

# **Software Requirements Specification (SRS)**

## **for**

### **Society Management System (SMS)**

**Group Members:** 1. Vaibhavi Shelar (1213)  
2. Vrushali Mohite (1217)

#### **List of Actors:**

- Administrator
- Secretary
- House/Flat Owners

#### **Introduction**

##### **1.1 Purpose:**

This document is meant to delineate the features of SMS, so as to serve as a guide to the developers on one hand and a software validation document for the prospective client on the other.

SMS are designed to provide utility to daily operations of society. This software helps maintains society Maintenance as well as cultural funds and other payments. Secretary can add Electricity billing, Water billing, security maintenance and also manage flat owner data. Then Flat owner need to add there family members and if owner have any issue then owner can complaint online through this system. And all other rest work manage by system like send payments notification through mails and managing due date. Secretary can add events and cancel events, issue notice from notice board.

##### **1.2 Scope:**

This system allows secretary to add flat owners data, bills, events and issue notice. Flat owners allow add family members details and have any issue then apply online complaints.

##### **1.3 Definitions:**

SMS – Society Maintenance Management System

SRS – Software Requirement Specification

GUI– Graphical User Interface

UML– Software Engineering Notation for visualising System in the form diagrams (Unified Modelling Diagram)

SSL–Secure Socket Layer used for providing restricted access to application.

##### **1.4 Overview:**

This System Provide for Society maintenance. It help to manage daily society operations online using SMS.

### **1.5 Additional Information:**

The system work on internet server, so it will be operated by any Society secretary and Flat owners to manage society's works. Also it keep society information.

### **1.6 General Description:**

This SMS system helpful for society secretary to manage society maintenance.

### **Functional Requirements:**

This section provides requirement overview of the system.

#### **2.1 Features:**

- Managing details of flat Owners
- Billing Management
- Events Managing
- Complaint Management
- Share Announcement
- Security Managements

#### **2.2 Technical Issues:**

It will require an internet server. The system should support some commonly used browser such as Chrome etc.

There will be screen displaying information about society.

The Secretary may select different options which open another screen as

- Login page
- Registration page
- Add new flat owner details
- Add flat details
- See member list
- Issue Bills

- Announce Event
- Update Notice
- Logout page

The House Owner may select different options which open another screen as

- Login page
- Registration page
- Update Profile (Add Family members details)
- Bills Payment
- About Events
- Notice Board
- Logout page

### **2.3 Hardware Interface:**

The System must run over the internet. All the hardware shall require to connect to internet will be hardware interface for the system.

### **2.4 Software Interface:**

The system is on server so it requires the scripting languages like JAVA(.jsp), Javascript (Node.js,angular).

The system should be able to exchange data using XML, JASON or any other.

The system require DataBase for the store the Data of members, Society information and many others like MySql etc.

System also require DNS (Domain Name space) for the naming on the internet. E.g. <http://housing-society.com>  
<http://mysociety.com>

At the end-user need web browser for interact with the system.

### **2.5 Performance Requirement:**

There is no such performance requirement in this system, because the server request and response to client is totally based on internet connection. Only if House Owner Complaint about any issues then secretary have to manage that complaint.

## **2.6 Design Constrains:**

This system should be developed using Standard Web Page Development Tool, which conforms GUI standards such like HTML, XML, JSON, etc. The system should support various RDMS.

## **Non-Functional Requirements:**

### **2.1 Security: SSL**

The System use SSL (Secure Socket Layer) in all transactions that include any confidential customer information.

The system's back-end servers shall only be accessible to authenticated administrators.

### **2.2 Reliability:**

The system provides storage of all databases on redundant computers with automatic switchover. The main pillar of reliability of the system is the backup of the database which is continuous maintained and update to reflect the most recent changes.

### **2.3 Availability:**

The system should be available at all times meaning the user can access it using web browser, only restricted by the down time of the server on which the system runs. In case of a hardware failure or database corruption, a replacement page will be shown.

uptime : It mean  $24 * 7$  availability

### **2.4 Maintainability:**

A commercial database is used for maintaining the database and application server takes care of the site. The maintainability can be done efficiently.

### **2.5 Portability:**

The application is HTML and scripting language based (Javascript). So the end user part is fully portable and any system using any web browser should be able to use the features of the system, including any hardware platform that is available or will be available in the future. An end-user is used this system on an OS; either it is Windows or Linux. The System shall run on PC, Laptops, etc. The technology should be transferable to different environments easily.

### **2.6 Accessibility:**

Only registered users should be allowed to process the orders after authentications.

### **2.7 Policies:**

The system should maintain security related to sensitive data.

### **2.8 Efficiency:**

The system should provide good throughput and response to multiple users without burdening the system by using appropriate number of servers.

### **2.9 Safety:**

Software should not harm ethical and environmental conditions of the end users machine.

### **2.10 Modularity:**

The system should have user friendly interface.

It should be easily updated, modified and reused.

### **Operational Scenario:**

- Admin Interaction with system:

Admin Manages Society details accordingly. Manage and authorization and authentication, manage system for work smoothly.

- Secretary Interaction with system:

Secretary want to register himself as to manage society maintenance. Using system Secretary update profile, Manipulate members list, Update Bills, Update notice board, Update events details, Update fund calculations, Check Complaint Box, etc.

- House Owner Interaction with system:

House owner have to register to use system. Using this system House owner can Update Profile as uploading image, Add Contact details, Add family Members details, Add Flat details, etc also make maintenance payments online, see about events in society, Notice Boards, etc.

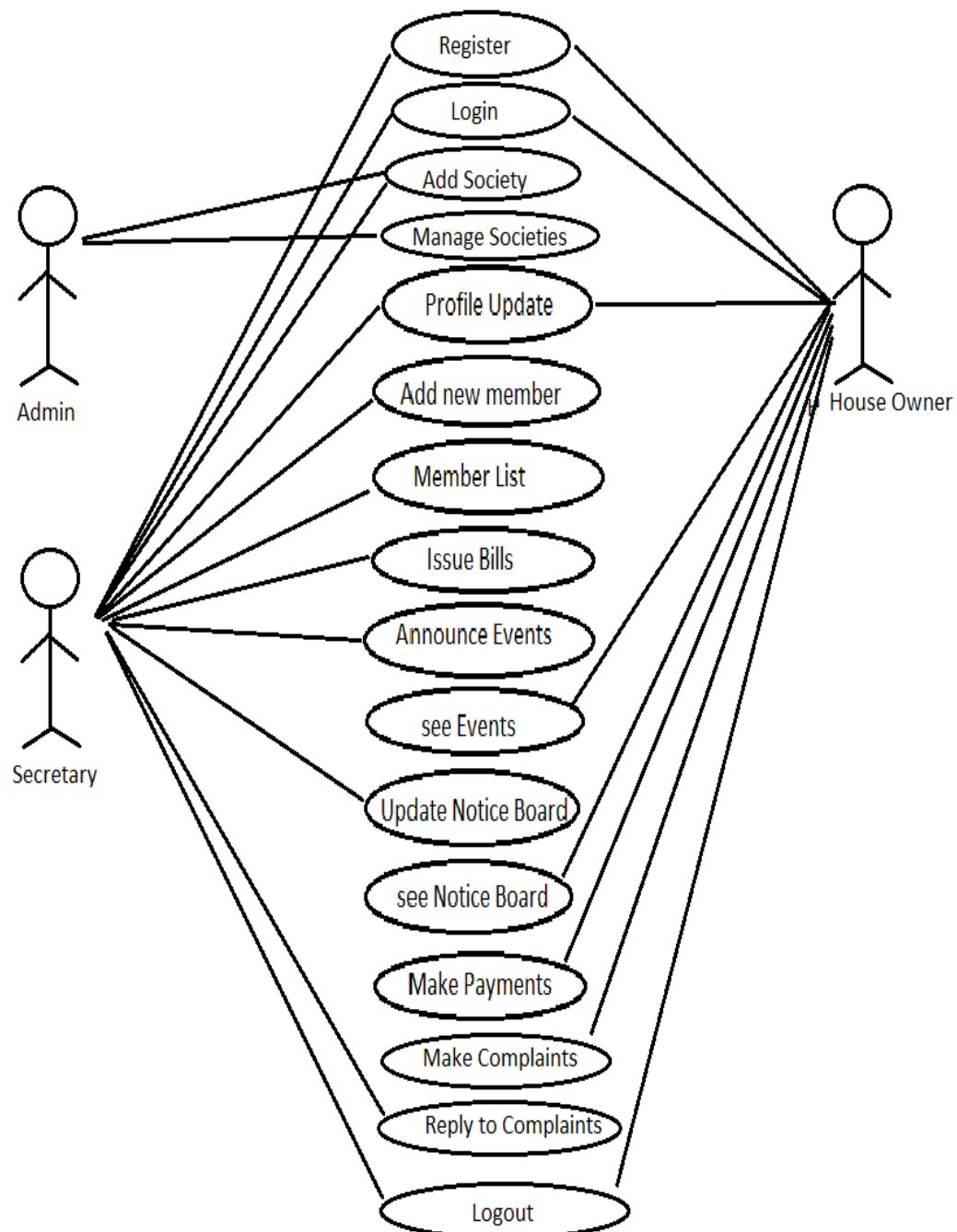
### **Preliminary Schedule:**

- Login
- Manage database
- Event Portal Manage
- Manage Billing
- Payment
- Registration
- Update Profile
- Updating notice board
- Updating member List

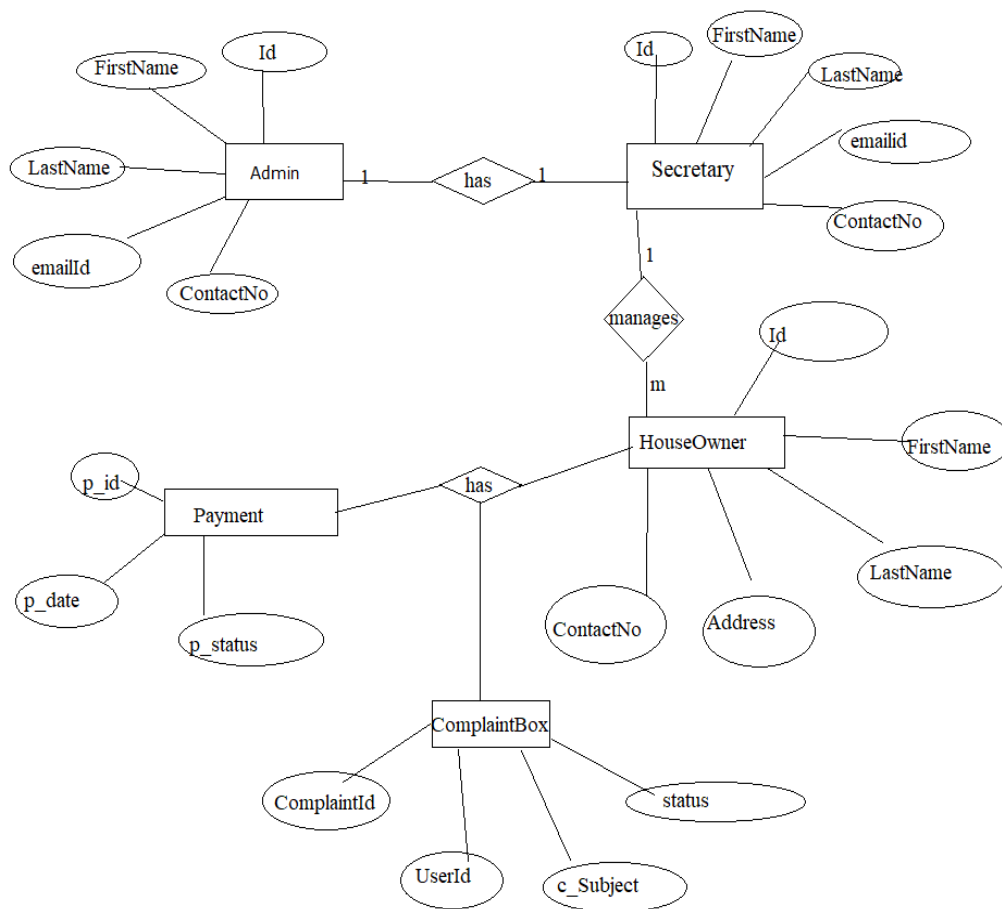
- Complaint Box
- Logout

## UML Diagrams:

- Use Case Diagram:



- ER Diagram:



- Class Diagram:

