Non unique	Key name	Seq in index	Column name	Collation	Cardinality	Sub part	Packed	Null	Index type	Comment	Index comment
forum_cmnts	0	PRIMARY	1	fc_id	A	4	(NULL)	(NULL)		BTREE		
forum_cmnts	1	post_id	1	post_id	A	2	(NULL)	(NULL)	YES	BTREE		
forum_cmnts	1	mem_id	1	mem_id	A	4	(NULL)	(NULL)	YES	BTREE		

Foreign Key Relationships

FK Id	Reference Table	Source Column	Target Column	Extra Info
forum_cmnts_ibfk_1	forum	`post_id`	`post_id`	
forum_cmnts_ibfk_2	members	`mem_id`	`mem_id`	

Figure 5.9 forum_cmnts table

expense

Fields

Field	Type	Collation	Null	Key	Default	Extra	Privileges	Comment
member_id	int(10)	(NULL)	YES	MUL	1		select,insert,update,references	
e_id	int(10)	(NULL)	NO	PRI	(NULL)	auto_increment	select,insert,update,references	
e_amount	int(20)	(NULL)	NO		(NULL)		select,insert,update,references	
e_type	varchar(20)	utf8_general_ci	YES		spend		select,insert,update,references	
e_receiver	varchar(20)	utf8_general_ci	YES		(NULL)		select,insert,update,references	
e_category	varchar(20)	utf8_general_ci	YES		(NULL)		select,insert,update,references	
e_date	varchar(40)	utf8_general_ci	YES		(NULL)		select,insert,update,references	
e_receipt	varchar(100)	utf8_general_ci	YES		(NULL)		select,insert,update,references	
e_notes	varchar(200)	utf8_general_ci	YES		(NULL)		select,insert,update,references	

Indexes

Table	Non unique	Key name	Seq in index	Column name	Collation	Cardinality	Sub part	Packed	Null	Index type	Comment	Index comment
expense	0	PRIMARY	1	e_id	A	2	(NULL)	(NULL)		BTREE		
expense	1	member_id	1	member_id	A	2	(NULL)	(NULL)	YES	BTREE		

Foreign Key Relationships

FK Id	Reference Table	Source Column	Target Column	Extra Info
expense_ibfk_1	members	`member_id`	`mem_id`	

Figure 5.10 expense table

5.1.2LOGICAL REPRESENTATION OF DATA

- **user_accounts table:** This table includes user account information including his full name, email, password, profile details, country and city etc.
- **admin_accounts table:** This table includes admin account information such as email, password etc .
- **countries table:** This table includes information about countries.
- **cities table:** This table includes information about cities and country they belonged to. It has country_id as foreign key for many to one relationship.
- **trips table:** This table includes information about user trips including destination, places, category, schedule, description etc.
- **trip_categories table:** This table includes information about trip categories.
- **trip_photos table:** This table includes path for storing photo and details about trip photos uploaded by users.
- **trip_ratings table:** This table includes information about ratings given by users to the trips.
- **Places table:** This table is to store information about places.

- **trips_places_asc table:** This is an association table to achieve many to many (M-M) relationship for trips and places.
- **trip_notes table:** This table is for storing notes by user.
- **posts table:** This table includes information about blog posts, their creation time, contents, votes and information about user who authored the post.
- **posts_places_asc table:** This is an association table to achieve many to many (M-M) relationship for posts and places.
- **comments table:** this table includes information about comments on the post.
- **flag_types table:** This table includes information about types of flags.
- **flagged_post table:** This table includes information about all flagged posts by users to claim inweb appropriate content.

5.2 SYSTEM PROCEDURAL DESIGN

5.2.1 DESIGN PSEUDO CODE OR ALGORITHM FOR METHOD OR OPERATION

User Side:

Step 1: User have to login into System using their credentials .

Step 2: After login, DashBoard is displayed

Step 3: User can use many features of system such as navigating through meeting, adding task, adding expenses of society.

Step 4: Web Web application can update their status automatically which will be seen by other residents..

Step 5: DashBoard contains navigation to main operations.

Step 6: Close the web application.

Admin Side:

Step 1: Enter the url to Open the system.

Step 2: Provide username and password.

Step 3: If username and password is correct then it will login successfully

Step 4: It shows the home page which shows the navigation to all admin activities.

Step 5: Admin can able to perform operation like insert, update and delete

Step 6: Logout

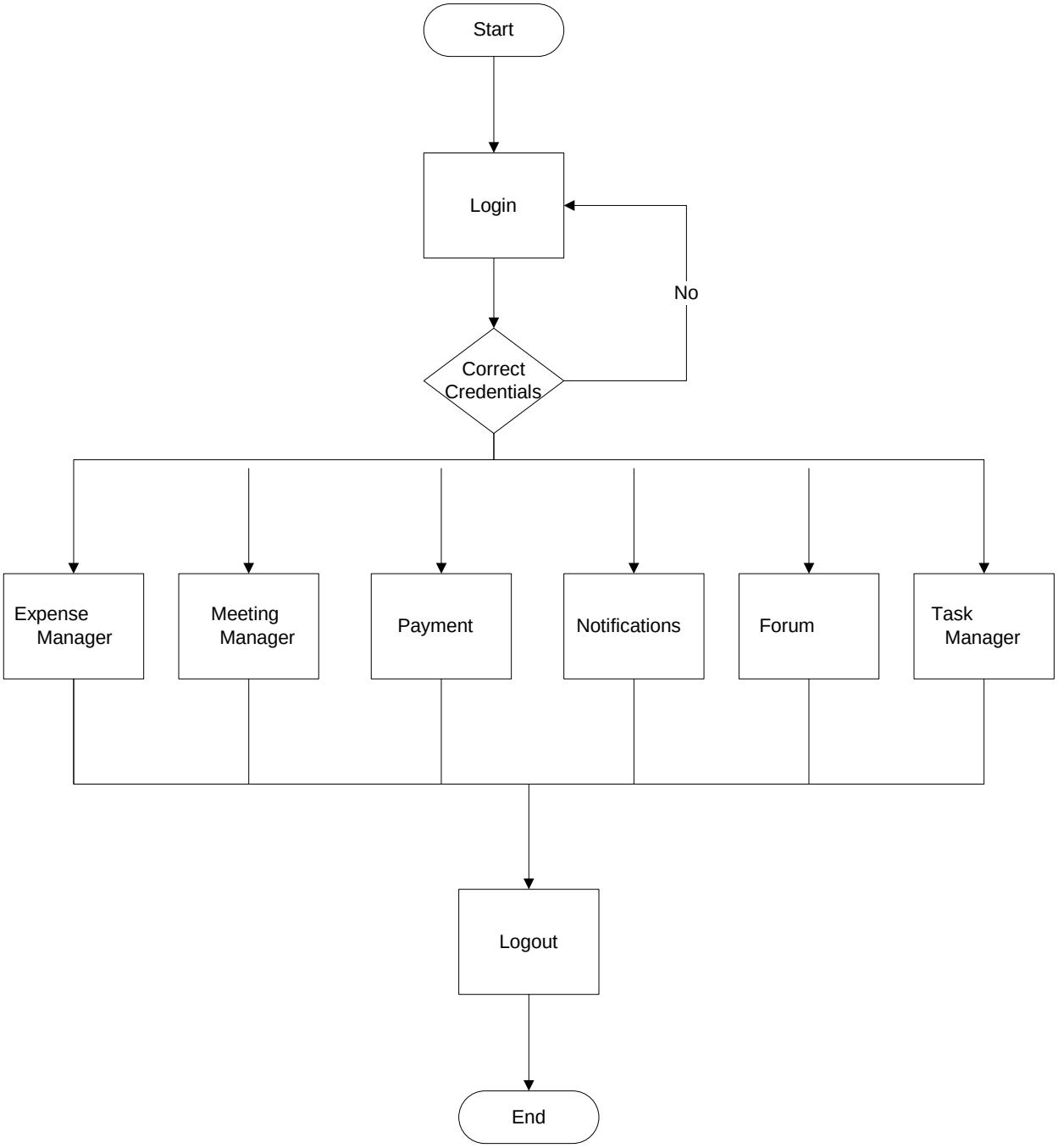


Figure 5.11 Flow Chart for User

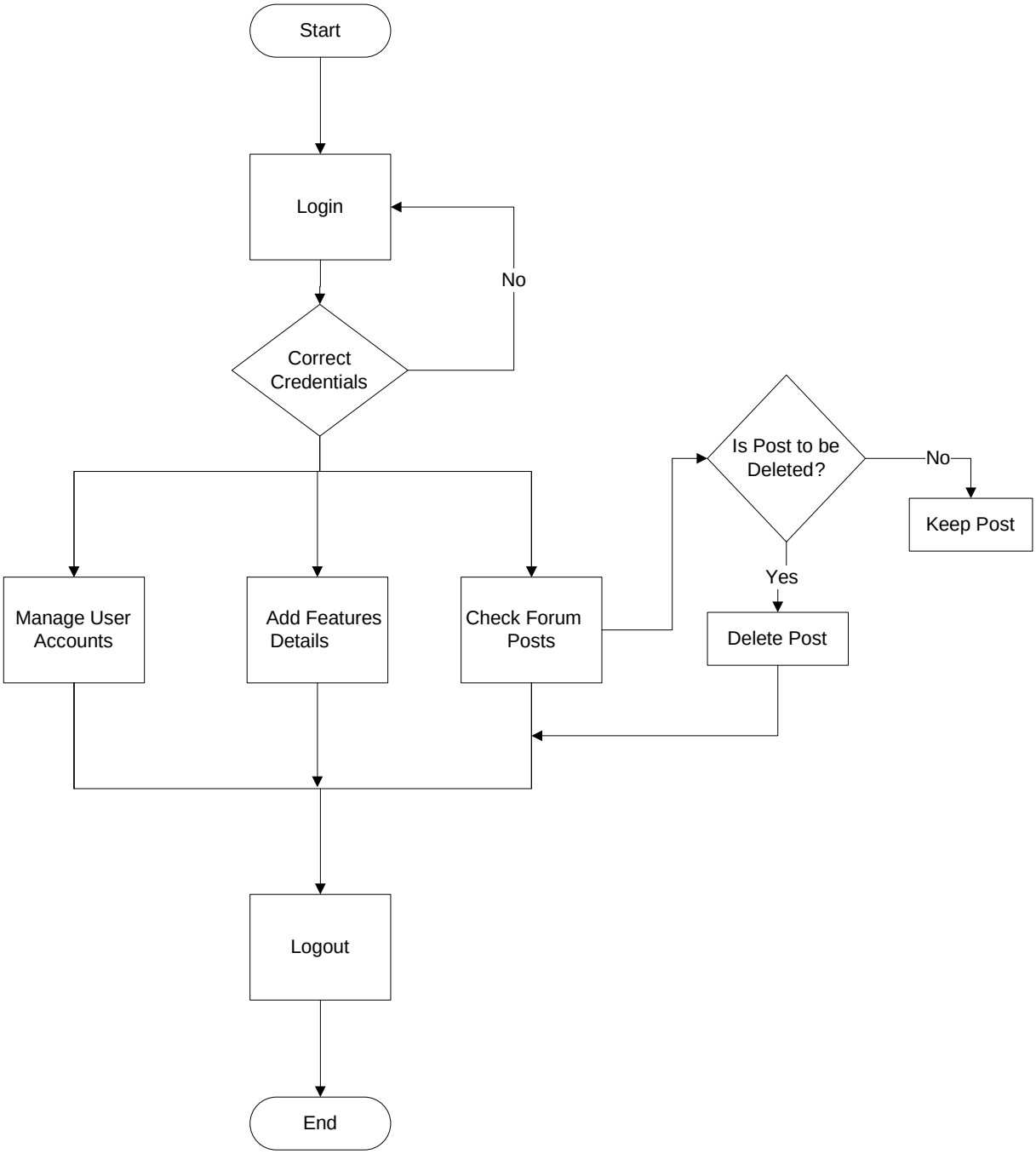


Figure 5.12 Flow Chart for Admin

5.3 INPUT/OUTPUT & INTERFACE DESIGN

5.3.1 SAMPLES OF FORMS, REPORTS & INTERFACE

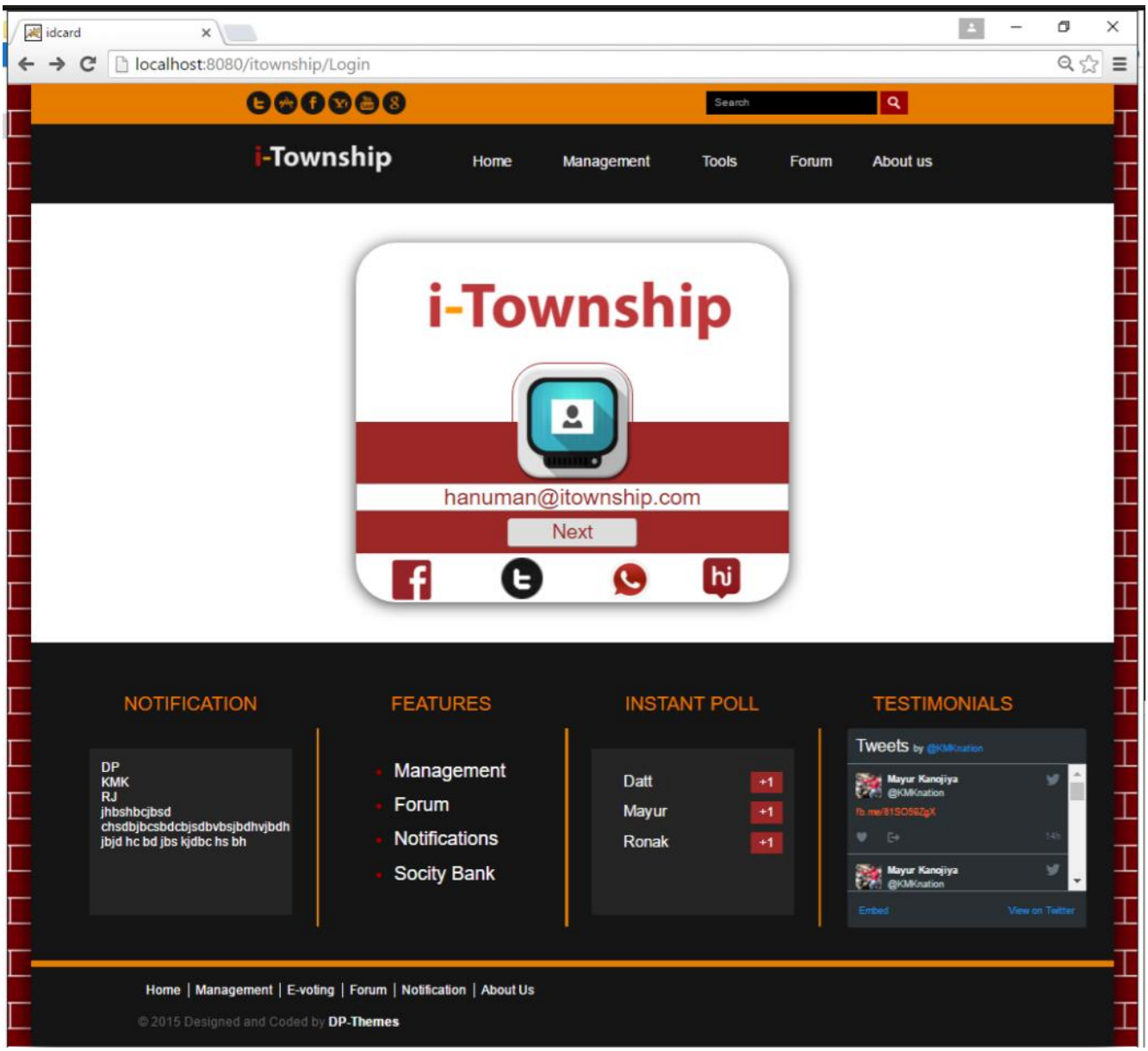


Figure 5.13 Sample Form for User Login

Figure 5.14 Sample Form for User Login

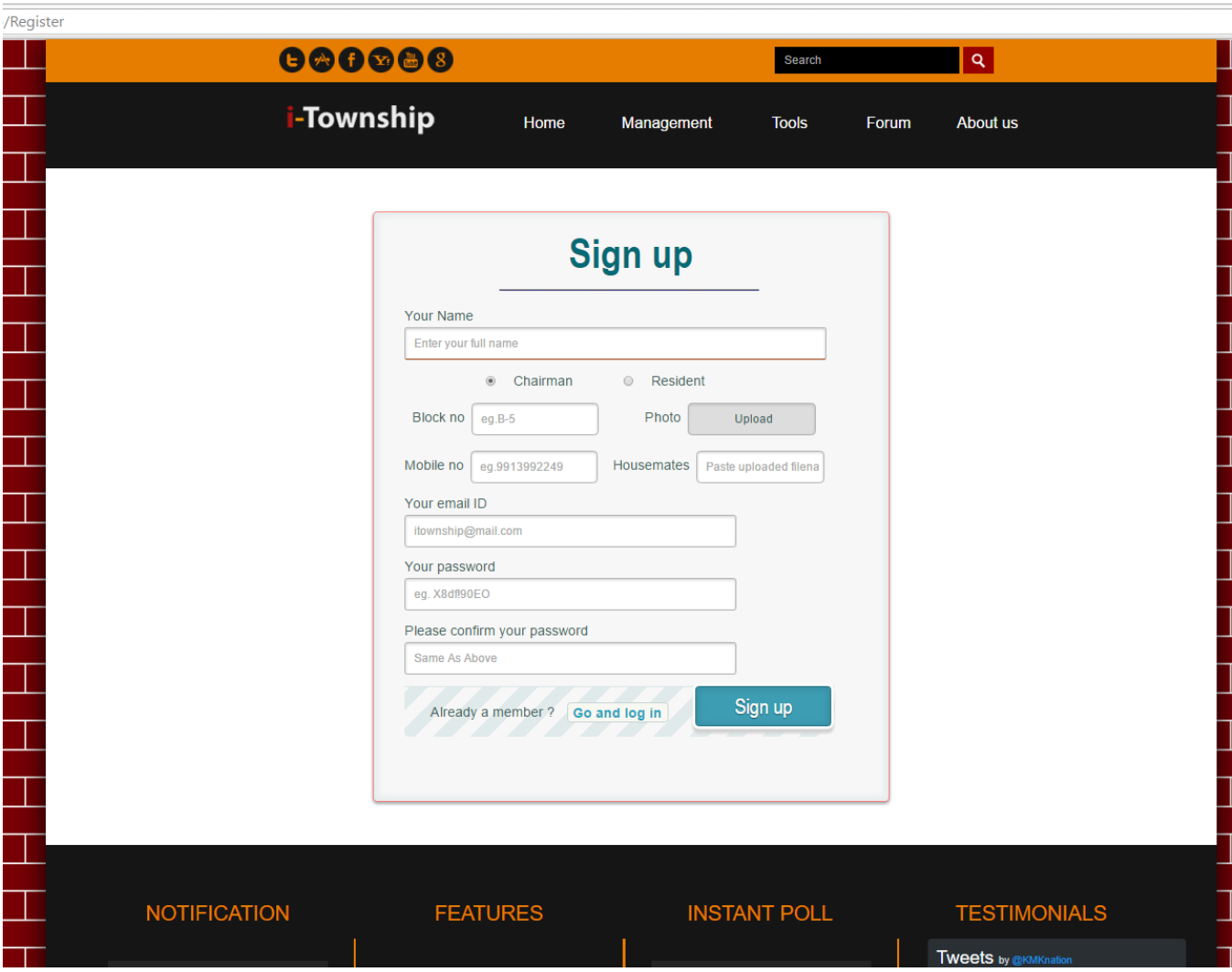


Figure 5.15 Sample Form for User Registration

CHAPTER

6

IMPLEMENTATION PLANNING AND DETAILS

CHAPTER:6 IMPLEMENTATION PLANNING AND DETAILS

6.1 IMPLEMENTATION ENVIRONMENT

- I-Township is web based portal. In this project, We are going to develop the java based software which involves management of a Society. In this portal, the chairman of society suggests the idea for completing such tasks. This task involves management of society from different areas of interest.

- By accessing this portal user can view whether the work of society has done or not, can complain, give suggestion, take part in E-voting, view a VR sitemap on the portal. The chairman maintains detail about all the Modules.

- It is very useful for the members of society. The portal provides the notifications on an android web application, gives attractive user interaction functionalities. So we can achieve more suggestions from user and task will be performed in proper manner.

❖ Hardware Requirement

- Pentium 4 processor equivalent
- Windows OS
- 1GB RAM
- 4GB Hard Disk

❖ **Software Requirements**

- Windows Operating System
- Eclipse with Android SDK API-19
- Web Services- As a Back End

6.2 MODULES SPECIFICATION

❖ **Admin**

- Calculate expense and verify transactions
- Update Forum post
- Add and Manage Functionalities
- Add Users
- Post Notifications

❖ **User**

- Edit profile
- Add Expense
- Insert, Update and delete posts
- View Expense Records
- Use features
- Make Transactions

6.3 SECURITY FEATURES

- Android is a modern mobile platform that was designed to be truly open. Securing an open platform requires robust security architecture and rigorous security programs. Android was designed with multi-layered security that provides the flexibility required for an open platform, while providing protection for all users of the platform.
- Android was designed to both reduce the probability of these attacks and greatly limit the impact of the attack in the event it was successful.
- Android seeks to be the most secure and usable operating system for mobile 1platforms by re-purposing traditional operating system security controls to:
  - Protect user data
  - Protect system resources (including the network)
  - Provide web application isolation
- To achieve these objectives, Android provides these key security features:
  - Robust security at the OS level through the Linux kernel
  - Mandatory web application sandbox for all web applications
  - Secure inter-process communication
  - Web application signing
  - Web application-defined and user-granted permissions

6.4 CODING STANDARDS

We follow standard Java coding conventions. We add a few rules

- **Java Language Rules➢Conventions:**

We follow standard Java coding conventions. We add a few Android specific rules.

- **Package and Import Statements :**

The first non-comment line of most Java source files is a package statement.

After that, import statements can follow. For example: packagejava.awt;import

java.awt.peer.CanvasPeer;

➢**Order Import Statements :**

The ordering of import statements is:

1. Android imports
2. Imports from third parties (com, junit, net, org)
3. java and javax

To exactly match the IDE settings, the imports should be:

- Alphabetical within each grouping, with capital letters before lower case letters (e.g. Z before a).
- There should be a blank line between each major grouping (android, com, junit, net, org, java, javax).

➢ **Fully Qualify Imports:**

When you want to use class Bar from package foo, there are two possible ways to import it:

import foo.*; import
foo.Bar;

Use the latter for importing all Android code. An explicit exception is made for java standard libraries (java.util.*, java.io.*, etc.) and unit test code (junit.framework.*).

➢ **Number per Line:**

One declaration per line is recommended since it encourages commenting. In other words,

int level; // indentation level int size; // size
of table

➢ **Class and Interface Declarations:**

When coding Java classes and interfaces, the following formatting rules should be followed:

- No space between a method name and the parenthesis ―(― starting its parameter list .Open brace ―{‖ web appears at the end of the same line as the declaration statement
- Closing brace ―}‖ starts a line by itself indented to match its corresponding opening statement, except when it is a null statement the ―}‖ should web appear immediately after the ―{― .
- Methods are separated by a blank line.

- **Naming Conventions:**

Naming conventions make programs more understandable by making them easier to read. They can also give information about the function of the identifier-for example, whether it's a constant, package, or class-which can be helpful in understanding the code.

- Use full English descriptors that accurately describe the variable/field/class/interface

  For example, use names like firstName, grandTotal, or CorporateCustomer.

- Use terminology web applicable to the domain

  If the users of the system refer to their clients as Customer, then use the term Customer for the class, not client.

- Use mixed case to make names readable.
- Use abbreviations sparingly, but if you do so then use then intelligently and document it .

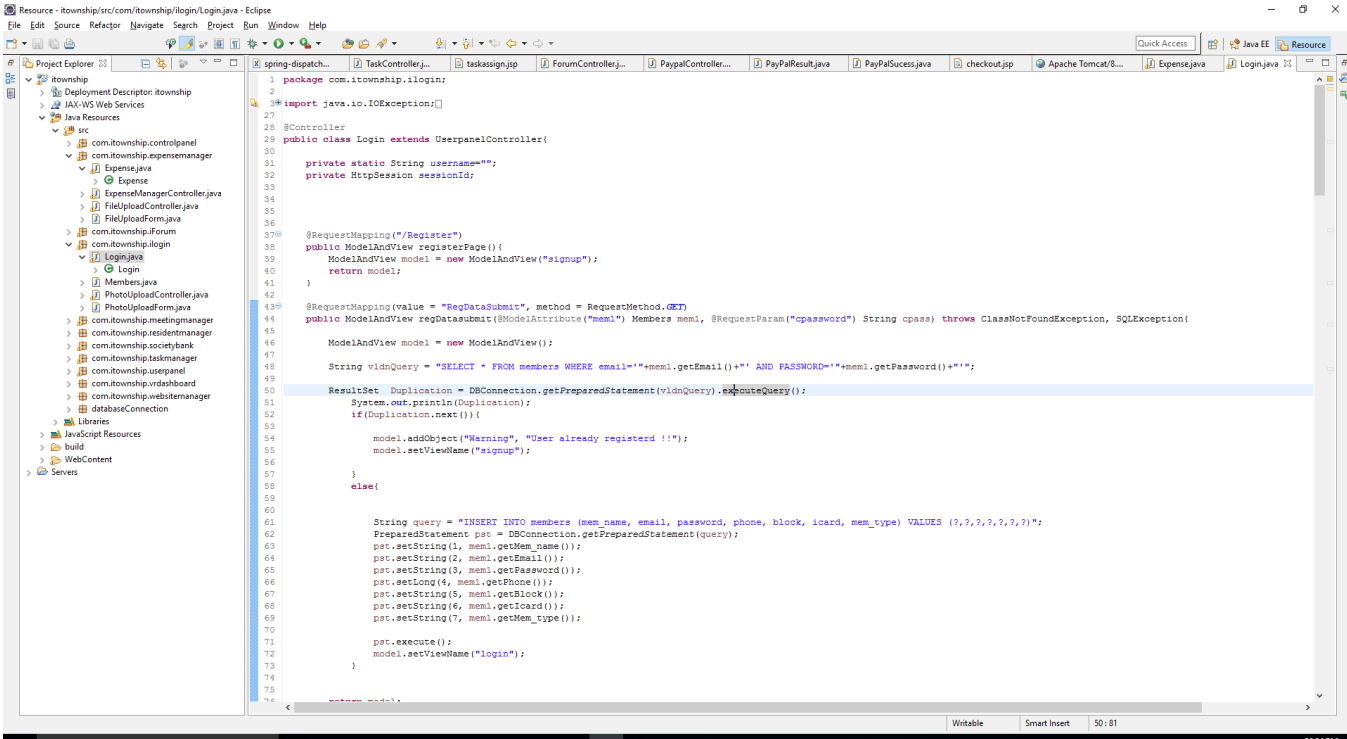
- **Don't Ignore Exceptions:**

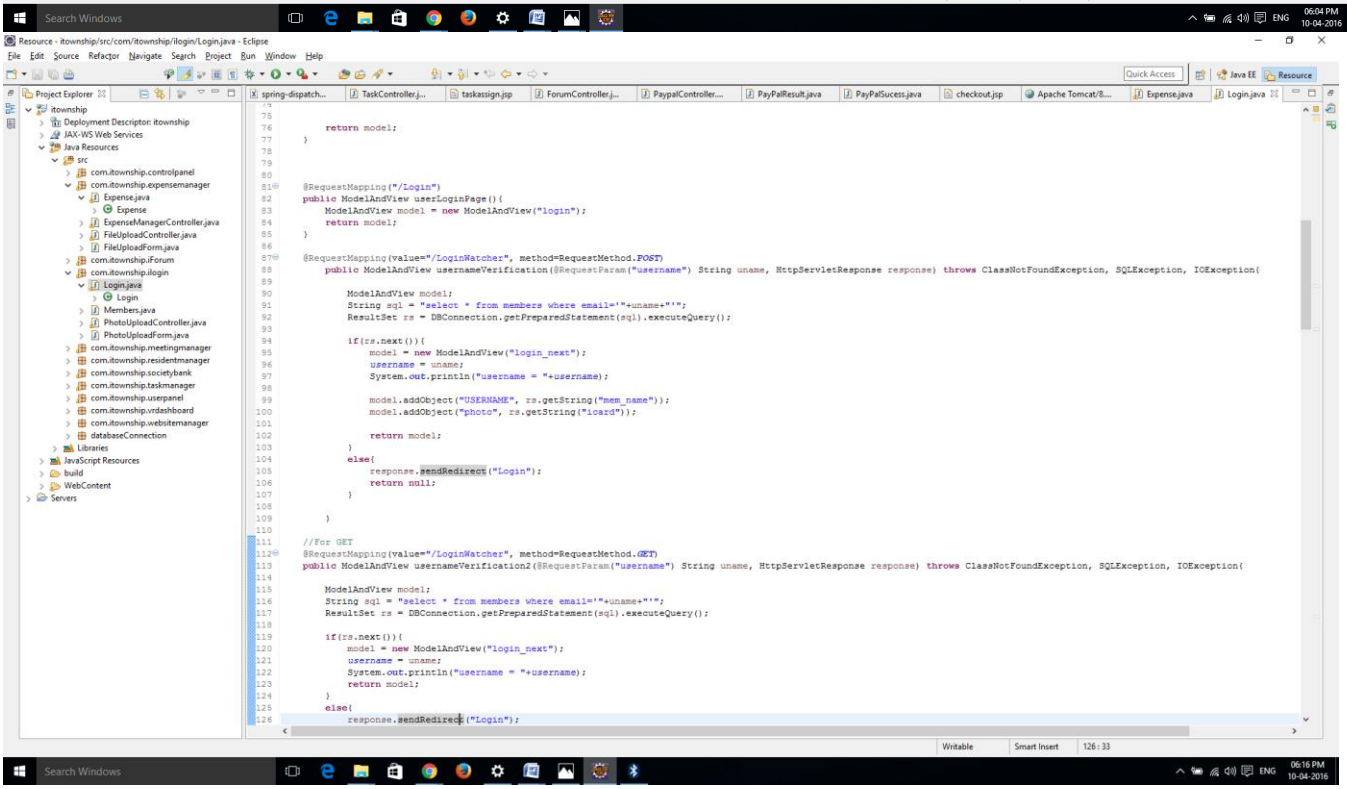
- Sometimes it is tempting to write code that completely ignores an exception like this:voidsetServerPort(String value)

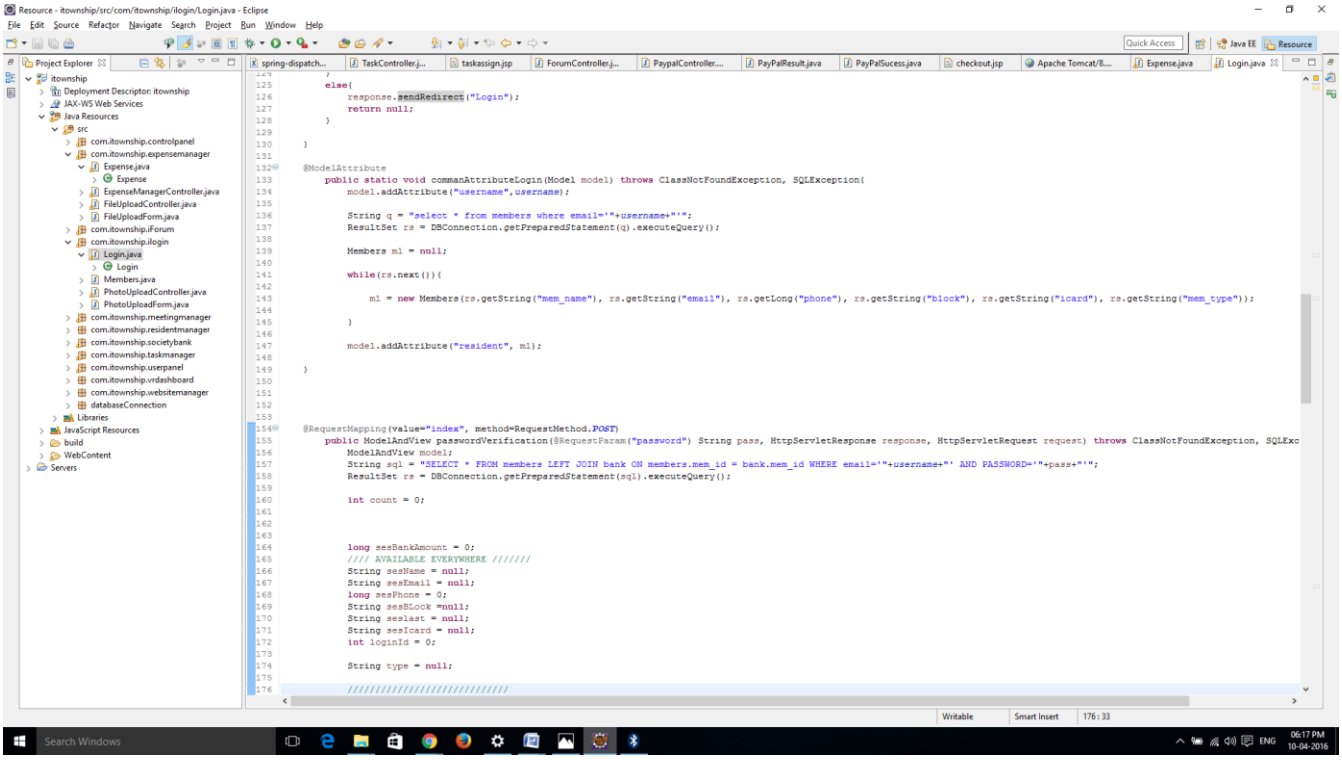
```
{
try
{
serverPort = Integer.parseInt(value);
} catch (NumberFormatException e) { }
}
```

- You must never do this. While you may think that your code will never encounter this error condition or that it is not important to handle it, ignoring exceptions like above creates mines in your code for someone else to trip over some day. You must handle every Exception in your code in some principled way. The specific handling varies depending on the case.

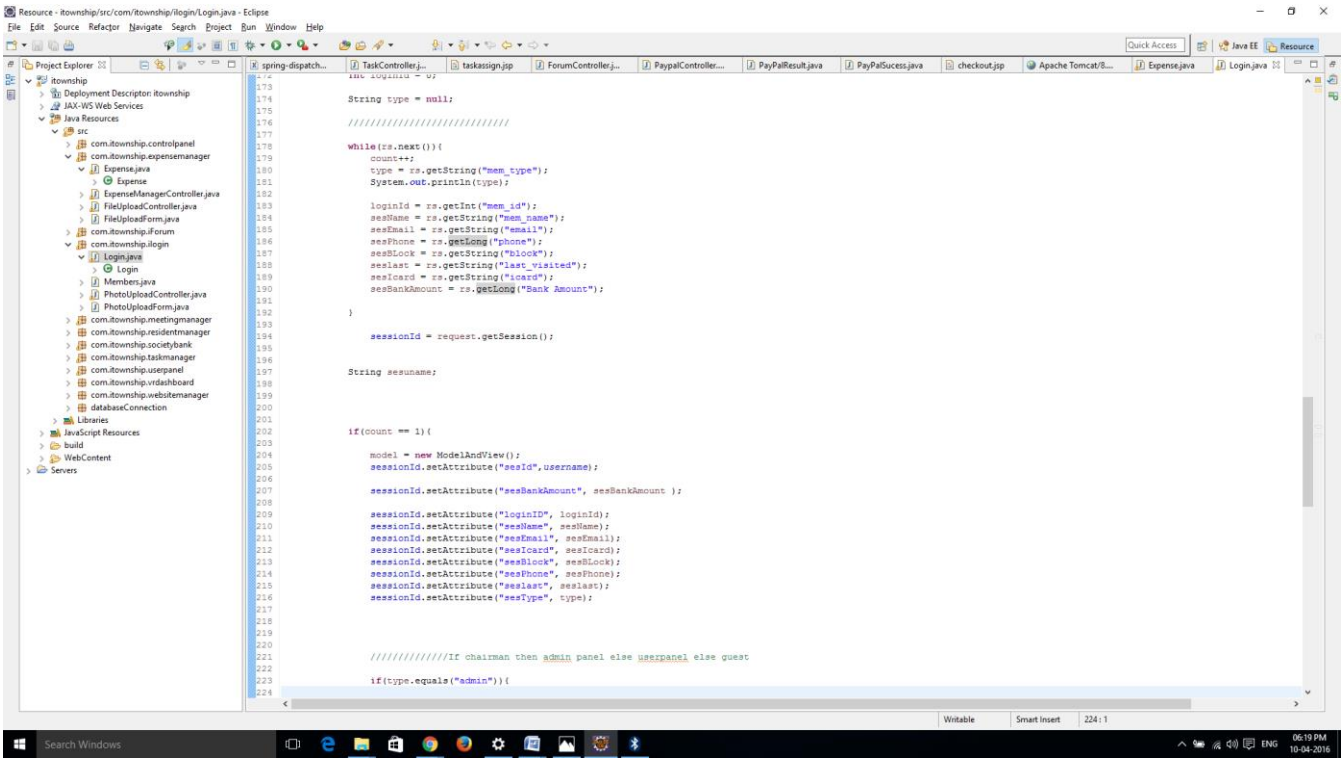
6.5 SAMPLE CODING


Resource - itownship/src/com/itownship/ilogin/Login.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
itownship
Deployment Descriptor: itownship
JAX-WS Web Services
Java Resources
src
com.itownship.controlpanel
com.itownship.expensemanager
Expense.java
Expense
ExpenseManagerController.java
FileUploadController.java
FileUploadForm.java
com.itownship.iForum
com.itownship.ilogin
Login.java
Login
Members.java
PhotoUploadController.java
PhotoUploadForm.java
com.itownship.meetingmanager
com.itownship.residentmanager
com.itownship.societybank
com.itownship.taskmanager
com.itownship.userpanel
com.itownship.vrdashboard
com.itownship.websitemanager
databaseConnection
Libraries
JavaScript Resources
build
WebContent
Servers
spring-dispatch...
TaskController.j...
taskassign.jsp
ForumController.j...
PaypalController....
PayPalResult.java
PayPalSucess.java
checkout.jsp
Apache Tomcat/8....
Expense.java
Login.java
Quick Access
Java EE
Resource
package com.itownship.ilogin;
import java.io.IOException;
@Controller
public class Login extends UserpanelController{
private static String username="";
private HttpSession sessionId;
@RequestMapping("/Register")
public ModelAndView registerPage(){
ModelAndView model = new ModelAndView("signup");
return model;
}
@RequestMapping(value = "RegDataSubmit", method = RequestMethod.GET)
public ModelAndView regDatasubmit(@ModelAttribute("mem1") Members mem1, @RequestParam("cpassword") String cpass) throws ClassNotFoundException, SQLException{
ModelAndView model = new ModelAndView();
String vldnQuery = "SELECT * FROM members WHERE email='"+mem1.getEmail()+"' AND PASSWORD='"+mem1.getPassword()+"'";
ResultSet Duplication = DBConnection.getPreparedStatement(vldnQuery).executeQuery();
System.out.println(Duplication);
if(Duplication.next()){
model.addObject("Warning", "User already registerd !!");
model.setViewName("signup");
}
else{
String query = "INSERT INTO members (mem_name, email, password, phone, block, icard, mem_type) VALUES (?,?,?,?,?,?,?)";
PreparedStatement pst = DBConnection.getPreparedStatement(query);
pst.setString(1, mem1.getMem_name());
pst.setString(2, mem1.getEmail());
pst.setString(3, mem1.getPassword());
pst.setLong(4, mem1.getPhone());
pst.setString(5, mem1.getBlock());
pst.setString(6, mem1.getIcard());
pst.setString(7, mem1.getMem_type());
pst.execute();
model.setViewName("login");
}
Writable
Smart Insert
50 : 81
Search Windows
ENG
06:04 PM
10-04-2016

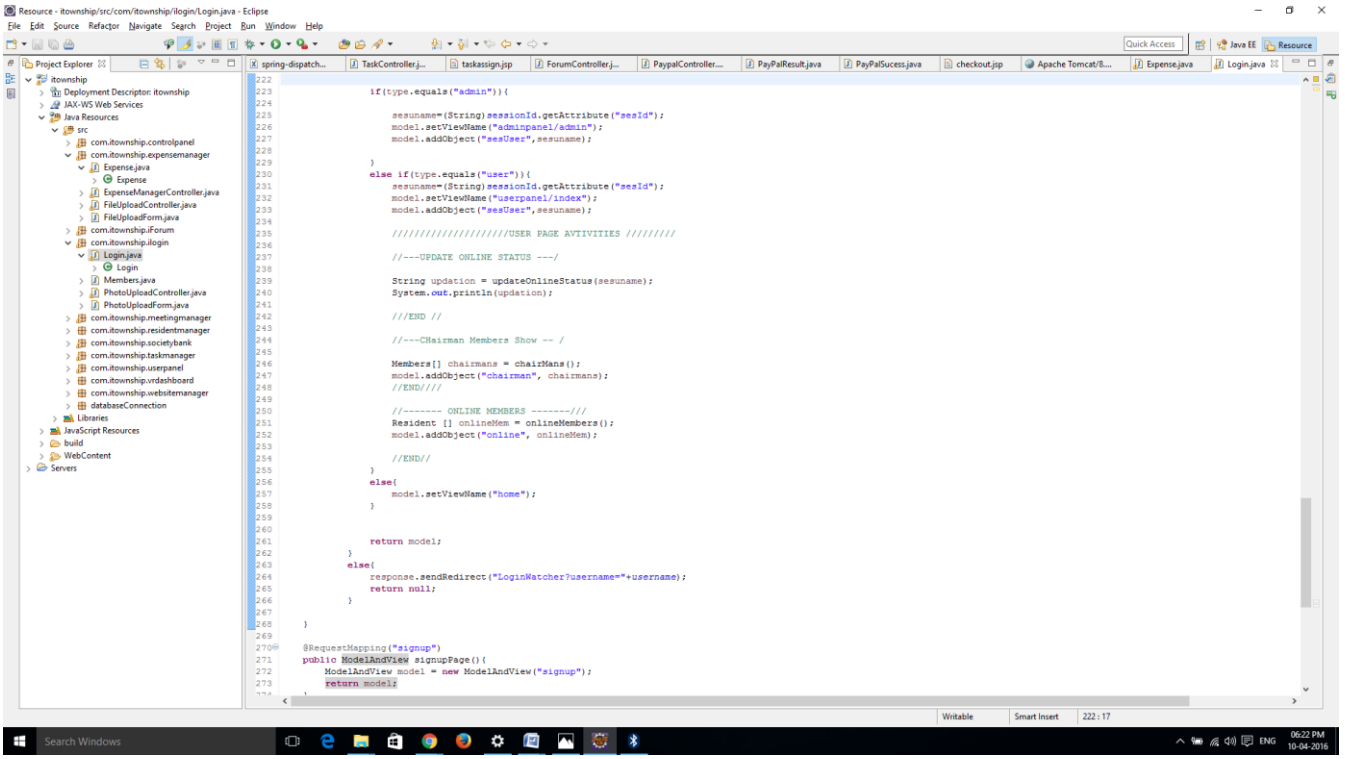

Resource - itownship/src/com/itownship/ilogin/Login.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
itownship
Deployment Descriptor: itownship
JAX-WS Web Services
Java Resources
src
com.itownship.controlpanel
com.itownship.expensemanager
Expense.java
Expense
ExpenseManagerController.java
FileUploadController.java
FileUploadForm.java
com.itownship.iForum
com.itownship.ilogin
Login.java
Login
Members.java
PhotoUploadController.java
PhotoUploadForm.java
com.itownship.meetingmanager
com.itownship.residentmanager
com.itownship.societybank
com.itownship.taskmanager
com.itownship.userpanel
com.itownship.vrdashboard
com.itownship.websitemanager
databaseConnection
Libraries
JavaScript Resources
build
WebContent
Servers
spring-dispatch...
TaskController.j...
taskassign.jsp
ForumController.j...
PaypalController....
PayPalResult.java
PayPalSucess.java
checkout.jsp
Apache Tomcat/8....
Expense.java
Login.java
Quick Access
Java EE
Resource
return model;
}
@RequestMapping("/Login")
public ModelAndView userLoginPage(){
ModelAndView model = new ModelAndView("login");
return model;
}
@RequestMapping(value="/LoginWatcher", method=RequestMethod.POST)
public ModelAndView usernameVerification(@RequestParam("username") String uname, HttpServletResponse response) throws ClassNotFoundException, SQLException, IOException{
ModelAndView model;
String sql = "select * from members where email='"+uname+"'";
ResultSet rs = DBConnection.getPreparedStatement(sql).executeQuery();
if(rs.next()){
model = new ModelAndView("login_next");
username = uname;
System.out.println("username = "+username);
model.addObject("USERNAME", rs.getString("mem_name"));
model.addObject("photo", rs.getString("icard"));
return model;
}
else{
response.sendRedirect("Login");
return null;
}
}
//For GET
@RequestMapping(value="/LoginWatcher", method=RequestMethod.GET)
public ModelAndView usernameVerification2(@RequestParam("username") String uname, HttpServletResponse response) throws ClassNotFoundException, SQLException, IOException{
ModelAndView model;
String sql = "select * from members where email='"+uname+"'";
ResultSet rs = DBConnection.getPreparedStatement(sql).executeQuery();
if(rs.next()){
model = new ModelAndView("login_next");
username = uname;
System.out.println("username = "+username);
return model;
}
else{
response.sendRedirect("Login");
Writable
Smart Insert
126 : 33
Search Windows
ENG
06:16 PM
10-04-2016



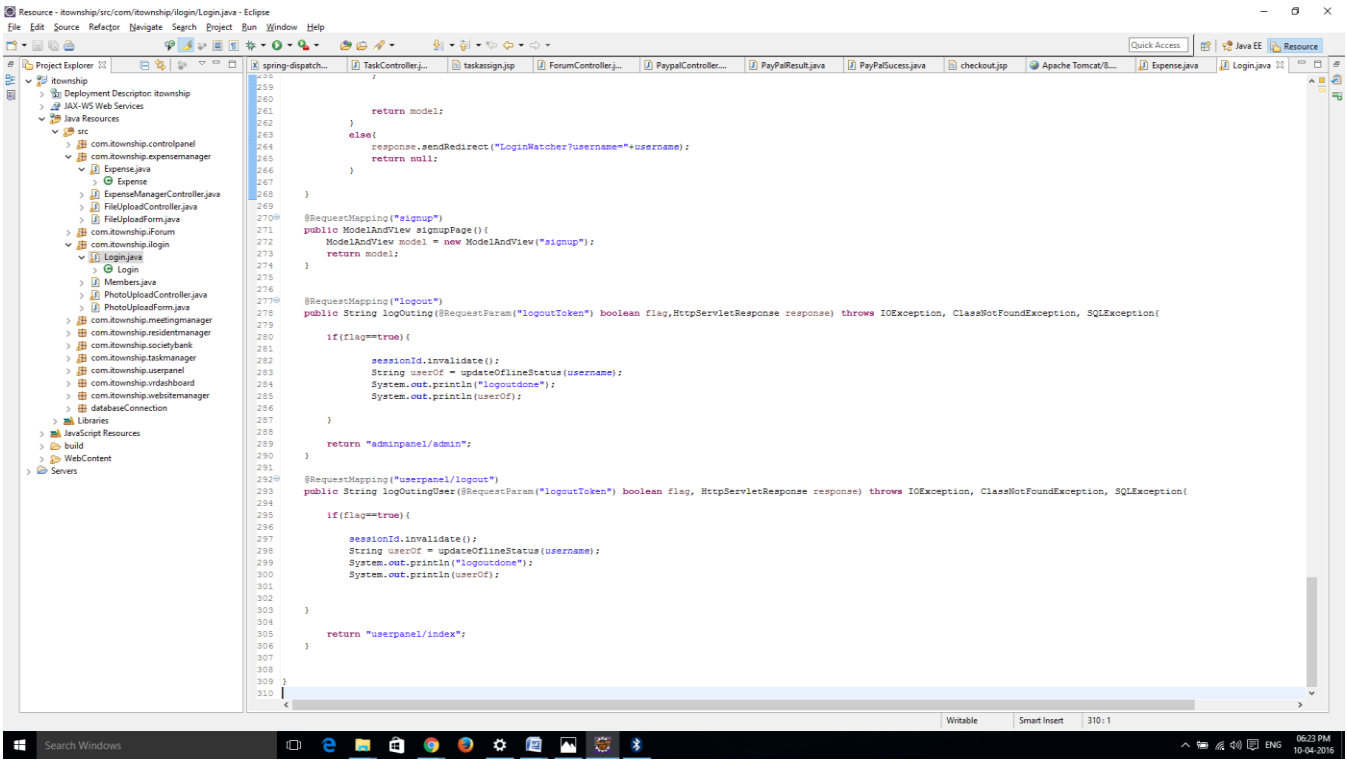
```
        else{
            response.sendRedirect("Login");
            return null;
        }

    }
    @ModelAttribute
        public static void commanAttributeLogin(Model model) throws ClassNotFoundException, SQLException{
            model.addAttribute("username",username);

            String q = "select * from members where email='"+username+"'";
            ResultSet rs = DBConnection.getPreparedStatement(q).executeQuery();

            Members ml = null;

            while(rs.next()){

                ml = new Members(rs.getString("mem_name"), rs.getString("email"), rs.getLong("phone"), rs.getString("block"), rs.getString("icard"), rs.getString("mem_type"));

            }

            model.addAttribute("resident", ml);

    }



    @RequestMapping(value="index", method=RequestMethod.POST)
        public ModelAndView passwordVerification(@RequestParam("password") String pass, HttpServletResponse response, HttpServletRequest request) throws ClassNotFoundException, SQLExc
            ModelAndView model;
            String sql = "SELECT * FROM members LEFT JOIN bank ON members.mem_id = bank.mem_id WHERE email='"+username+"' AND PASSWORD='"+pass+"'";
            ResultSet rs = DBConnection.getPreparedStatement(sql).executeQuery();

            int count = 0;



            long sesBankAmount = 0;
            //// AVAILABLE EVERYWHERE ///////
            String sesName = null;
            String sesEmail = null;
            long sesPhone = 0;
            String sesBLock =null;
            String seslast = null;
            String sesIcard = null;
            int loginId = 0;

            String type = null;

            ////////////////////////////
```



```
            String type = null;

            ////////////////////////////

            while(rs.next()){
                count++;
                type = rs.getString("mem_type");
                System.out.println(type);

                loginId = rs.getInt("mem_id");
                sesName = rs.getString("mem_name");
                sesEmail = rs.getString("email");
                sesPhone = rs.getLong("phone");
                sesBLock = rs.getString("block");
                seslast = rs.getString("last_visited");
                sesIcard = rs.getString("icard");
                sesBankAmount = rs.getLong("Bank Amount");

            }

                sessionId = request.getSession();


            String sesuname;



            if(count == 1){

                model = new ModelAndView();
                sessionId.setAttribute("sesId",username);

                sessionId.setAttribute("sesBankAmount", sesBankAmount );

                sessionId.setAttribute("loginID", loginId);
                sessionId.setAttribute("sesName", sesName);
                sessionId.setAttribute("sesEmail", sesEmail);
                sessionId.setAttribute("sesIcard", sesIcard);
                sessionId.setAttribute("sesBlock", sesBLock);
                sessionId.setAttribute("sesPhone", sesPhone);
                sessionId.setAttribute("seslast", seslast);
                sessionId.setAttribute("sesType", type);



                //////////////If chairman then admin panel else userpanel else guest

                if(type.equals("admin")){
```



```
if(type.equals("admin")){

    sesuname=(String)sessionId.getAttribute("sesId");
    model.setViewName("adminpanel/admin");
    model.addObject("sesUser",sesuname);

}
else if(type.equals("user")){
    sesuname=(String)sessionId.getAttribute("sesId");
    model.setViewName("userpanel/index");
    model.addObject("sesUser",sesuname);

    ////////////////////USER PAGE AVTIVITIES ////////

    //---UPDATE ONLINE STATUS ---/

    String updation = updateOnlineStatus(sesuname);
    System.out.println(updation);

    ///END //

    //---CHairman Members Show -- /

    Members[] chairmans = chairMans();
    model.addObject("chairman", chairmans);
    //END///

    //------- ONLINE MEMBERS -------///
    Resident [] onlineMem = onlineMembers();
    model.addObject("online", onlineMem);

    //END//
}
else{
    model.setViewName("home");
}


    return model;
}
else{
    response.sendRedirect("LoginWatcher?username="+username);
    return null;
}

}

@RequestMapping("signup")
public ModelAndView signupPage(){
    ModelAndView model = new ModelAndView("signup");
    return model;
}
```



```
            }

            return model;
        }
        else{
            response.sendRedirect("LoginWatcher?username="+username);
            return null;
        }

}

@RequestMapping("signup")
public ModelAndView signupPage(){
    ModelAndView model = new ModelAndView("signup");
    return model;
}


@RequestMapping("logout")
public String logOuting(@RequestParam("logoutToken") boolean flag,HttpServletResponse response) throws IOException, ClassNotFoundException, SQLException{

    if(flag==true){

        sessionId.invalidate();
        String userOf = updateOflineStatus(username);
        System.out.println("logoutdone");
        System.out.println(userOf);

    }

    return "adminpanel/admin";
}

@RequestMapping("userpanel/logout")
public String logOutingUser(@RequestParam("logoutToken") boolean flag, HttpServletResponse response) throws IOException, ClassNotFoundException, SQLException{

    if(flag==true){

        sessionId.invalidate();
        String userOf = updateOflineStatus(username);
        System.out.println("logoutdone");
        System.out.println(userOf);


}

    return "userpanel/index";
}


}
```

CHAPTER

7

TESTING

7.1 TESTING PLAN

7.2 TESTING STRATAGY

7.3 TESTING METHODS

7.4 TEST CASES

CHAPTER:7 TESTING

7.1 WEB WEB APPLICATION TESTING

Web web application testing, a software testing technique exclusively adopted to test the web applications that are hosted on web in which the web application interfaces and other functionalities are tested.

Web Web application Testing - Techniques:

1. **Functionality Testing** - The below are some of the checks that are performed but not limited to the below list:

- Verify there is no dead page or invalid redirects.
- First check all the validations on each field.
- Wrong inputs to perform negative testing.
- Verify the workflow of the system.
- Verify the data integrity.

2. **Usability testing -** To verify how the web application is easy to use with.

- Test the navigation and controls.
- Content checking.
- Check for user intuition.

3. **Interface testing -** Performed to verify the interface and the dataflow from one system to other.

4. **Compatibility testing-** Compatibility testing is performed based on the context of the web application.

- Browser compatibility
- Operating system compatibility
- Compatible to various devices like notebook, mobile, etc.

5. **Performance testing -** Performed to verify the server response time and throughput under various load conditions.

- **Load testing -** It is the simplest form of testing conducted to understand the behaviour of the system under a specific load. Load testing will result in measuring important business critical transactions and load on the database, web application server, etc. are also monitored.
- **Stress testing -** It is performed to find the upper limit capacity of the system and also to determine how the system performs if the current load goes well above the expected maximum.
- **Soak testing -** Soak Testing also known as endurance testing, is performed to determine the system parameters under continuous expected load. During soak tests the parameters such as memory utilization is monitored to detect memory leaks or other performance issues. The main aim is to discover the system's performance under sustained use.
- **Spike testing -** Spike testing is performed by increasing the number of users suddenly by a very large amount and measuring the performance of the system. The main aim is to determine whether the system will be able to sustain the work load.

6. **Security testing -** Performed to verify if the web application is secured on web as data theft and unauthorized access are more common issues and below are some of the techniques to verify the security level of the system.

- Injection
- Broken Authentication and Session Management
- Cross-Site Scripting (XSS)
- Insecure Direct Object References
- Security Misconfiguration
- Sensitive Data Exposure
- Missing Function Level Access Control
- Cross-Site Request Forgery (CSRF)
- Using Components with Known Vulnerabilities
- Unvalidated Redirects and Forwards

7.2 TESTING STRATAGY

Different levels of testing are used in the test process; each level of testing aims to test different aspects of the system.

- Understand the business workflow
- Develop test cases using various techniques (use case, decision table, etc.).
- Verify the flow with various user types (viz - Admin, Update user, View).
- Perform positive and negative tests.
- Compare the expected and actual results and log defects.
- Fix defects and deploy.

7.3 TESTING METHODS

- **White Box Testing**
  - White box testing is the detailed investigation of internal logic and structure of the code.
  - White box testing is also called glass testing or open box testing.
  - In order to perform white box testing on an web application, the tester needs to possess knowledge of the internal working of the code
  - The tester needs to have a look inside the source code and find out which unit/chunk of the code is behaving inweb appropriately.

- **Black Box Testing**
  - The technique of testing without having any knowledge of the interior workings of the web application is Black Box testing.
  - The tester is oblivious to the system architecture and does not have access to the source code.
  - Typically, when performing a black box test, a tester will interact with the system's user interface by providing inputs and examining outputs without knowing how and where the inputs are worked upon.
- **Unit Test**

- A Unit test tests only the functionality of a certain component.
- Let's, for example, assume a button in an Android activity is used to start another activity. A unit test would determine if the corresponding intent was issued, not if the second activity was started. A functional test would also check if the activity was correctly started.
- Android (up to Android 2.3 Gingerbread) uses JUnit 3. This version doesn't use annotations and uses introspection to detect the tests.

➢ **Integration Test**

- Integration tests are designed to test the way individual components work jointly. Modules that have been unit tested independently are now combined together to test the integration.
- Usually Android Activities require some integration to the system infrastructure to be able to run. They need the Activity lifecycle provided by the Activity Manager, and access to resources, file system, and databases.

➢ **Interface Testing**

In the system, standards tests for GUIs have been performed, which are as follows.

- Testing the screen control for its position and size.
- The position and related labels for all controls were checked.
- Name of the form in system is given web appropriately.
- All menu functions and sub functions were verified for correctness.

Each menu functions were tested, whether it invokes the corresponding functionality properly.

7.4 TEST CASES

➢ The purpose of a test case is to describe how you intend to empirically verify that the software being developed conforms to the specifications. In other words, you need to be able to show that it can correctly carry out its intended functions. The test case

should be written with enough clarity and detail that it could be given to an independent tester and have the tests properly carried out.

- **Test Case Description:**

  A test case contains all the information necessary to verify some particular functionality of the software:

- **Purpose:**

  Describe the features of the software to be tested, and the particular behavior being verified by this test. Requirement Traceability: A cross reference to the numbers of the requirements (in the system specification) which are being verified in this test

- **Setup:**

  Describe all the steps necessary to setup the software environment necessary to carry out the test.

- **Test Data and get expected output:**

  Write the actual input data to be provided and the expected output for your actual working product. You must provide the actual input data values, not just a description. Often the test data can be shown in tabular form, with a column of input items and the corresponding column of expected outputs.

1) Test case For Log In

Project: - I-Township

Objective: - To check whether user name & Password valid or invalid.

Page: `-Sign Up

Data:

Sr.no.	Steps	Expected Data	Status
1	Enter user Email , Password then password and press next button	Should navigate user to Dashboard.	Pass
2	Enter Email only and press Sign In button.	Should display page password page	Pass
3	Enter wrong Password and press Sign In button	Should still on the same page which asking for correct password.	Pass

4	Enter blank Email and blank Password and press Sign In button.	Should display message ―Email is required!‖ & ― Password is required!‖	Invalid
5	Enter wrong Email and Password	Should display a message ―Invalid Email or Password‖	Invalid

Table 7-1 Test case For Sign In

2) Test case For Sing Up:

Project: - I-Township.

Objective: - To check whether entered details are right or wrong.

Page: `-Sign Up

Test Data:

Sr.no.	Steps	Expected Data	Status
1	Enter Email , Password , Confirm Password and press Sign Up button	Should navigate user to Dashboard.	Pass
2	Enter Email and press Sign Up button.	Should display message ―Enter password & confirm password ―.	Invalid
3	Enter distinct data in password & confirm password.	Should display message ―password mismatch ! ―	Invalid
4	Enter Email in wrong format .	Should display message ―Enter email in correct format‖	Invalid

Table 7-2 Test case For Sign Up

8

SCREEN SHOTS

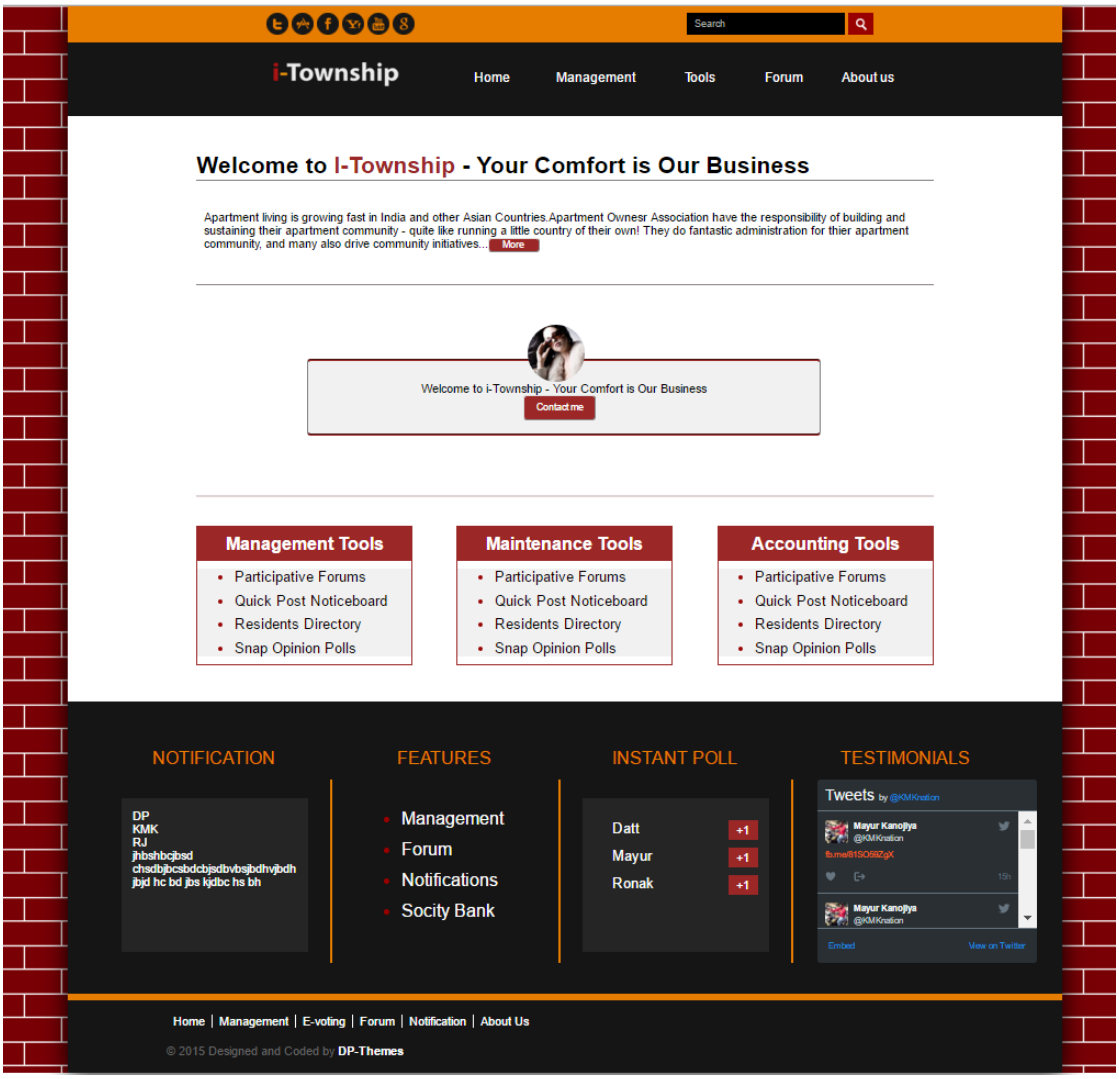


Figure 8.1 Home Screen

Figure 8.2 About Us

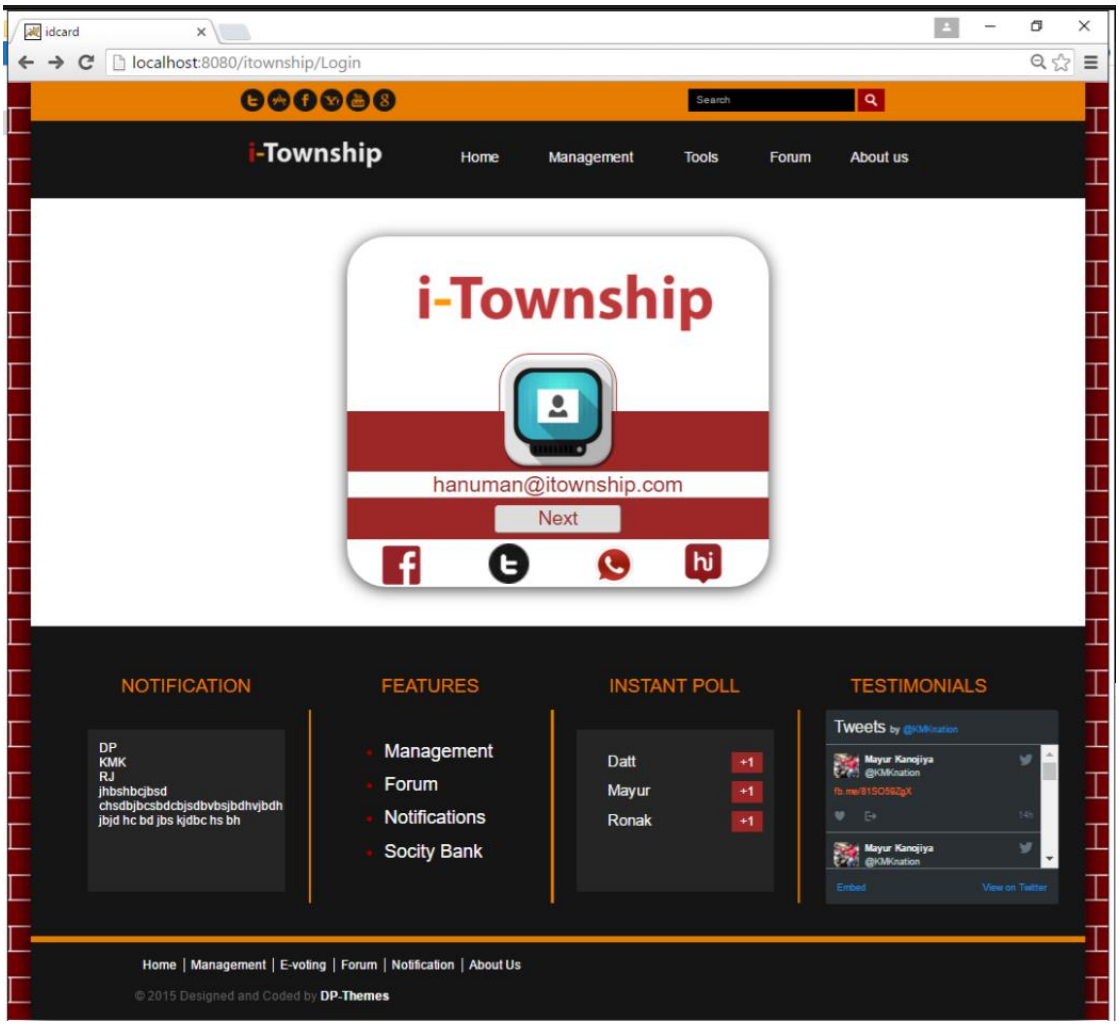


Figure 8.3 Login

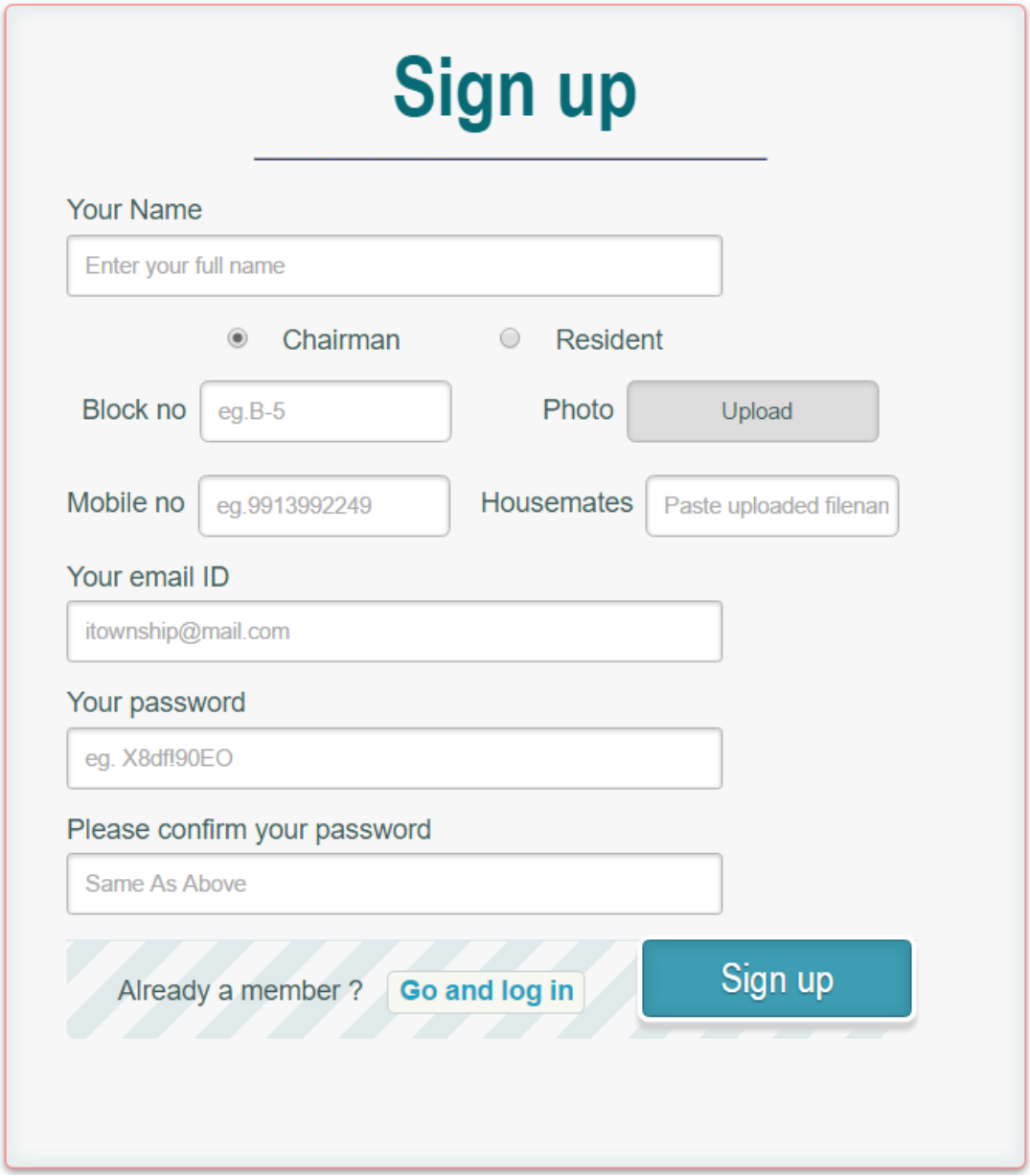


Figure 8.4 Sign Up

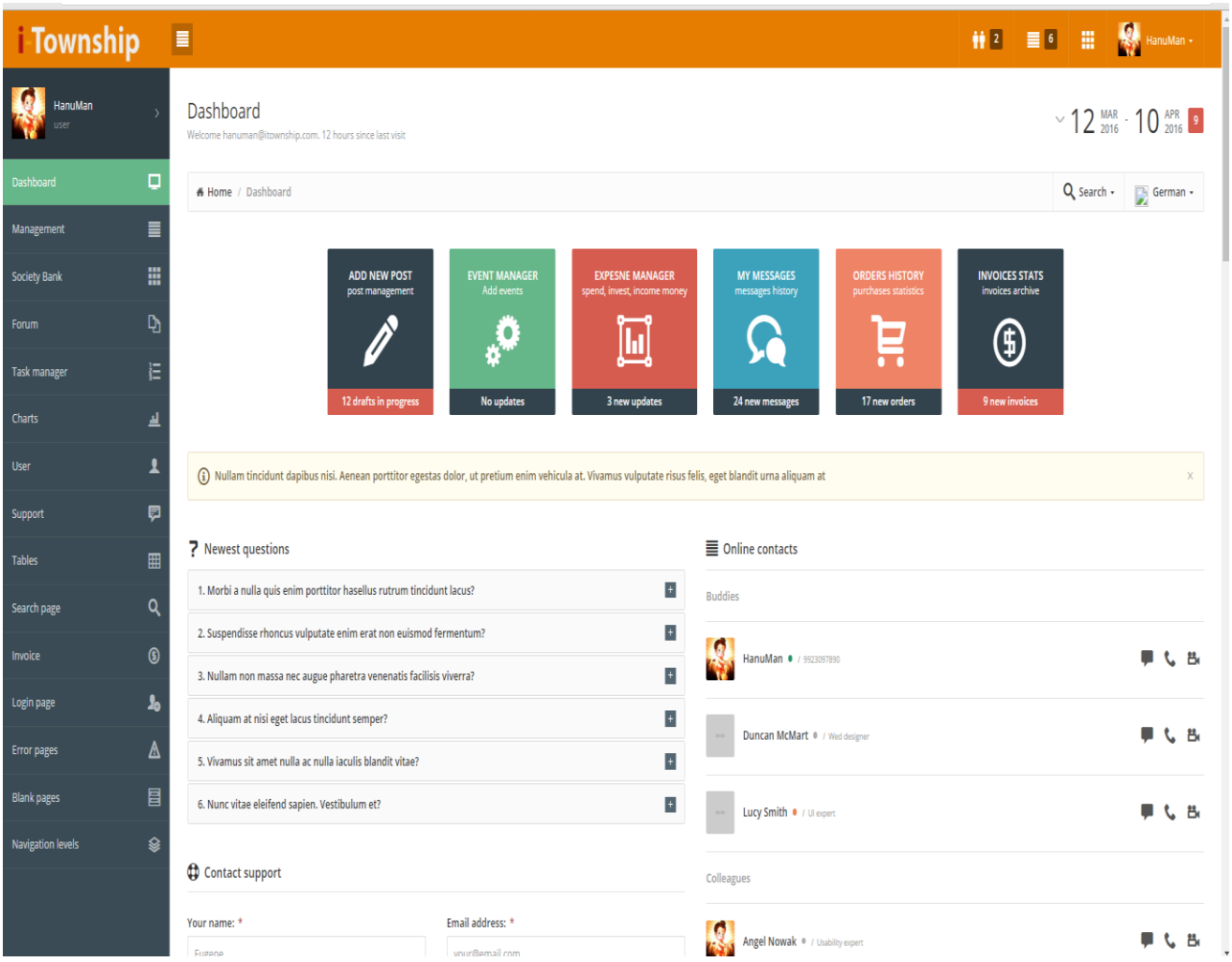


Figure 8.5 Successfully Login

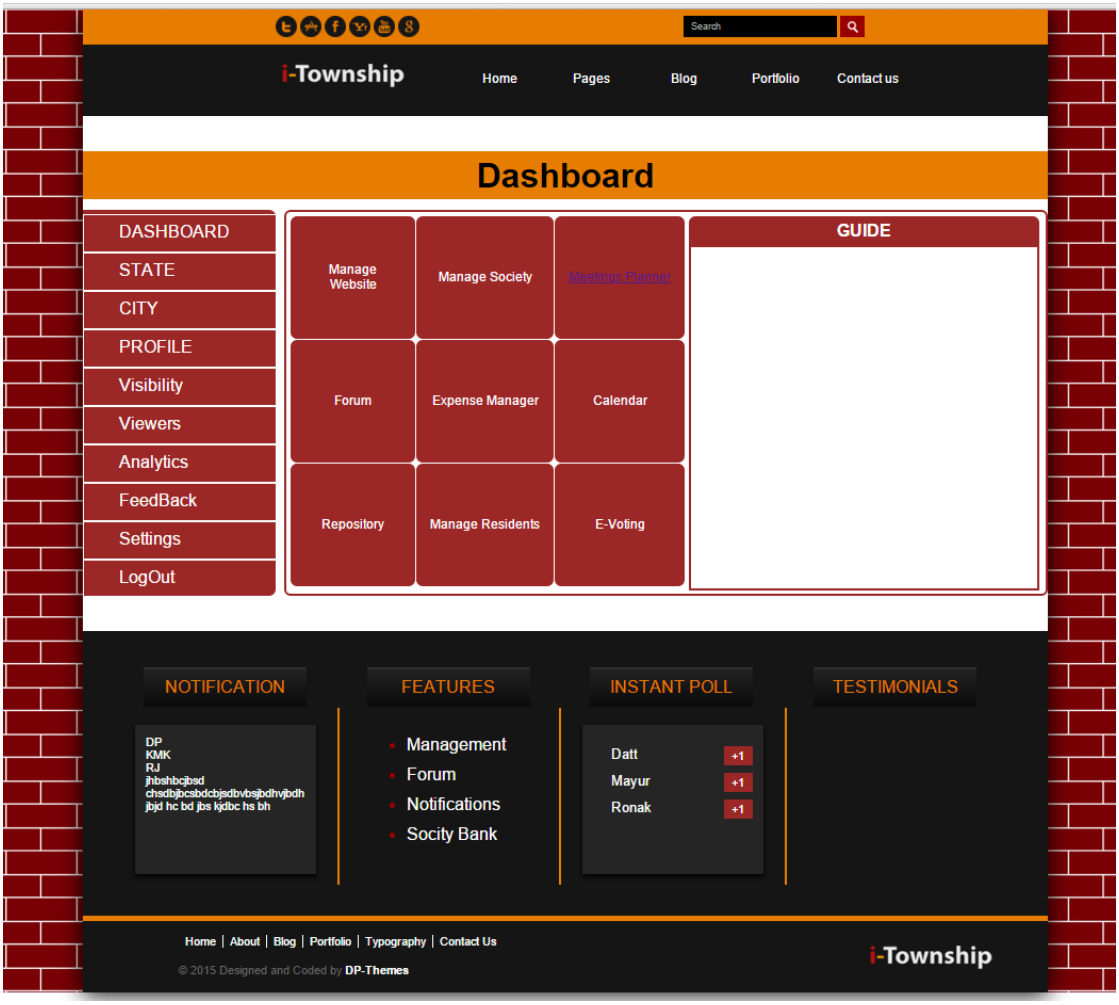


Figure 8.6 Dash Board

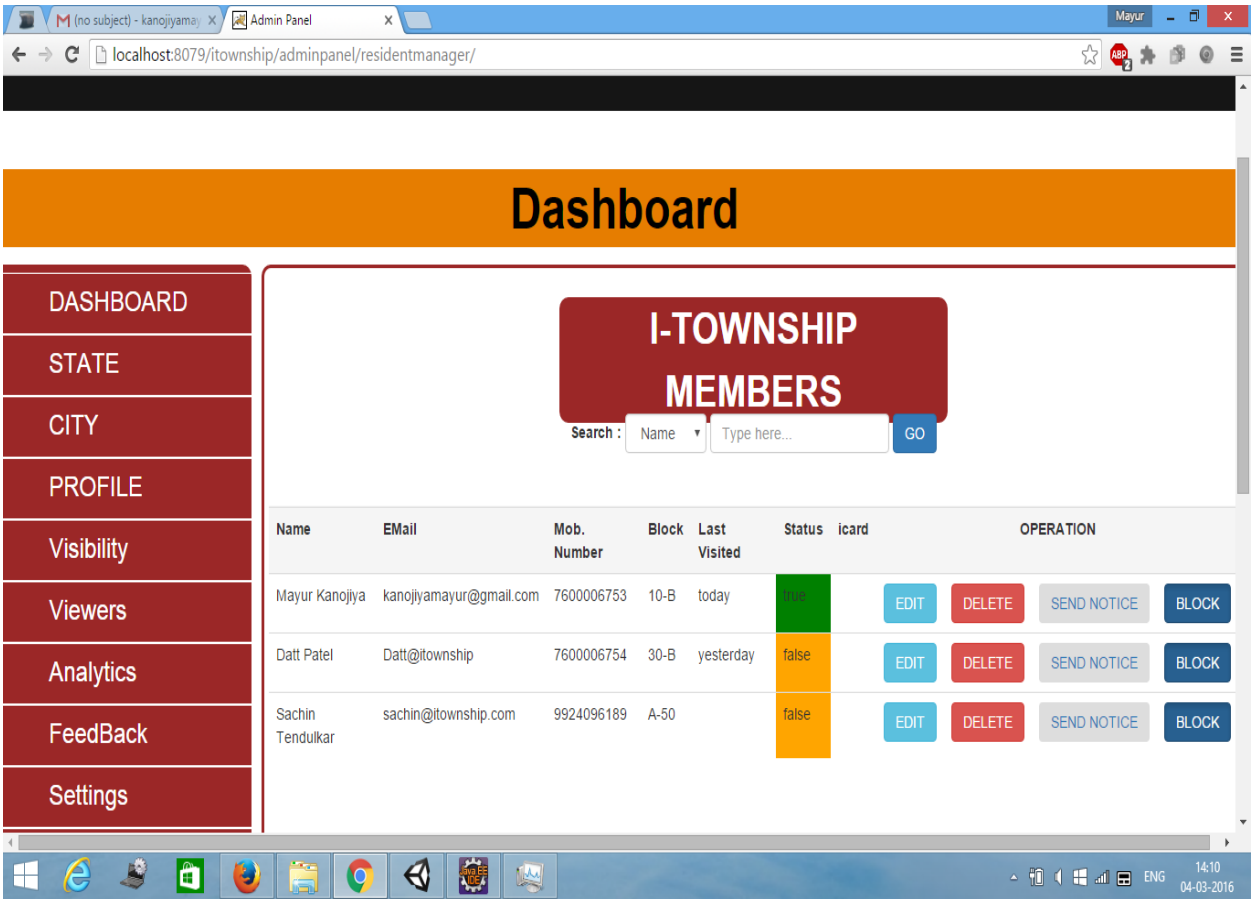


Figure 8.7 Resident Manager

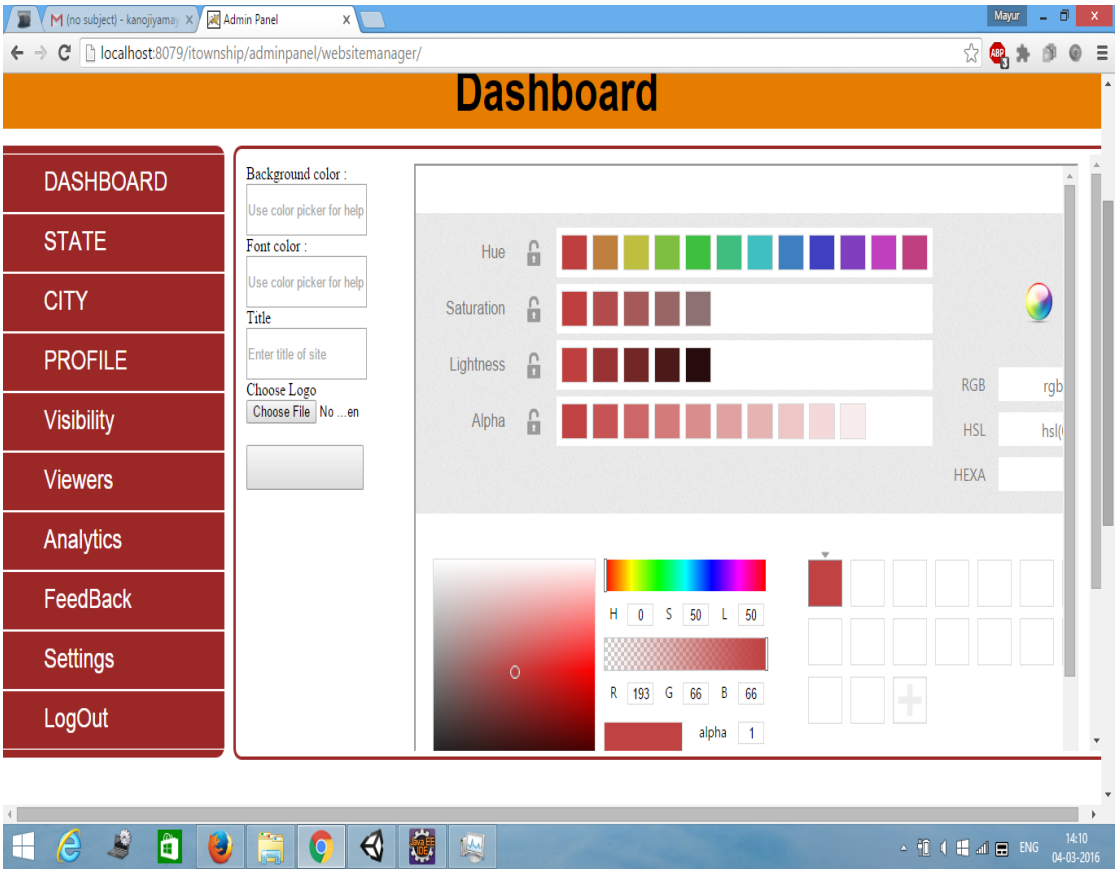


Figure 8.8 Website Manager

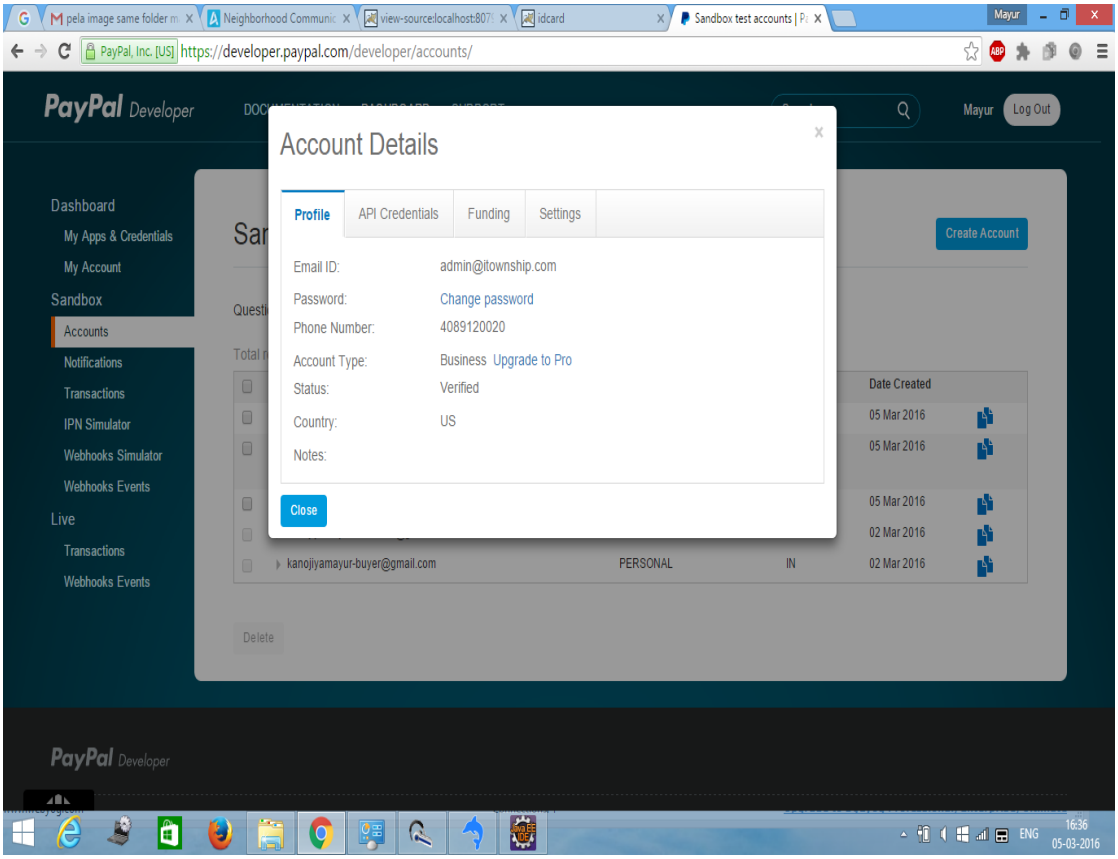


Figure 8.9 Payment Gateway

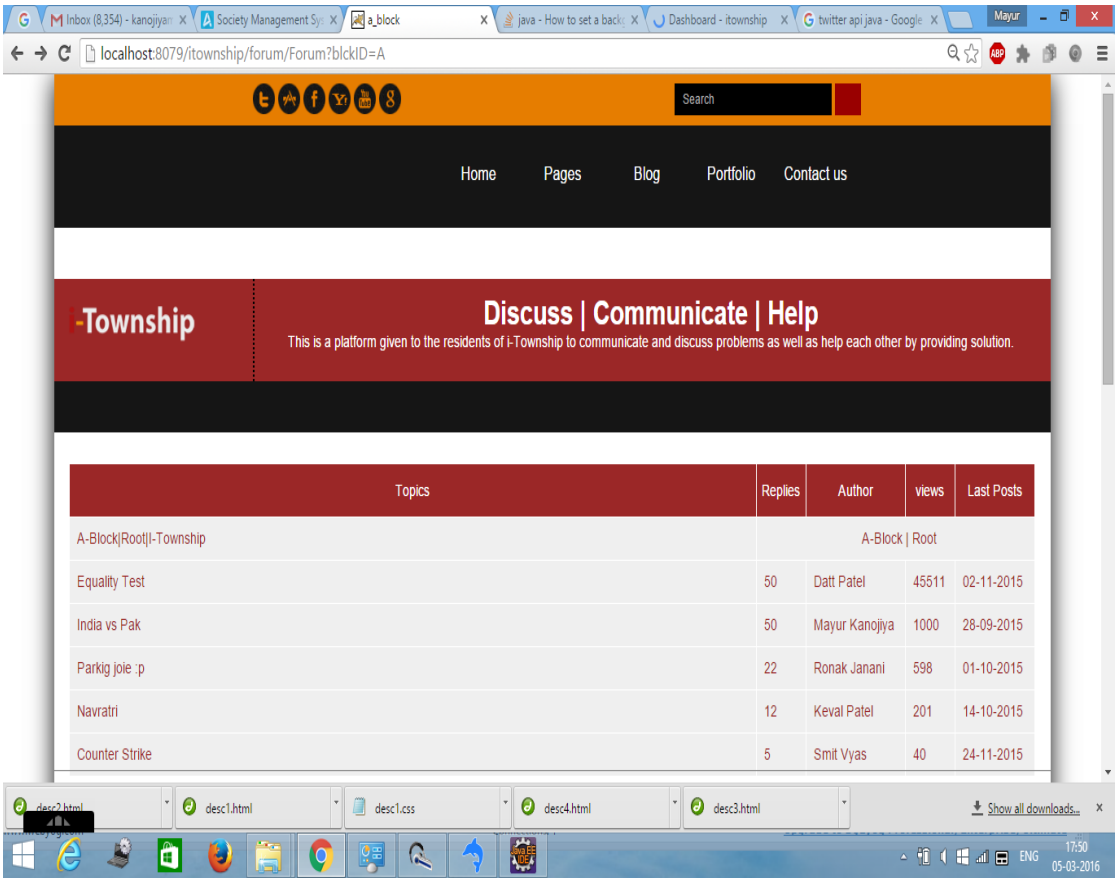


Figure 8.10 My Forum

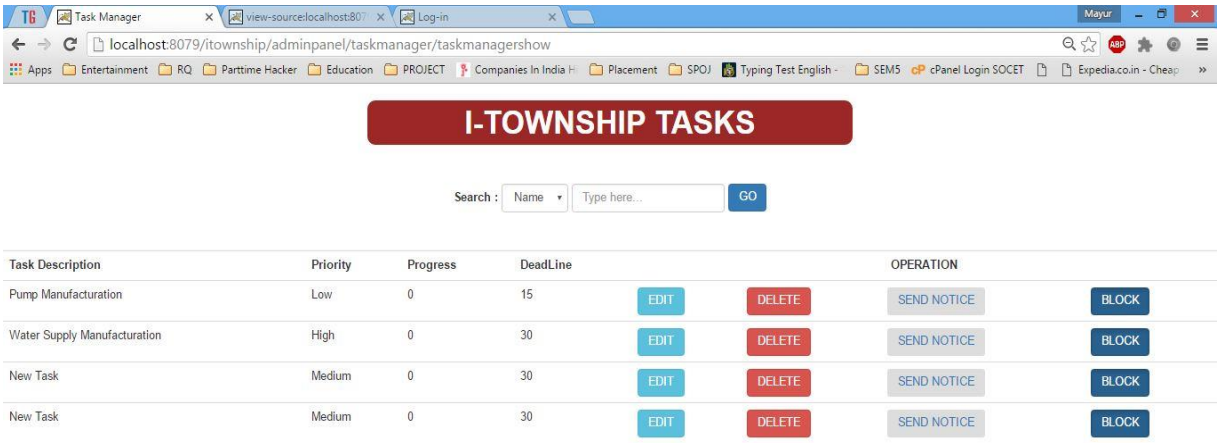


Figure 8.11 Task Manager

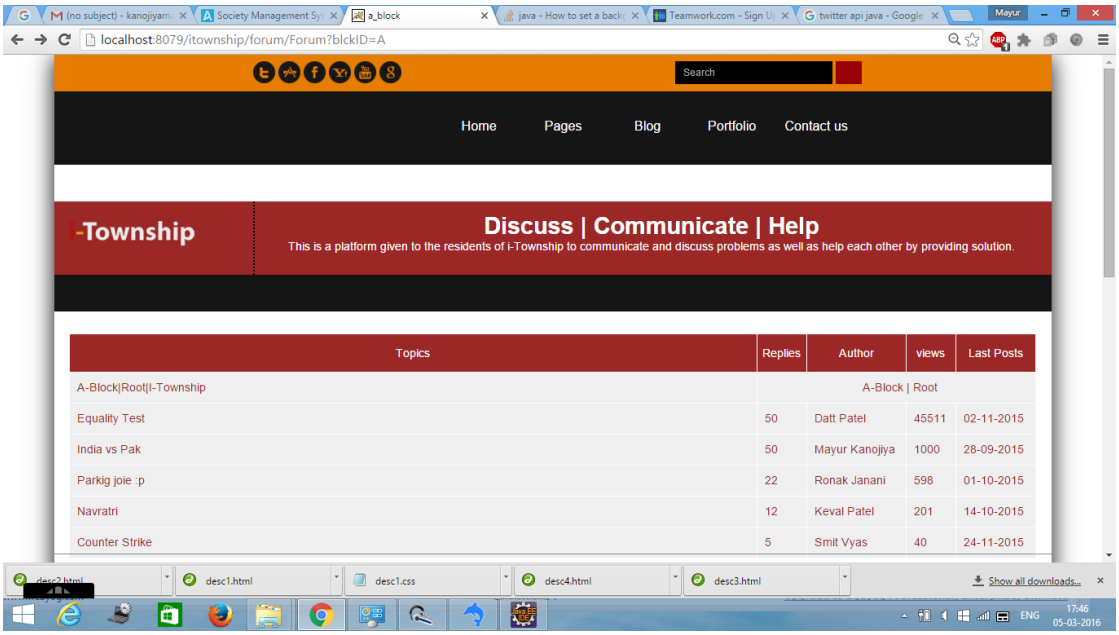


Figure 8.12 Forum Topics

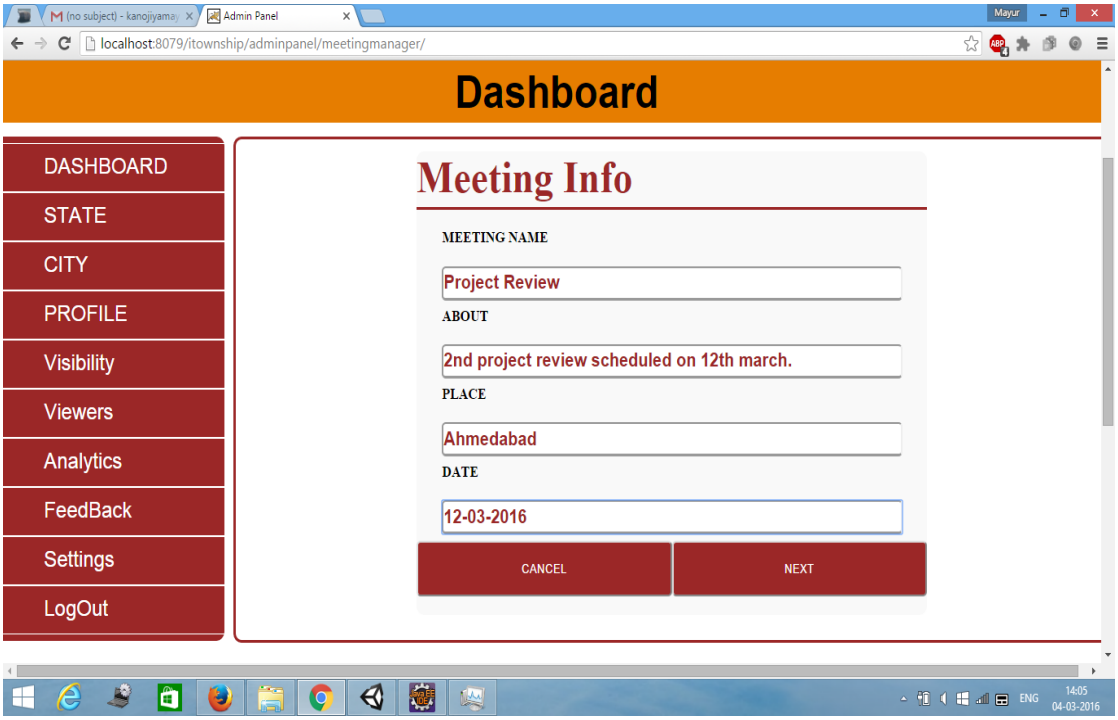


Figure 8.13 Meeting Manager

Figure 8.14 Meeting Manager Report

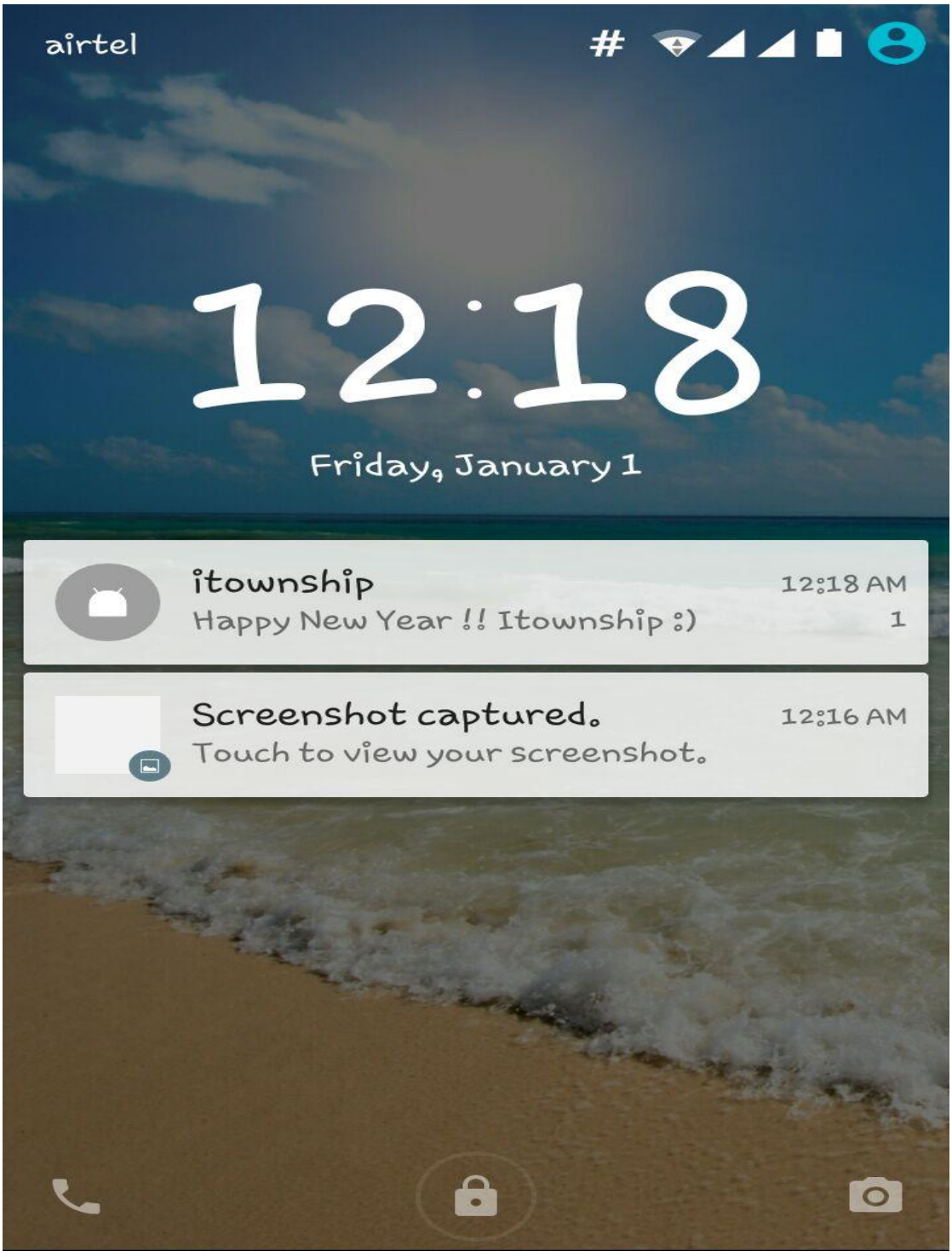


Figure 8.15 Notes

CHAPTER

9

LIMITATIONS AND FUTURE ENHANCEMENT

CHAPTER:9

LIMITATION AND FUTURE ENHANCEMENT

- **LIMITATIONS:**

  Though we tried best in developing this web application but as limitation are mare part of any web app are of our web web app. Some limitations are as follow.

  - As we are not aware of how to use different compression algorithm for compressing the web application, the size of web app is comparatively large .
  - The web application is complex in size, so the execution time & time required to load the maps is high.

- **FUTURE ENHANCEMENT :**

  Enhancements are the perquisite for development of a web application. Every existing web app has proposed enhancements which make it better and easier to use and more secure. The enhancements that have been proposed for this website is listed here.

  - Our web app is running successfully and we are looking for the solution to improve the size of the web application .
  - We are finding the solution to improve the execution time and fetching time from the maps.
  - Though the UI of web application is good , we are looking to make it better . So the user can easily handle it.
  - We are planning to add some new modules and features in the web application .
  - We are planning to upload our web application on hosting server after some improvements on it.

CHAPTER

10

CONCLUSION AND DISCUSSION

CHAPTER:10 CONCLUSION AND DISCUSSION

CONCLUSION :

- We have successfully developed our web application that helps residents to manage society related work remotely and establish communication between other residents.
- Forum provides discussion platform to residents to resolve ongoing issues, present innovative ideas to improve society for better management.
- I-township is supporting several existing features like expense manager, meeting manager and task manager combined with some unique features like real time tracking, expense data management and meeting organization has a vast scope in this era.
- It will run on any web browser, smart phone , tablet and mobile devices.

Bibliography:

- http://developer.android.com/training/basics/fragments/fragment-ui.html
- http://www.vogella.com/tutorials/AndroidBackgroundProcessing/article.html#androidbackground.
- http://www.androidbegin.com/tutorial/android-viewpager-gallery-images-andtexts-tutorial/
- http://stackoverflow.com/questions/6053602/what-arguments-are-passed-intoasynctaskarg1-arg2-arg3
- https://developers.google.com/places/training/additional-places-features
- http://examples.javacodegeeks.com/android/android-google-places-api-example/