# CHAPTER - 1

# INTRODUCTION

# 1. Introduction

**1.1 Project Profile:**

**Project Name :** Smart Society

**Project Type :** UDP

**Development Team :** RACHIT BHANAGE (140413107003)

BHUSHAN JOSHI (140413107009)

NAKUL PATEL (140413107013)

JAY UNEWAL (140413107025)

**Internal Guide :** Prof. Shrina Patel

**Tools & Technology:** MVC .Net , CSS, JavaScript, HTML, jQuery, Android, JSON, XML, Visual Studio, Android Studio

**1.2 Problem Summary:**

Currently, there are lot of problems occurring in a commercial society. Problems like:

* Improper management
* Lack of proper resources • Insecurity
* Opaque bill management
* Unknown vendors in the area
* Unknown residents nearby

To solve these problems, we have tried to provide best possible software called Smart Society.

Smart society is and internet based solution for the end to end management of commercial and housing societies. It is a Web based plus Android based platform for the management of residents and staff, who can only see the information that they require. Through computerized management of resources and timely data, smart society helps in reduction of time, efforts and manual paperwork, leading to lowering overall costs of managing commercial society.

**1.3 Aim and Objectives of the project:**

**AIM:**

* The main objective of the project is to design and develop a user friendly ecosystem making tasks of members easy and productive.
* Reduction in manual paperwork.
* Easy to use and efficient computerized automated system.
* Computerized automation can be helpful as means of saving time & money.
* To provide better graphical user interface in.
* Portable application which can be accessed from anywhere.
* A secure system for household purpose providing good security features

**Objective:**

* The objective of the “Smart Society” is to provide a system which handles the information of the residents of a commercial housing society.
* Providing the residents, a system for using the resources of the society efficiently like booking an asset of the society for a private or public use, checking the vendors available for society, visitor’s information etc.

**1.4 Problem Specification:**

Today’s societies are working manually. The current system is costly and time consuming as it involves a lot of manual paper work which is tedious and tiring.

As we have computerization and automation it is possible to make a system for solving the mentioned problems.

The following are the reasons why the current system should be computerized:

* Recording details of each and every member and staff consumes time.
* It is very difficult to maintain the visitor log information.
* The records are manually maintained which consumes time and energy.
* Lot of paper work is involved as the records are maintained in the registers
* Less efficiency and costly.
* As files and registers are used the they are vulnerable to environmental calamities as they can wear and tear.
* Also files and registers uses lot of space.
* Use of papers for storing valuable data information is not at all reliable.
* Manual systems mean increased human error.
* Manual collection of society maintenance payments is tiring job for the secretary.

**1.5 Existing System and Proposed System Comparison:**

**Drawbacks of Existing System:**

* Time consuming.
* Updating and Retrieval tasks are very tedious and tiring.
* Vulnerable to errors.
* Less social.
* Lot of paper work results in lot of confusion.
* No direct role for the higher officials.
* Zero transparency in expenditures of the maintenance cheques.

To avoid all these limitations and make the system working more accurately it needs to be computerized.

**Merits of Proposed System:**

The system is very simple in design and to implement. The system requires Very low system resources and the system will work in almost all Configurations.

Merits are:

* Security of data and residents/members
* Ensure data accuracy
* Administrator and secretary controls the entire system
* Portable
* E-notice generation.
* Online payments
* Error prone
* Proper event management of assets of society
* Proper Visitor log generation for increased security
* Direct Connection with the security guards for increased security
* Minimize manual data entry
* Greater efficiency
* User friendly and interactive
* Time saving
* Retrieval and Updating tasks can be performed much faster and easier

**CHAPTER - 2**

**DESIGN ANALYSIS, DESIGN METHODOLOGY AND IMPLEMENTATION STRATEGY**

# 2. Design analysis, Design Methodology and Implementation Strategy

**2.1 Requirement Gathering**

**2.1.1 Functional Requirement**

* **Wing configuration and Unit configuration**
  + Create a new wing
  + Configure wing information
  + Define a new unit within a wing
  + Manage all units
  + Browse units
  + Remove wing
  + Remove unit
* **Customer account setup and Unit association**
  + Register through Mobile App
  + Show registration status
  + Approve registration and associate unit
  + Login by customer login
  + View my profile
  + Edit my profile
* **Service catalogue configuration**
  + Create a new service type for
  + Manage service type
  + Remove service type
  + List view of services
  + Grid view of services
* **Service provider configuration and browsing**
  + Select a service type
  + Add new service provider
  + Upload a photo of a service provider
  + Browse service provider list
  + Remove/upload service provider profile
  + View provider profile with photo and pricing
  + Call the provider
  + Send message to provider
* **Service selection and reviews**
  + View all units who have selected a service provider
  + View user reviews for provider
  + Select a service
  + Write review for provider
  + Remove service selection
* **Secretary configuration and note generation**
  + Appoint a customer as secretary
  + Select wing for secretary
  + Generate a system notice
  + Attach optional photo to notice
  + Send notice to all members
  + View/download notice photo
* **Group event invitations and configuration**
  + create a new group event
  + send invitation to all members
  + member’s confirmation
  + summary report to secretary
  + graphical report for confirmation
* **Amenities configuration, usage and registration**
  + Amenities creation with capacity, space and pricing
  + Photo upload for amenity
  + Amenity lookup and availability
  + Amenity reservation and configuration
  + Daily schedule/ monthly schedule
* **Visitor logs and photo check-ins**
  + Watchman visitor check-in
  + Visitor’s photo from app and upload
  + Visitor vehicle number and count with unit
  + Time based check-in
  + Visitor checkout
  + Visitor history report
* **Watchman account configuration**
  + create a new watchman account
  + configure mobile number and details
  + view all watchmen
  + call a watchman
  + upload photo of watchman
  + view photo of watchman

**2.1.2 Non-Functional Requirement**

**Reliability**

* User can be able to access system if he/she is an authorized one.
* Service providers and services available are reliable.

**Maintainability**

* Bugs should be fixed in a quick amount of time when reported.
* Changes, if any, in the schema/data, should be easy to update and alter.

**Performance**

* For very large number of users, the server should be able to handle requests efficiently.
* Data Storing and Retrieval should be quick.
* The system must be interactive and the delays involved must be less .

**Security**

* Only authorized users must be able to view the visitor log, events and services etc.
* Only the secretary who is authorized can update the notice and circulars.
* Data such as password and personal details are in encrypted.

**Scalability**

* The system can be implemented from a small scale to a large scale.

**Safety**

* Information transmission should be securely transmitted to server without any changes in information.

### **2.1.3 Hardware and Software Requirement**

The following Software’s required to fulfil the requirement for our application for development

#### **Minimum Software Configuration**

|  |  |
| --- | --- |
| OS | Windows 10, Android 4.0.3 |
| Front End | Microsoft visual studio 15 |
| Back End | MS SQL 2012 or above |
| Web Server | Microsoft IIS |
| Supportive Development Tools | Microsoft Word 2010 |
| Browser | Chrome, opera |

#### **Minimum Hardware Configuration**

|  |  |
| --- | --- |
| Computer/Processor | Intel Core i5,2.60GHz |
| Memory | 4.00GB RAM |
| Hard disk | 500GB |

### **2.2 STUDY OF FEASIBLITY**

#### **2.2.1 Technical Feasibility**

The following technical feasibility areas were probed during theFeasibility study phase:

* The necessary technology i.e. front-end development tool, back-end database technologies are various tools for developing the system are already available within the organization.
* The front-end tool proposed is easily compatible with the current hardware configuration in the organization.
* The back-end tool proposed has the capacity to hold the data require for using the new system.
* The system is expandable in many dimensions with respect to addition of more functionality, features etc.
* The front-end and the back-end technologies provide a way to preserve the accuracy, reliability and ease of access and data security.
* Begin a web-based system; the system does not require to be installed on all client machines.

#### **2.2.2 Time Feasibility**

* The system is made in such a way that it takes less time in data processing and gives the result as fast as possible.
* Order evaluation is also done fast by estimating total money and recognizing receiver’s address from database.

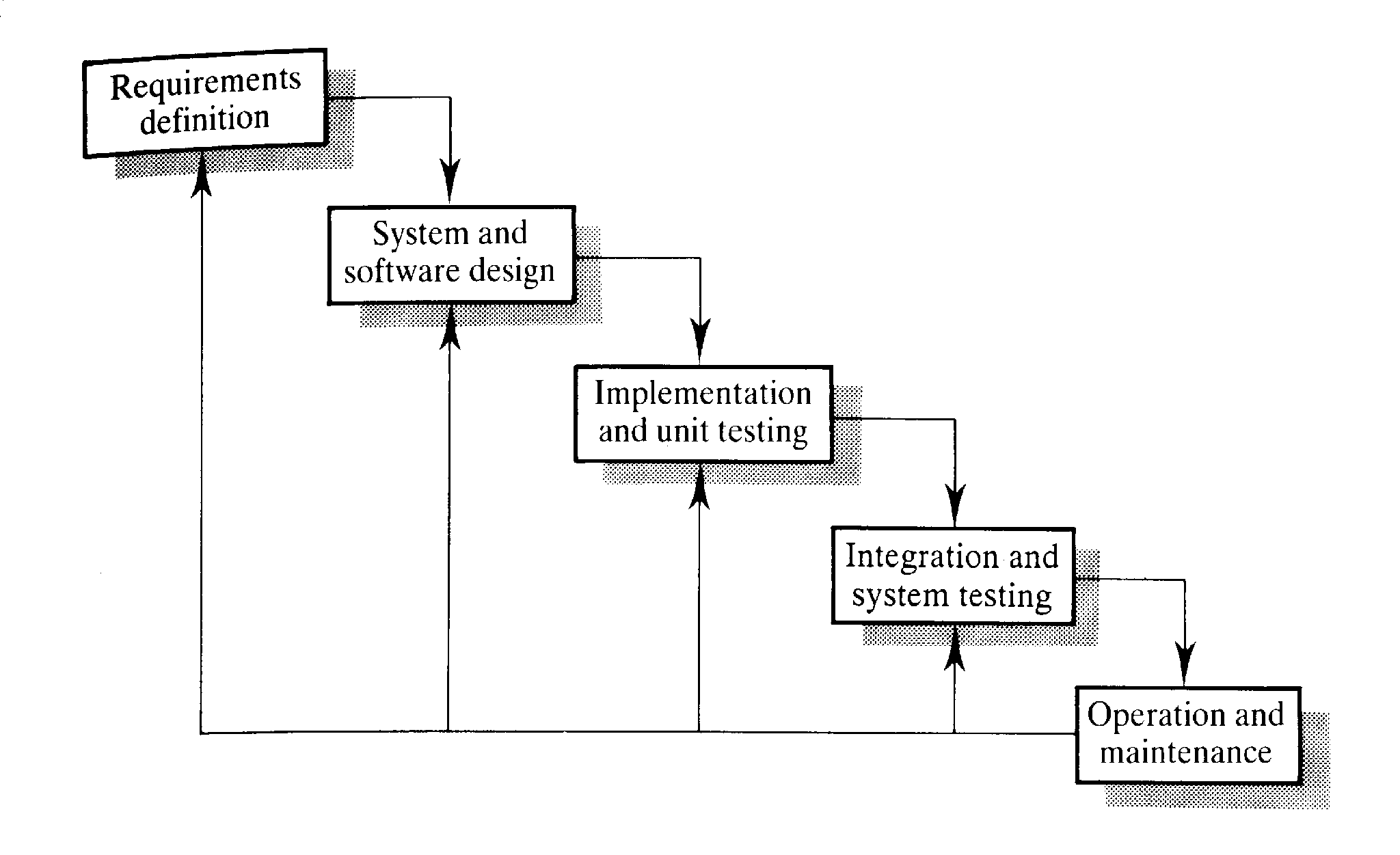
#### **2.2.3 Economical Feasibility**

* The purpose of the economic feasibility assessment is to determine the positive economic benefits to the organization that the proposed system will provide. The following are the benefits that would be derived from the proposed system.
* Speed up the archival of information in the form of corresponding between employees, Approval or disapproval of order etc. When necessary.

All this benefits in terms of saving time, minimization of error, etc. can improve the quality of software being development, since more time can be devoted to the development

### **2.3 SYSTEM DESIGN**

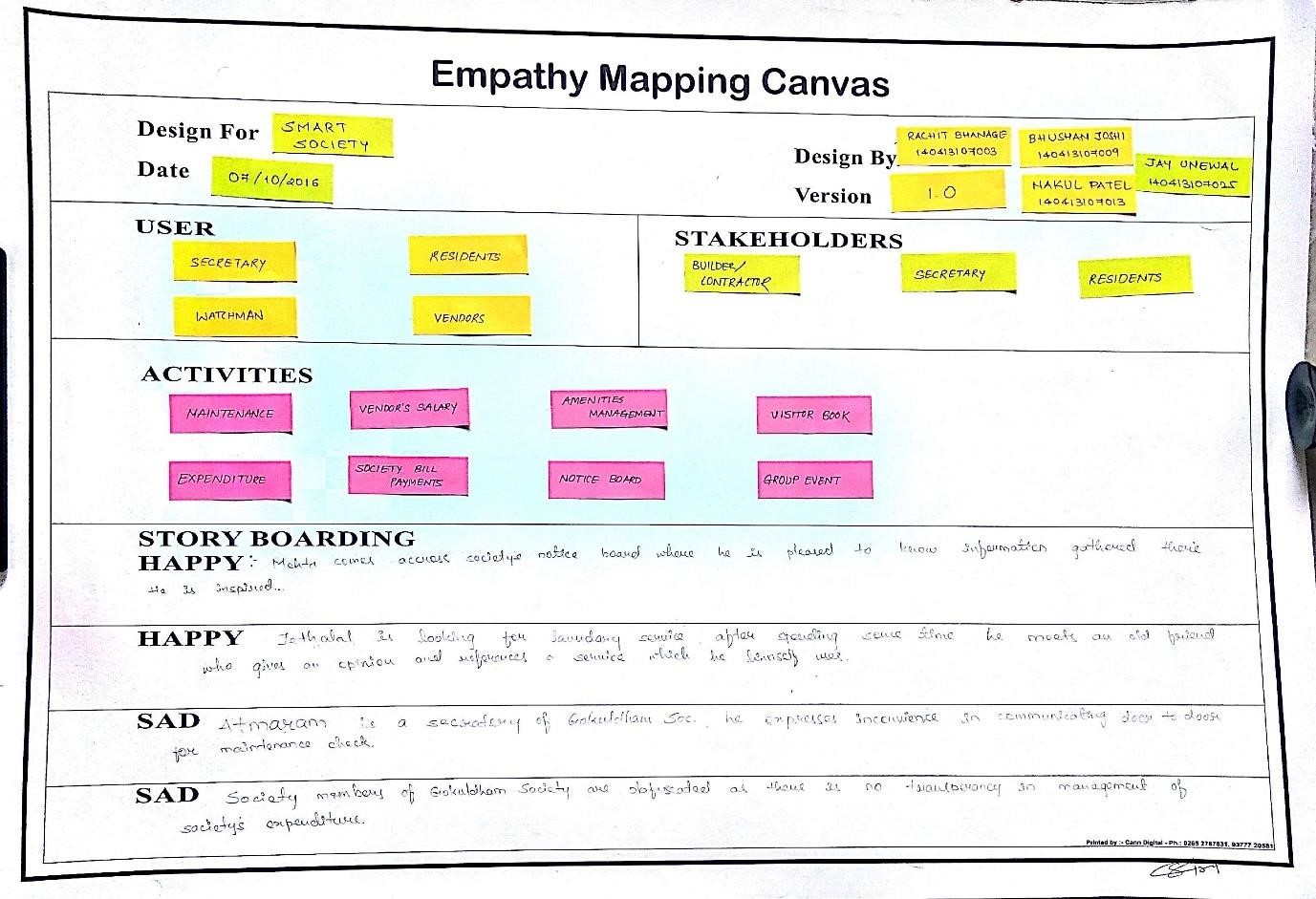
* Water Fall model has been revised later to be Iterative.
* Everything remains the same as Waterfall Model.
* It is easy to use.



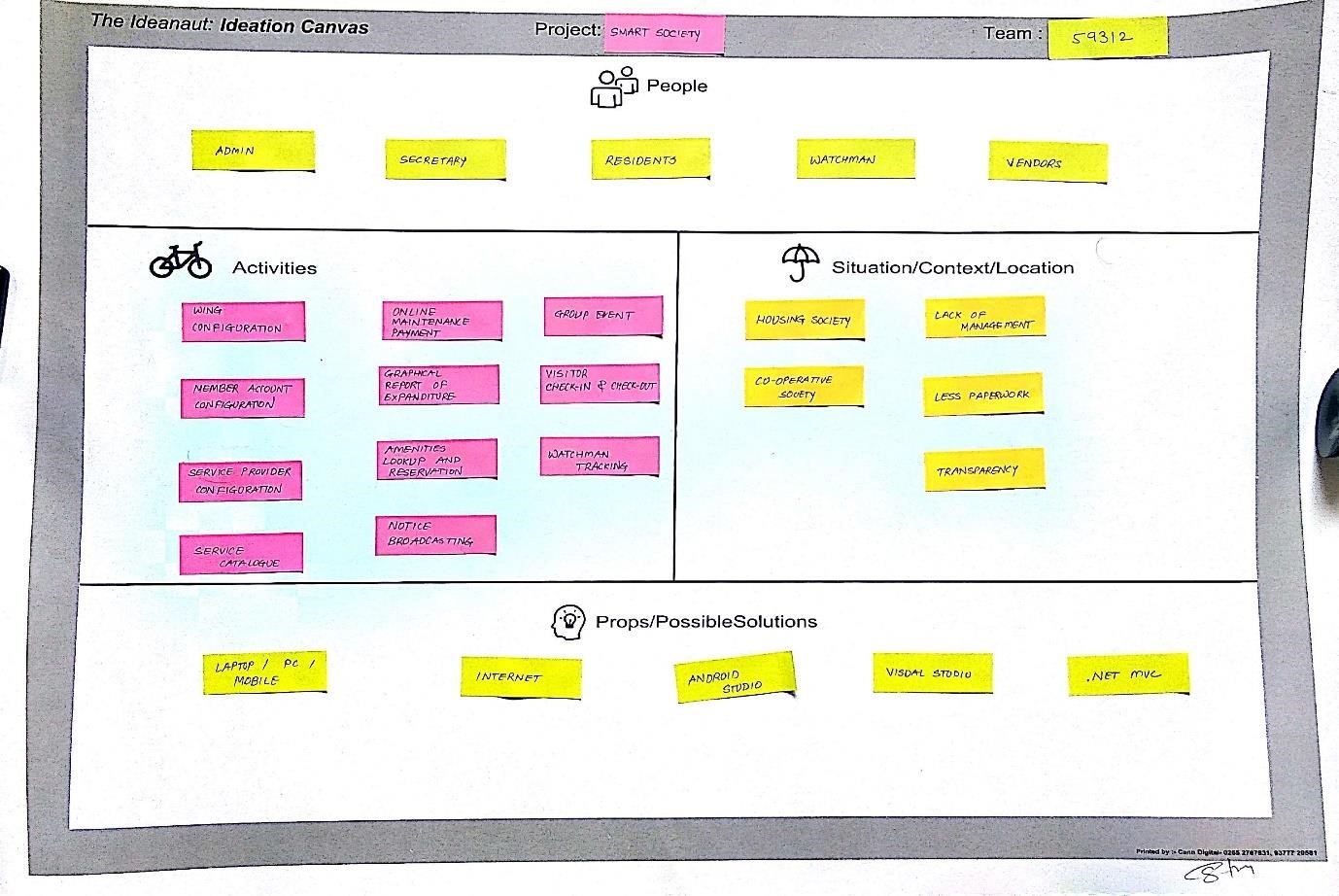
* Each iteration involves Design Analysis and Implementation as well as verification of the current build/version of the system.
* If the application lacks any requirements or has any problems then the phase will be looped back to previous iteration.
* It supports redesign, acceptance and review of any new requirement.
* It involves analysis of usability, achievement of goals, reliability, efficiency, and structure.

**2.4 Canvases**

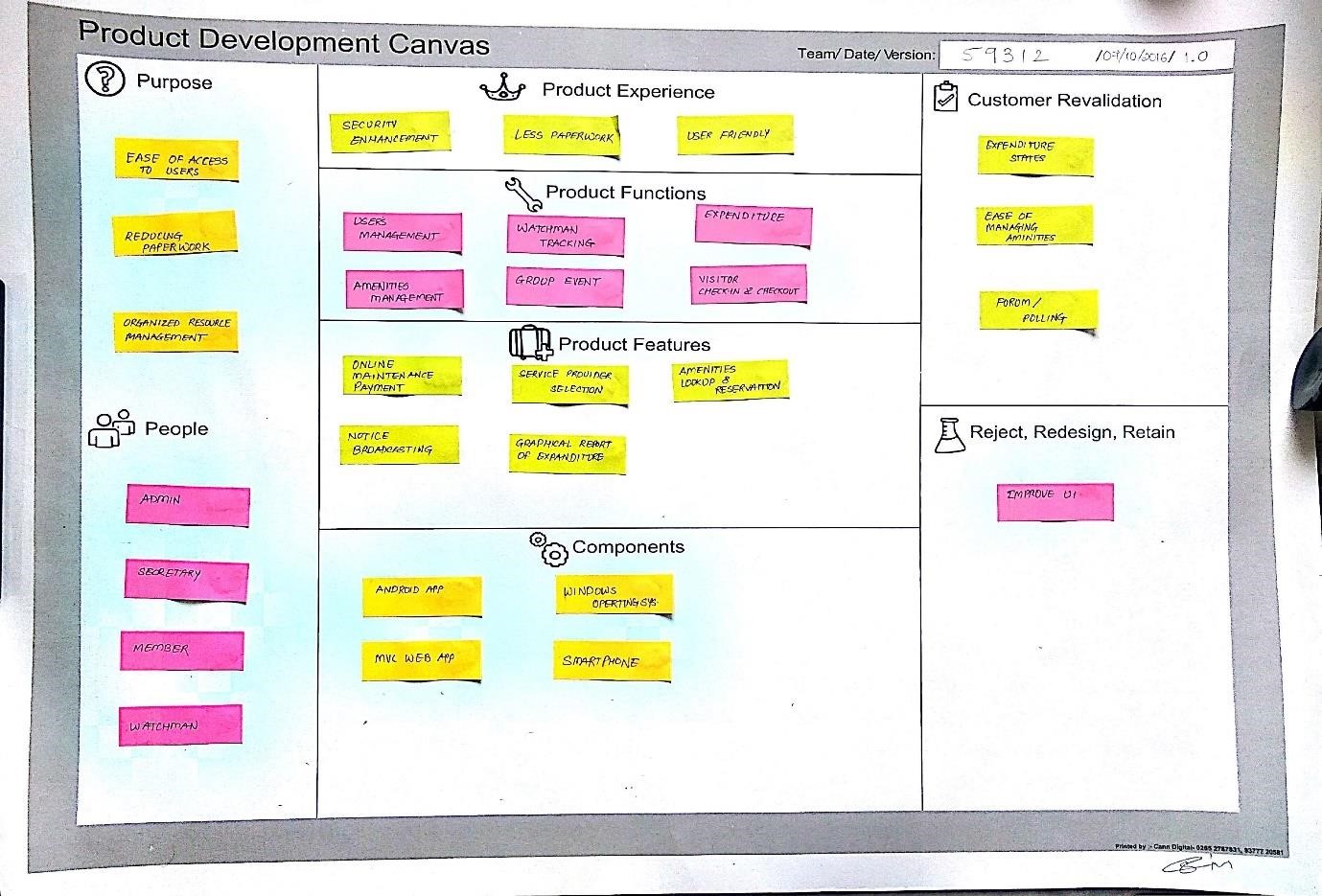
**2.4.1 EMPATHY MAPPING CANVAS**



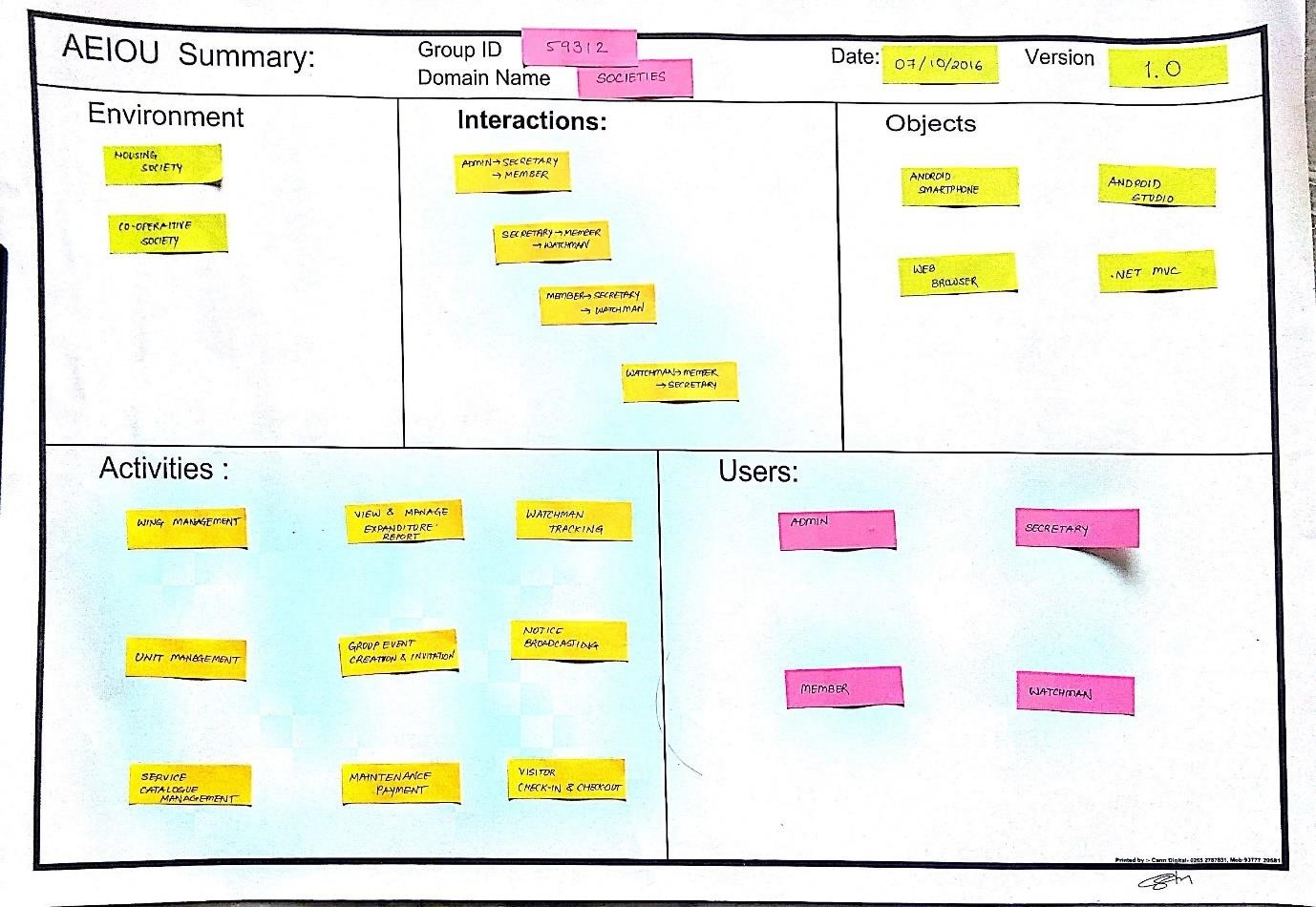
**2.4.2 IDEATION CANVAS**



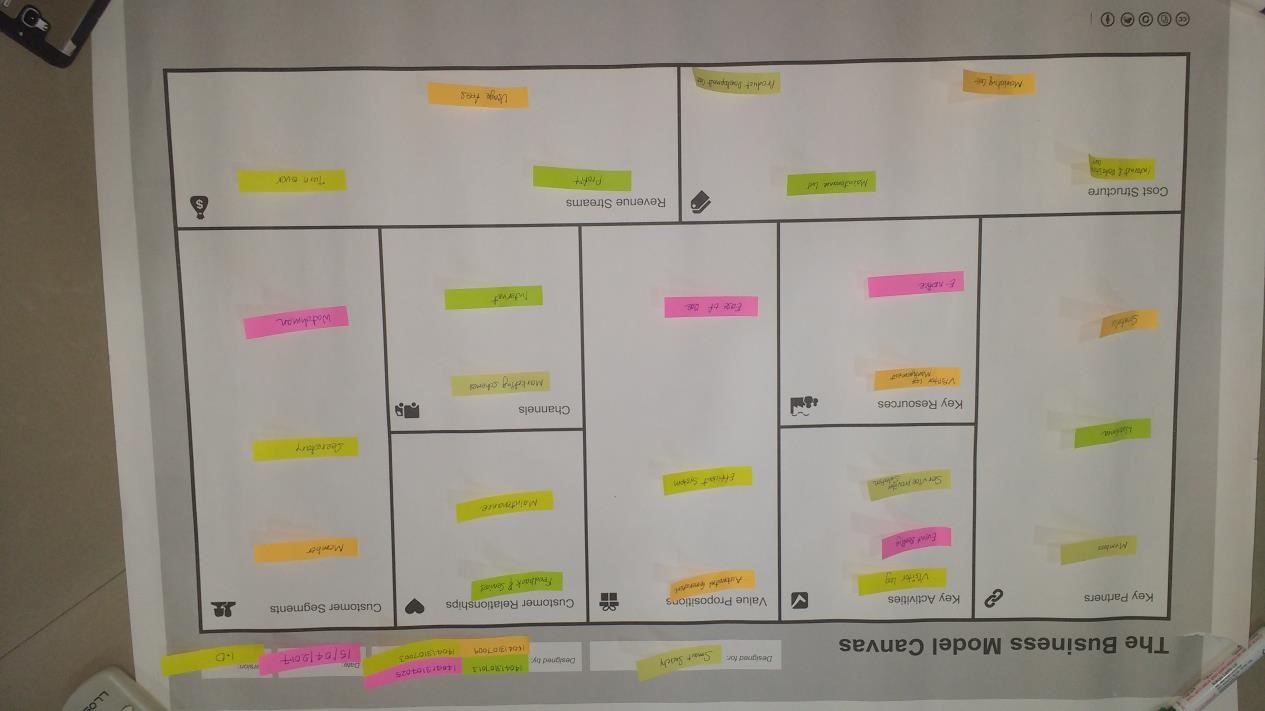
**2.4.3 PRODUCT DEVELOPMENT CANVAS**



**2.4.4 AEIOU FRAMEWORK**



**2.4.5. BUSINESS MODEL CANVAS**



The Business Model Canvas is a strategic management and lean start-up template for developing new or documenting existing business models. It is a visual chart with elements describing a company or product’s value, infrastructure, customers, and finances. It assists firms in aligning their activities by illustrating potential trade-offs.

# KEY PARTNERS

In order to optimize operations and reduce risks of a business model, organization usually cultivate buyer-supplier relationships so they can focus on their core activity. Complementary business alliances also can be considered through joint ventures, strategic alliances between competitors or non-competitors. The following constitute as our key partners:

* Members, Watchmen: The people who use the products and services produced by the business. Customer is the one who demands for goods and services.
* Secretaries: They provide the necessary details about the society layout, facilities and vendor details.

# KEY ACTIVITIES

These are the crucial things the business needs to do in order to deliver on its propositions and make the rest of the business work. It includes maintaining superior expertise on the segment(s) and creating or acquiring products and services that are a good fit, whatever that entails. Similarly for an infrastructure business, it includes keeping the infrastructure working reliably and making it more efficient. The key activities of our business model are as follows:

* Visitor log: It is very important part of any residential or commercial area. To maintain all the records on your figure tap is the ultimate goal.
* Event booking: Residents of the society can post their events in the society App and book the amenities for the event and the secretary will approve or disapprove the event through the system.
* Service provider selection: Residents of the society can know all the service providers who provide different types of services to the other society members and can contact them as well.

# KEY RESOURCES

These are the resources that are necessary to create value for the customer. They are considered an asset to a company, which are needed in order to sustain and support the business. These resources could be human, financial, physical and intellectual. The following are our key resources:

* Visitor log: Watchman will be able to check in and checkout of the visitor through the application only (no need for the paper). All the data will be stored on the server securely.
* E-notice: all the circulars and the general notices can be viewed with the image and the attachment by the members and can only be uploaded by the secretary or admin.

# VALUE PROPOSITIONS

It includes problems and requirements of people, that we are going to solve and satisfy through our product and various reasons for which they would like to choose our product over the rest. The various value proposition of our product are as follows:

* Automated generation: Our system completely eliminates the manual system and provides an automated expenditure.
* Efficient system: High degree of efficiency as it removes the errors that manual efforts are prone to.
* Ease of use: removes unnecessary complexity and makes it very user friendly system to interact with.

# CUSTOMER RELATIONSHIPS

When it comes in increasing profits, it's tempting to concentrate on making new sales or pursuing bigger accounts. But providing attention to your existing customers, no matter how small they are, is essential to keep your business thriving. The following will be our initiatives to build strong

* Feedback and Service: Take the feedback of customers about the product and provide support for the product in terms of servicing, part replacement, and warranties.
* Maintenance: It is a facility provided to repair damaged or faulty product of customer in minimum time period.

# CHANNELS

An organization can reach its clients either through its own channels, partner channels or a combination of both. The channels which we may use are:

* Marketing Schemes: Marketing schemes mean different ways to represent the product in market so buyers gain interest.
* Internet

# CUSTOMER SEGMENTS

This section mainly focuses on three things: Segment Dimensions, Segment composition and Problems, Needs, Habits & Current Alternatives.

* Members
* Secretary
* Watchman

# COST STRUCTURE

* Product Development cost: It is the cost of production along with the initial investment in raw material.
* Maintenance cost: It is the cost of maintenance of all the equipment’s, machinery and salary of employees.
* Marketing cost: It is the cost of marketing and advertisement of product.
* Internet and data storage cost: costs for the connection and data-storage.

**REVENUE STREAMS**

# • Profits • Usage fees • Turn over

**CHAPTER – 3**

**SYSTEM DESIGN & IMPLEMENTATION**

# 3. SYSTEM DESIGN AND IMPLEMENTATION

**3.1 SYSTEM DIAGRAMS**

### 3.1.1 Use Case Diagram

3.1.1.1 Secretary configuration and notice generation

Secretary configuration and notice generaton

M

e

m

ber

Appoint a secretary

Notice generation

View and download a

notice and attachment

secretary wing

allocation

Secretary

Upload a notice

attachment

Send notice

Admin

3.1.1.2 Member account setup and unit association

Member account setup and Unit association

Member

Register through

mobile App

Show registration

status

Approve registation

Associate unit

Profile

Login through App

View profile

Edit profile

<<

extend

>>

<<

extend

>>

Secretary

Admin

3.1.1.3 Maintenance

Maintenance

Member

Collects maintenance

amount

Generates expanditures

report

Pays maintenance

amount

Secretary

View expanditure report

3.1.1.4 Group event creation and invitation

Group event configuration and inv

i

tat

i

on

Member

Create an Event

Give confirmtion

Send invitation

Secretary

View graphical report of

confirmation

3.1.1.5 Amenities creation and requisition

Amenities configuration and requisition

Member

Create and amenity

Amenity lookup and

availability

Update amenity

photo

Secretary

View schedule

Monthly schedule

Daily schedule

Amenity reservation and

confirmation

<<

extend

>>

<<

extend

>>

3.1.1.6 Service catalogue configuration

Service catalogue configuration

Member

Create a service type for

a service

Remove service

type

Show grid view of

services

Upload a photo for

a service

Show list view of

services

Manage Service

type

Secretary

3.1.1.7 Service Provider configuration and browsing

Service provider configuration and browsing

Member

Select a service

type

Upload a photo of a

service provider

Call the provider

through App

Send message to the service

provider through app

Browse service

provider list

Add service

provider

Secretary

update service

provider profile

View service provider

profile

3.1.1.7 : Service provider selection and review

Service provider selection and reviews

Member

View list of subscribers of a

service provider

Write reviews for

service provider

Remove service selection

Select a service

provider

Secretary

View user's review for

service provider

3.1.1.8 Visitor log and photo check-in

Visitor logs and photo check

-

in

Visitor Checkin

Enter vehicle number

and unit

Upload a photo of a

visitor from App

Watchman

Visitor history report

Visitor checkout

3.1.1.9 Watchman account configuration

Watchman account configuration

Member

create a watchman

View all watchmen

Configuration of

watchman profile

Secretary

View watchman profile

Call watchman

3.1.1.10 Wing creation and configuration

Wing creation and configuration

Admin

Create a Wing

Configure a Wing

Define a Unit within a Wing

Manage all Units

Browse all Units

Remove Unit

Remove Wing

By name

By unit

By wing

<<

extend

>>

<<

extend

>>

<<

extend

>>

<<

include

>>

**3.1.2 Activity Diagram:**

3.1.2.1 Admin

Login

Valid Credentials

Invalid Credentials

Create New Wing

Define New Unit

in WIng

Manage Units

Browse Unit

Wing wise

Appoint

Secretary

Select Wing for

Secretary

Generate Notice

Send Notice to

Members

Logout

3.1.2.2 Member

Login

Correct Credentials

Incorrect Credentials

View Profile

Manage Serv

i

ce

Edit Profile

Subscribe

Remove

Subscr

i

pt

i

on

Receives Invitation

Confirmation

Amenities Lookup

Reserve Amenities

Available

Not Avaliable

Confirmation

Approved

Disapproved

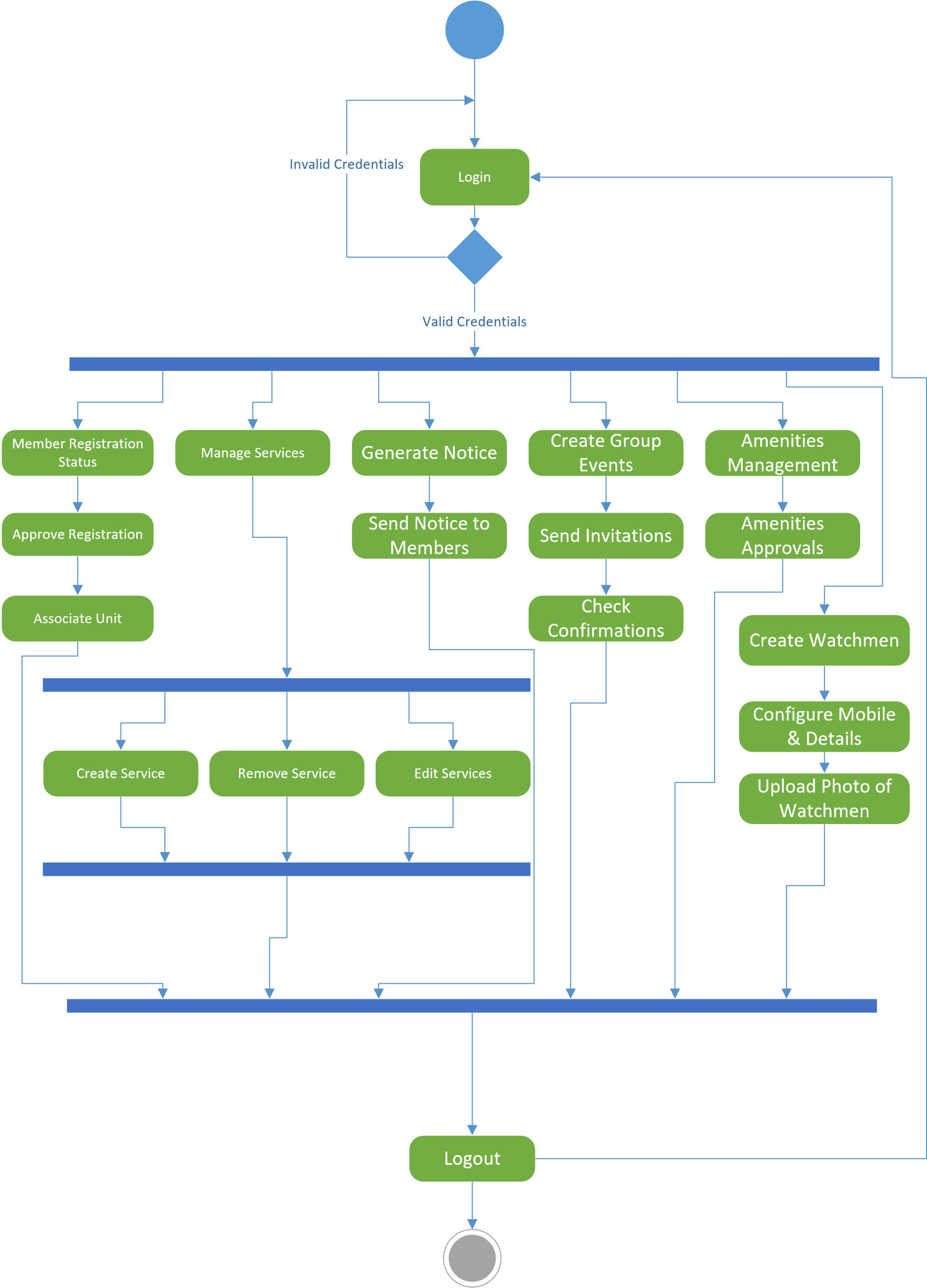
View Watchmen

Call Watchmen

Review & Rate

Service

3.1.2.3 Secretary



3.1.2.4 Watchman

Login

Valid Credentials

In

v

alid Credentials

Visitor Checkin

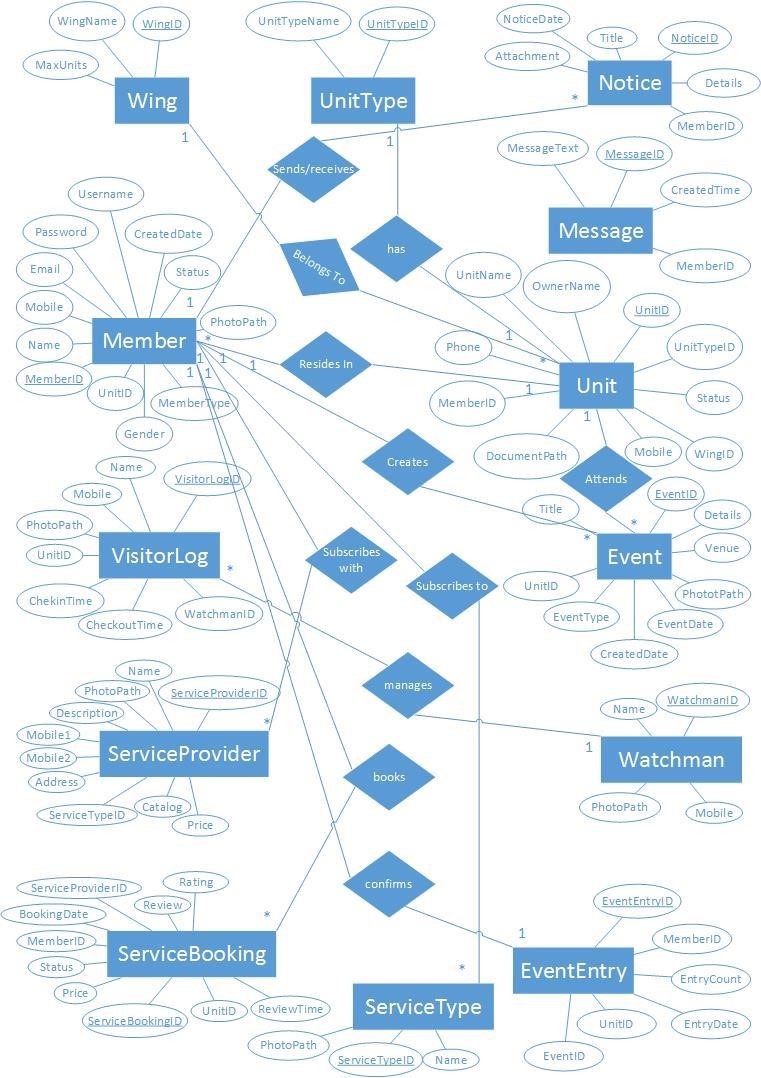
Visitors Photo

Vehicle Number

Current Time

Checkout

**3.1.3 E-R diagram**



**3.2 Data Dictionary**

**Wing**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Column Name | Datatype | Key | Nullable |
| 1 | WingID | Int | Primary Key | No |
| 2 | WingName | Varchar(100) |  | Yes |
| 3 | MaxUnit | Int |  | Yes |

**Unit**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Column Name | Datatype | Key | Nullable |
| 1 | UnitID | Int | Primary Key | No |
| 2 | UnitName | Varchar(100) |  | Yes |
| 3 | WingID | Int | Foreign Key | No |
| 4 | UnitTypeID | Int | Foreign Key | No |
| 5 | Status | Varchar(100) |  | Yes |
| 6 | OwnerName | Varchar(100) |  | Yes |
| 7 | DocumentPath | Varchar(100) |  | Yes |
| 8 | Mobile | Varchar(100) |  | Yes |
| 9 | Phone | Varchar(100) |  | Yes |
| 10 | MemberID | Int | Foreign Key | No |

**UnitType**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Column Name | Datatype | Key | Nullable |
| 1 | UnitTypeID | Int | Primary Key | No |
| 2 | UnitTypeName | Varchar(100) |  | Yes |

**Member**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Column Name | Datatype | Key | Nullable |
| 1 | MemberID | Int | Primary Key | No |
| 2 | Name | Varchar(100) |  | Yes |
| 3 | Mobile | Varchar(100) |  | Yes |
| 4 | Email | Varchar(100) |  | Yes |
| 5 | Username | Varchar(100) |  | Yes |
| 6 | Password | Varchar(100) |  | Yes |
| 7 | CreatedDate | DateTime |  | Yes |
| 8 | UnitID | Int | Foreign Key | No |
| 9 | Status | Varchar(100) |  | Yes |
| 10 | PhotoPath | Varchar(100) |  | Yes |
| 11 | Gender | Varchar(100) |  | Yes |
| 12 | MemberType | Varchar(100) |  | Yes |

**ServiceType**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Column Name | Datatype | Key | Nullable |
| 1 | ServiceTypeID | Int | Primary Key | No |
| 2 | Name | Varchar(100) |  | Yes |
| 3 | PhotoPath | Varchar(100) |  | Yes |

**ServiceProvider**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Column Name | Datatype | Key | Nullable |
| 1 | ServiceProvideID | Int | Primary Key | No |
| 2 | Name | Varchar(100) |  | Yes |
| 3 | PhotoPath | Varchar(100) |  | Yes |
| 4 | ServiceTypeID | Int | Foreign Key | No |
| 5 | Description | Varchar(500) |  | Yes |
| 6 | Mobile1 | Varchar(100) |  | Yes |
| 7 | Mobile2 | Varchar(100) |  | Yes |
| 8 | Address | Varchar(100) |  | Yes |
| 9 | Catalog | Varchar(100) |  | Yes |
| 10 | Price | Varchar(100) |  | Yes |

**Notice**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Column Name | Datatype | Key | Nullable |
| 1 | NoticeID | Int | Primary key | No |
| 2 | Title | Varchar(100) |  | Yes |
| 3 | Details | Varchar(500) |  | Yes |
| 4 | Attachment | Varchar(100) |  | Yes |
| 5 | NoticeDate | DateTime |  | Yes |
| 6 | MemberID | Int | Foreign Key | No |

**Event**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Column Name | Datatype | Key | Nullable |
| 1 | EventID | Int | Primary Key | No |
| 2 | Title | Varchar(100) |  | Yes |
| 3 | PhotoPath | Varchar(100) |  | Yes |
| 4 | Details | Varchar(500) |  | Yes |
| 5 | Venue | Varchar(100) |  | Yes |
| 6 | EventDate | DateTime |  | Yes |
| 7 | CreatedDate | DateTime |  | Yes |
| 8 | EventType | Varchar(100) |  | Yes |
| 9 | UnitID | Int | Foreign Key | No |

**ServiceBooking**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Column Name | Datatype | Key | Nullable |
| 1 | ServiceBookingID | Int | Priimary Key | No |
| 2 | UnitID | Int | Foreign Key | No |
| 3 | ServiceProviderID | int | Foreign Key | No |
| 4 | BookingDate | DateTime |  | Yes |
| 5 | MemberID | Int | Foreign Key | No |
| 6 | Status | Varchar(100) |  | Yes |
| 7 | Price | Varchar(100) |  | Yes |
| 8 | Review | Varchar(100) |  | Yes |
| 9 | Rating | Varchar(100) |  | Yes |
| 10 | ReviewTime | DateTime |  | Yes |

**EventEntry**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Column Name | Datatype | Key | Nullable |
| 1 | EventEntryID | Int | Primary Key | No |
| 2 | EventID | Int | Foreign Key | No |
| 3 | UnitID | Int | Foreign Key | No |
| 4 | MemberID | Int | Foreign Key | No |
| 5 | EntryCount | Int |  | Yes |
| 6 | EntryDate | DateTime |  | Yes |

**VisitorLog**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Column Name | Datatype | Key | Nullable |
| 1 | VisitorLogID | Int | Primary Key | No |
| 2 | Name | Varchar(100) |  | Yes |
| 3 | Mobile | Varchar(100) |  | Yes |
| 4 | PhotoPath | Varchar(100) |  | Yes |
| 5 | UnitID | Int | Foreign Key | No |
| 6 | CheckinTime | DateTime |  | Yes |
| 7 | CheckoutTime | DateTime |  | Yes |
| 8 | WatchmanID | int | Foreign Key | No |

**Watchman**

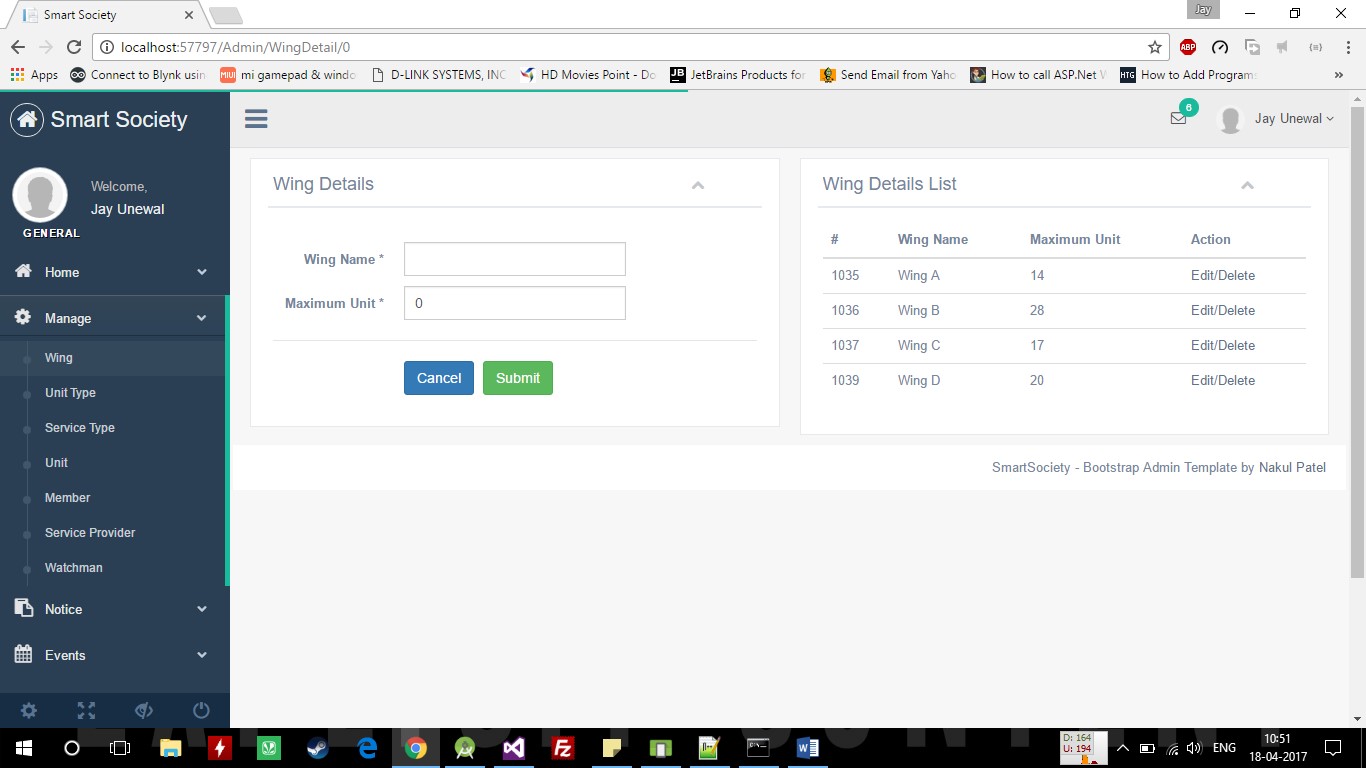
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Column Name | Datatype | Key | Nullable |
| 1 | WatchmanID | Int | Primary Key | No |
| 2 | Name | Varchar(100) |  | Yes |
| 3 | Mobile | Varchar(100) |  | Yes |
| 4 | PhotoPath | Varchar(100) |  | Yes |

**Message**

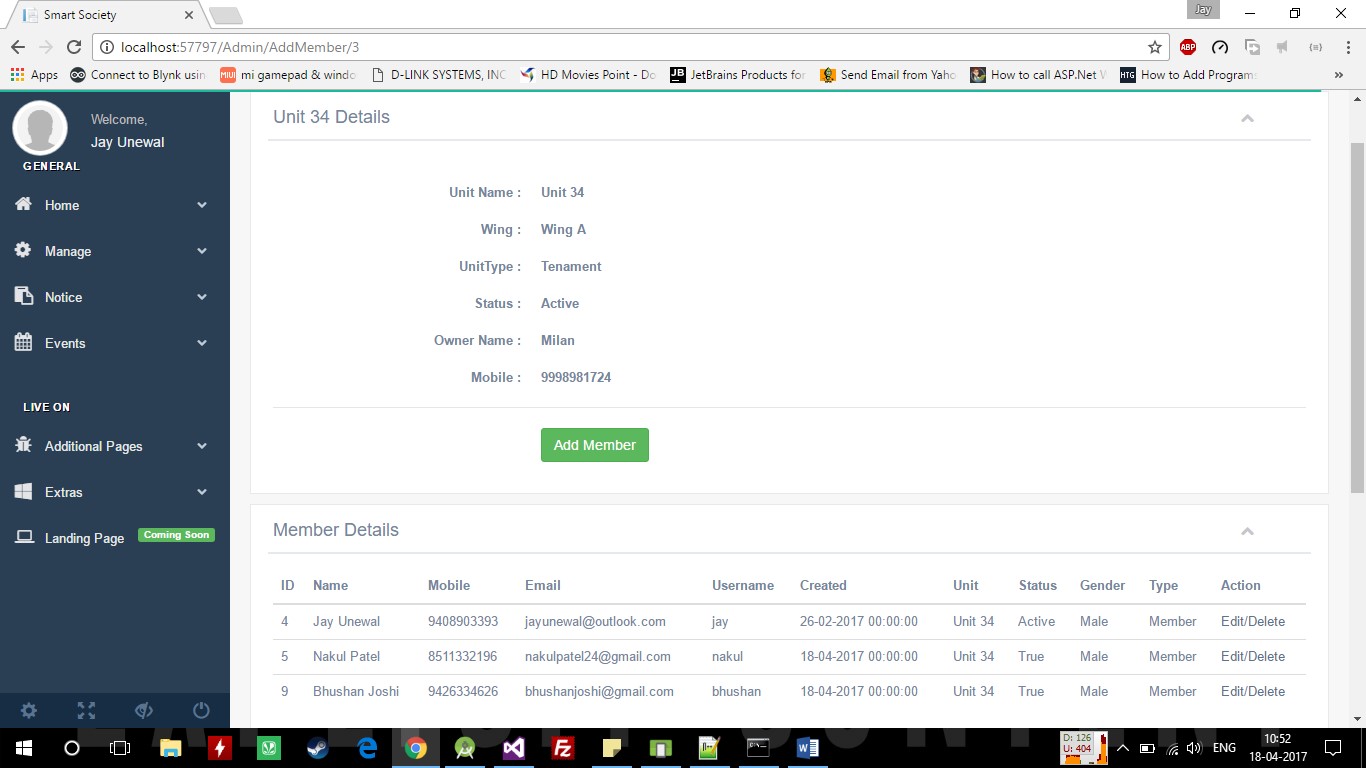
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Column Name | Datatype | Key | Nullable |
| 1 | MessageID | Int | Primary Key | No |
| 2 | MessageText | Varchar(100) |  | Yes |
| 3 | CreateTime | DateTime |  | Yes |
| 4 | MemberID | int | Foreign Key | No |

**3.3 Application Screenshots /Implementation:**

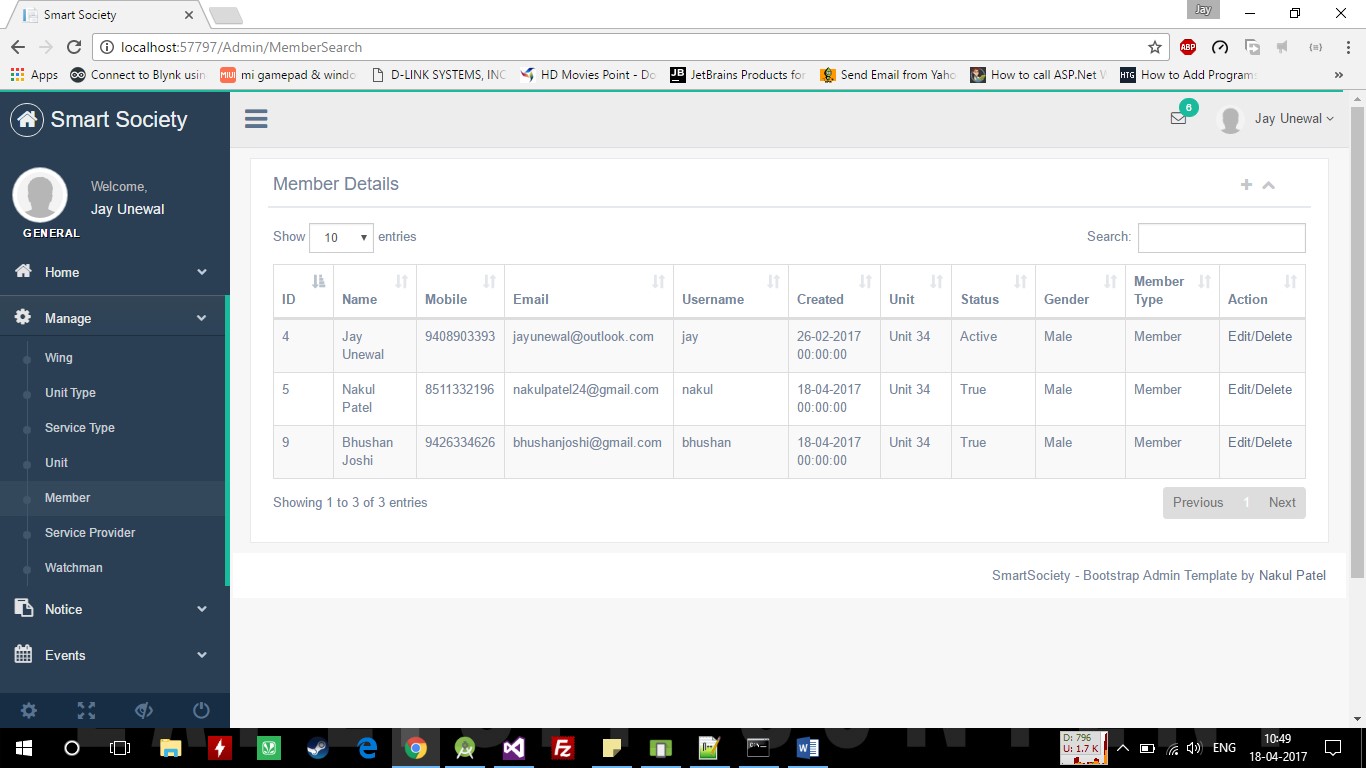
## Wing



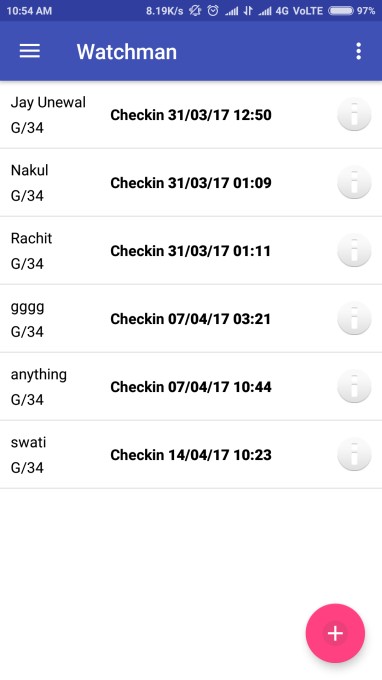
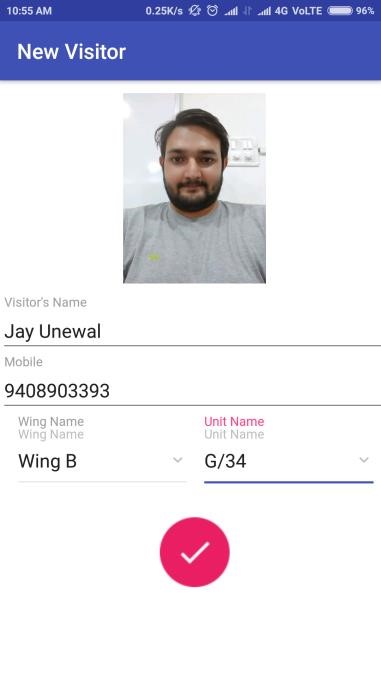
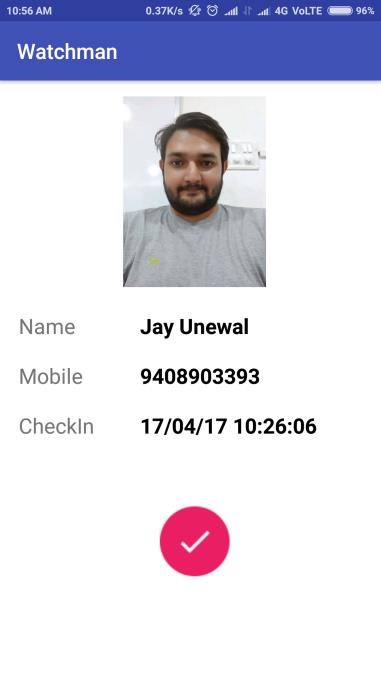
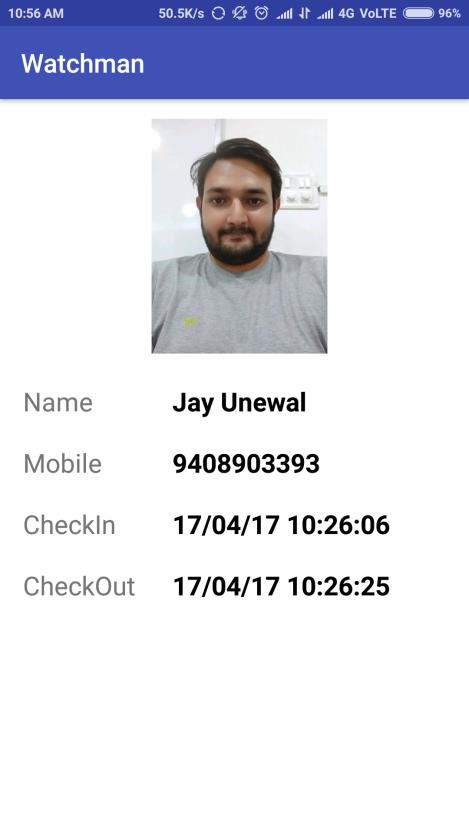
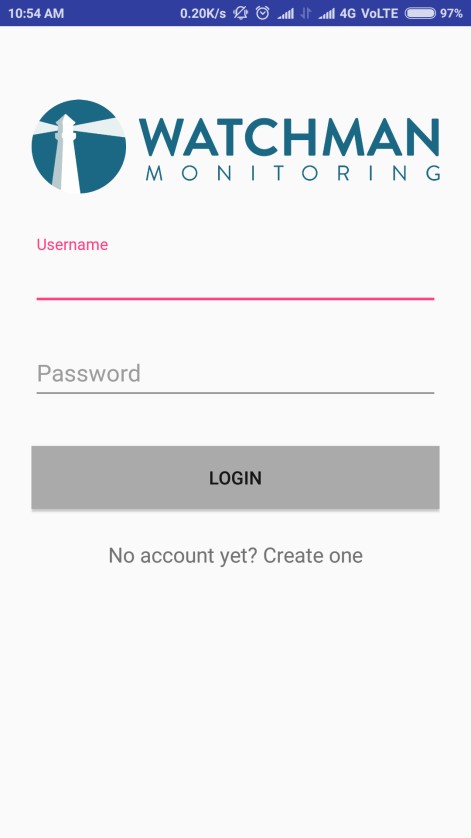
**Unit/Add Member**



**Member**



## Watchman



## 3.4 TESTING

Software testing is a process of executing a program or application with the intent of finding the software bugs. It can also be stated as the process of validating and verifying that a software program or application or product:

Meets the business and technical requirements that guided it’s design and development. Works as expected. Can be implemented with the same characteristic.

Let’s break the definition of Software testing into the following parts:

1. **Process:** Testing is a process rather than a single activity.
2. **All Life Cycle Activities:** Testing is a process that’s take place throughout the *Software Development Life Cycle (SDLC).* The process of designing tests early in the life cycle can help to prevent defects from being introduced in the code.

Sometimes it’s referred as “verifying the test basis via the test design”. The test basis includes documents such as the requirements and design specifications.

1. **Static Testing:** It can test and find defects without executing code. Static Testing is done during verification process. This testing includes reviewing of the documents (including source code) and static analysis. This is useful and cost effective way of testing. For example: reviewing, walkthrough, inspection, etc.
2. **Dynamic Testing:** In dynamic testing the software code is executed to demonstrate the result of running tests. It’s done during validation process. For example: unit testing, integration testing, system testing, etc.
3. **Planning:** We need to plan as what we want to do. We control the test activities, we report on testing progress and the status of the software under test.
4. **Preparation:** We need to choose what testing we will do, by selecting test conditions and designing test cases.

**Evaluation:** During evaluation we must check the results and evaluate the software under test and the completion criteria, which helps us to decide whether we have finished testing and whether the software product has passed the tests.

# CHAPTER - 4

**CONCLUSION**

**Conclusion**

Smart society is an android plus and MVC .net web application proposed for an efficient interaction between members of a society, secretary and admin for managing and utilizing various resources. Moreover, it provides visitor status which implies in taking care of all the visitor check in and check out information, keeping track of those activities for the formation of more secure system.

Efficient database management will be provided by this system which will reduce the effort and the time for managing the data.

These changes results in more portable, secure, efficient, resourceful, interactive system which is vital for every member living in a society.

**Future scope:**

The software has been developed in such a way that it can accept modifications and further changes. The software is very user friendly and future any changes can be done easily.

Software restructuring is carried out. Software restructuring modifies source code in an effort to make it amenable to future changes. In general, restructuring does not modify the overall program architecture. It tends to focus on the design details of individual modules and on local data structure defined within modules.

Every system should allow scope for further development or enhancement. The system can be adapted for any further development. The system is so flexible to allow any modification need for the further functioning of programs.

Since the objectives may be brought broad in future, the system can be easily modified accordingly, as the system has been modularized. The future expansion can be done in a concise manner in order to improve the efficiently of the system.

**CHAPTER - 5**   
**BIBLIOGRAPHY AND REFERENCES**

**References:**

* Google
* Wikipedia
* http://smart-society.co.in/
* [Database Management Systems: Designing & Building Business Applications W/ Workbook + Cd](https://www.goodreads.com/book/show/180223.Database_Management_Systems)
* [Learning Android](https://www.goodreads.com/book/show/9968133-learning-android?from_search=true)
* [Hello, Android: Introducing Google's Mobile Development Platform](https://www.goodreads.com/book/show/3429841-hello-android?from_search=true)
* Learn MVC
* Samay Software
* [.Net Knowledge Book: Web Development with ASP.Net MVC and Entity Framework: .Net Knowledge Book: Web Development with ASP.Net MVC and Entity Framework](https://www.goodreads.com/book/show/22809559-net-knowledge-book?from_search=true)
* [Net Knowledge Book : Web Development with Asp.Net MVC and Entity Framework](https://www.goodreads.com/book/show/23835287-net-knowledge-book?from_search=true)

**CHAPTER - 6**

**APPENDIX**