**Veer Narmad South Gujarat University,**

**Surat.**

**Department of Information and**

**Communication Technology**

**B.Sc. (Information Technology)**

**Project Report**

**6th Semester**

**M.Sc. (Information Technology)**

**5 Years Integrated Course**

**Year 2021 – 2022**

SOCIETY MANAGEMENT SYTSEM

**Guided By:** **Submitted By:**

**Dr. Tejas Shah Thumma Chandra Prakash. D**

**SPID:2019008987**

**Department of ICT**

**J.P.Dawar Institute Of Technology**

**Veer Narmad South Gujarat University,Udhna Magdalla Road,Surat.**



**Index**

|  |  |  |
| --- | --- | --- |
| **Sr. No** | **Topics** | **Page no** |
| **1** | **Introduction**  **1.1 Current System** | **5** |
| **2** | **Proposed System**   * 1. **Scope**   2. **Objective**   3. **Constraints**      1. **H/w**   **Constraints**  **2.3.2 S/W Constraints**   * 1. **Advantages**   2. **Limitation** | **6** |
| **3** | **Environment Specification**   * 1. **Hardware & Software Requirements**   2. **Development Description** | **8** |
| **4** | **System Planning**   * 1. **Feasibility Study**   2. **Risk Analysis** | **9** |
| **5** | **System Analysis**   * 1. **Detailed SRS**   2. **UML Diagram**      1. **Use Case Diagram**      2. **Sequence Diagram**      3. **DFD Diagram**   3. **E-R Diagram** | **10** |
| **6** | **Software Design**   * 1. **Database Design**   2. **Interface Design** | **29** |
| **7** | **Testing**  **7.1 Unit Testing** | **49** |
| **8** | **Future Enhancement** | **49** |
| **9** | **Reference** | **49** |

**INTRODUCTION**

**1.1 Current System**

|  |  |
| --- | --- |
| **Project Title** | Ambika row House |
| **Project Definition** | Ambika Row House– Online society management system web application is used by the customers who wants to buy and sell/rent their row house. There are different sellers in which seller add their property and make the property online and make it visible to the site. Admin manages the seller property.  There is no specific customer to these application. It’s a platform to make business easier and available to all types of customers. |
| **Project Type** | Website |
| **Website Category** | Online Society Management System |
| **Front End Technology** | HTML, CSS, BootStrap5, JavaScript |
| **Back End Technology** | PHP, MySQL |
| **Project Duration** | 3 Months |

**2. PROPOSED SYSTEM**

* 1. **Scope**

Ambika row house– Online Society Management website have three main panel like Admin, Member. We provide each of them some additional functionality. Our future scope is providing some more advance features for our website users so they can easily use it. Admin can manage all the details of website and also block member. member can search any property.

member can upload their property and their details.

# **Objective**

* + 1. Provides the system administrator with easy interface for different administrative tasks such as adding, deleting and updating of different entities.
    2. The system administrator will be able to manage Property and their Feature and also block Member.
    3. The system should properly maintain user privilege i.e., it should only allow registered member to access certain features, which it should not allow for unregistered users.
    4. Only registered member will be able to upload their property details.
    5. Unregistered member can search property and also to filter property by prices, sort by A-Z and also search by categories.
    6. Only administrator will have all privileges.

# **Constraints**

* + 1. **H/w Constraints**

Processor : Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz Memory : 8 GB

Hard Disk : 1TB+500GB SSD

Input Device : Keyboard, Mouse

Output Device : Monitor

# **S/w Constraints**

Operating System: Windows 10

Software : Php, MySQL Browser : All Browsers

* 1. **Advantages**
* Easily find any property details at home.
* More reach to members.
* Low operational costs and better quality of services.
* No need of physical company set-ups.
* Easy to start and manage a business.
* member can easily select property from different providers without moving around physically.

# **Limitations**

* Customer can only find properties which resides in India.

# **3. ENVIRONMENT** **SPECIFICATION**

* 1. **Hardware & Software Requirements**

Frontend : HTML, CSS, BootStrap5, JavaScript

Backend : PHP,MySQL

# **Development Description**

* The goal of developing this society management application is to manage

society property easier through India. Using these application admin can

manage member, property. member needs to add the property which he/she

wants to sells/rent. Visitor Customer can search the property and get contact

details of their desire property.

# **4. SYSTEM PLANING**

* 1. **Feasibility Study**
* Feasibility study plays very important role in the development of any system, but when

it is the case of development of any software then its importance increases much

more because in the case one should be very clear about availability of the time

and resources.

* The feasibility of our project has been judged on the basis of time, technology,

resources available and project length.

* **Technology:**

* + This project is developed with PHP, MySQL as database.
  + online Tutorials and other resources on internet might help to much extend.

*** Time:**

* + We have three months for developing entire system. It is less sufficient time for developing such relatively more complex application.

*** Resource:**

* + Resources are the basic thing for any software development.
  + We required good knowledge of software engineering and web-development.
  + We used proper development tools and application server to develop the project.

# **Risk Analysis**

Risk analysis for Ambika row house can be defined as an organized effort to identify, assess, and reduce risks appropriate to member and administrator. Risk analysis in its best form may be to use it in a pro- activemanner in identifying and managing the risks. Pro-active risks include server crash down because of crashing of server all data related to employee will be lost if not saved properly. So, to overcome this proactive risk, all data must be stored and saved properly. However, in case an incident has happened; after the event handling, it should still be tackled in line with the risk analysis principles. For reactive risks there is risk of security is there as anyone who came to know about the administrator login id and password, can make changes in the property, member details, amenities, In case any such situation occurs the initial data storage must be maintained somewhere to undo to those changes.

# **Project Schedule**

* + 1. **Task Dependency**
       - Dependency between upload property and admin member upload

the product but until admin doesn't approve the property that property

will not be displayed on member site.

* + - * Dependency between Admin and member Admin can active/block

Member based on their behavior.

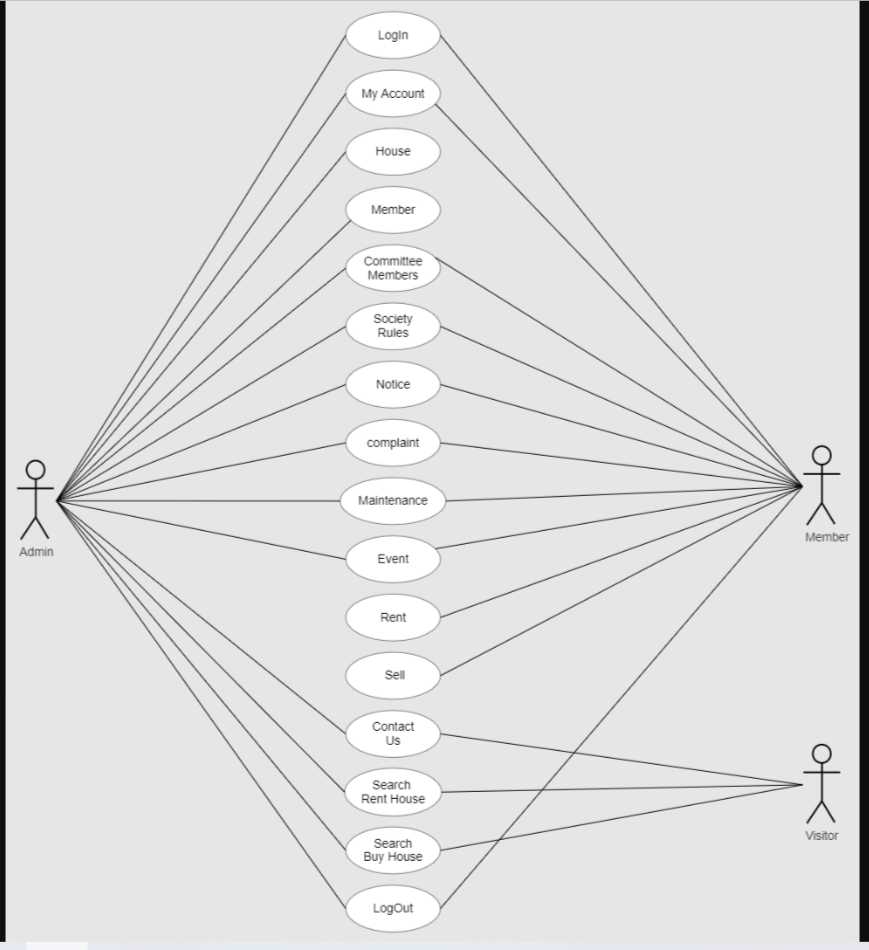
**5. SYSTEM ANALYSIS**

* 1. **Software Requirement Specification**

1. Registration Process
   * Member
     + Member needs to register their account by specific fields Email, Password and name, Mobile Number.
     + Login Module for admin, member login from the frontend.
   * Admin
     + Admin can block the member
2. Profile Page
   * Member
     + After login customer can view their profile page.
     + Profile page has following options like My Profile, Property and Others.
   * Admin
     + After login admin can view their profile page.
     + admin can edit the profile details.
     + admin can upload and manage their property details.
3. User Management
   * Admin
     + Admin can view all the member list.
     + Admin can block member and seller.
     + Admin can view all the member list.
4. Property Catalogue/Catalogue Management (Customer)
   * The page is designed with the list of property.
   * Each property is shown with the image and basic information like (property name, price, address).
   * Property is sorted by
     + Price (Low, Medium, High)

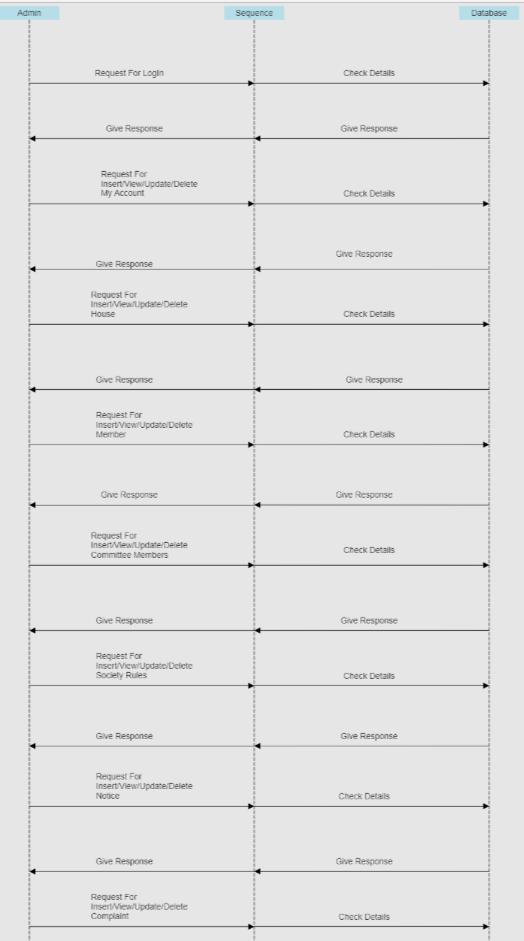
# **UML Diagram**

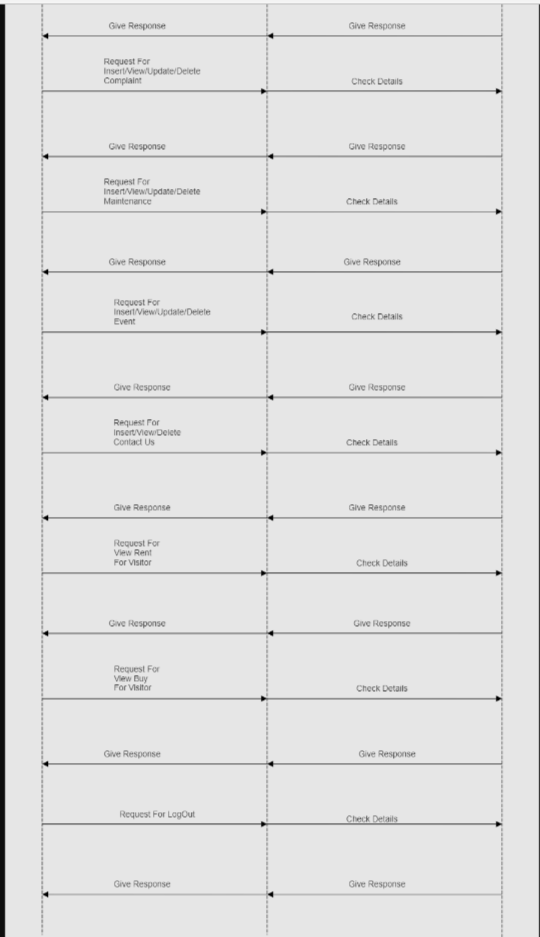
* + 1. **Use case Diagram**



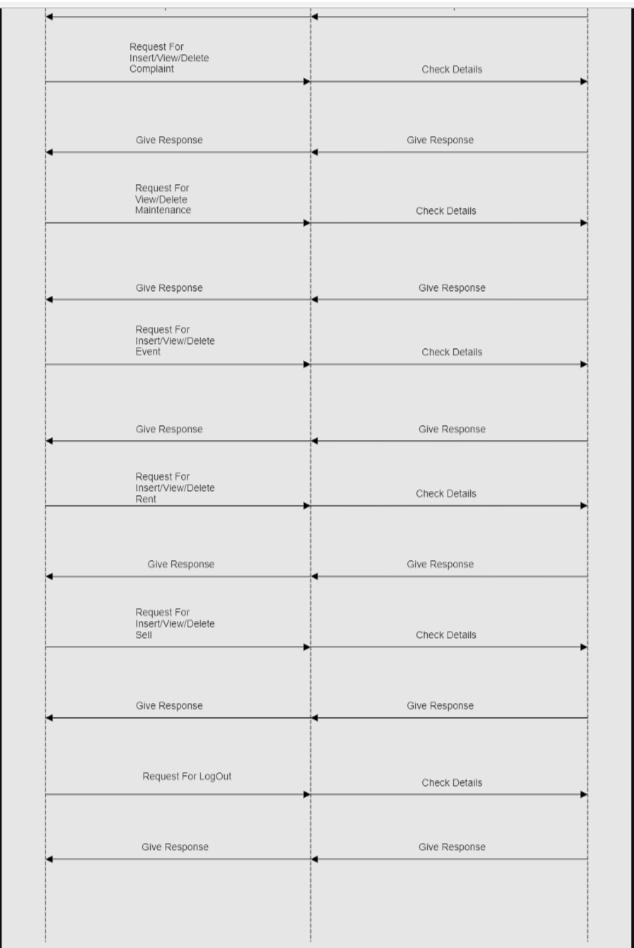
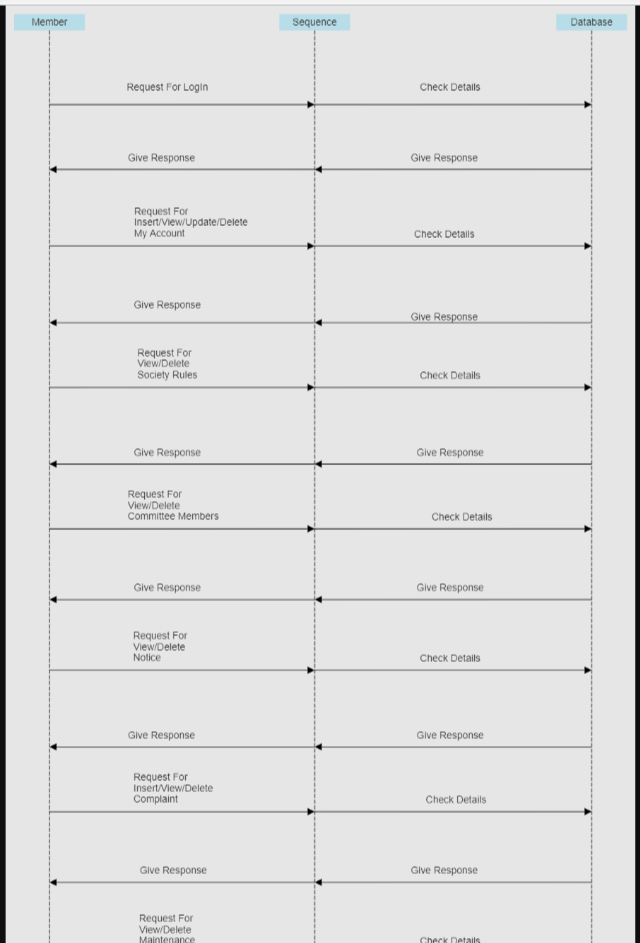
* + 1. **Sequence Diagram**

**Admin Sequence Diagram**

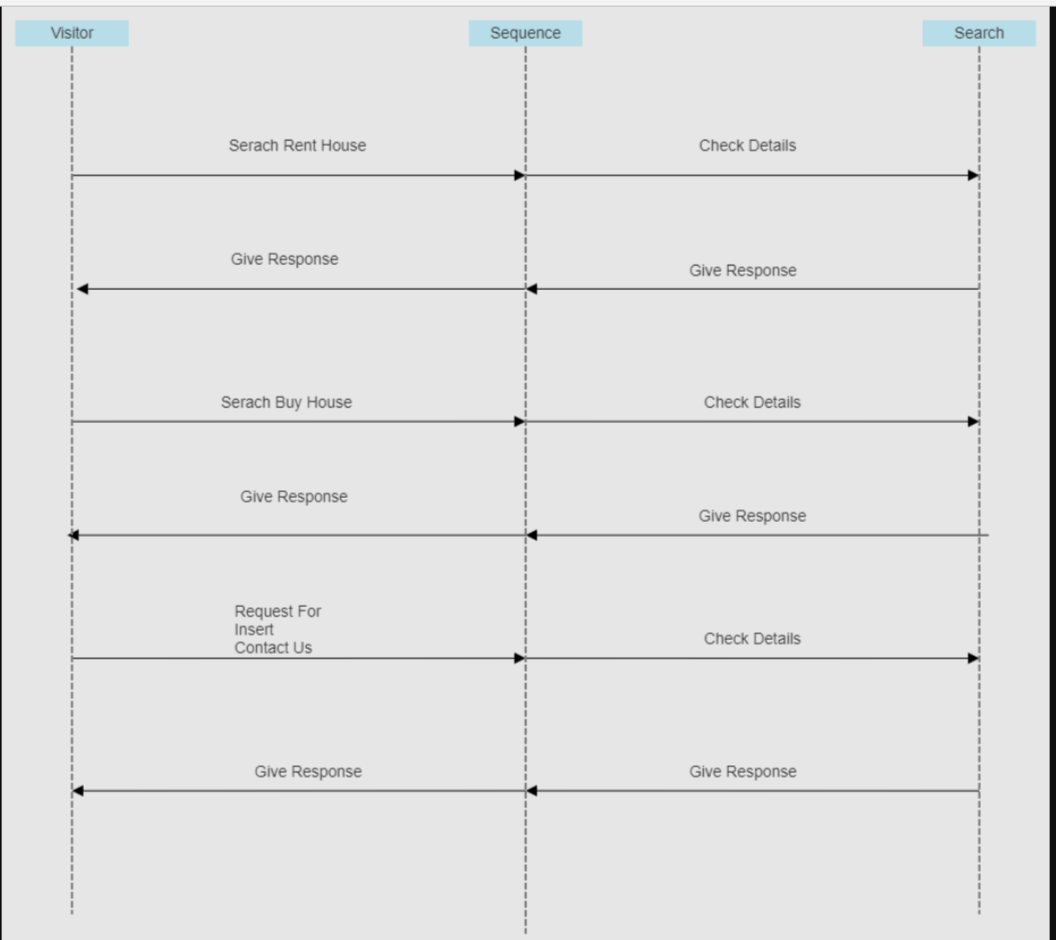




**Member Sequence Diagram**

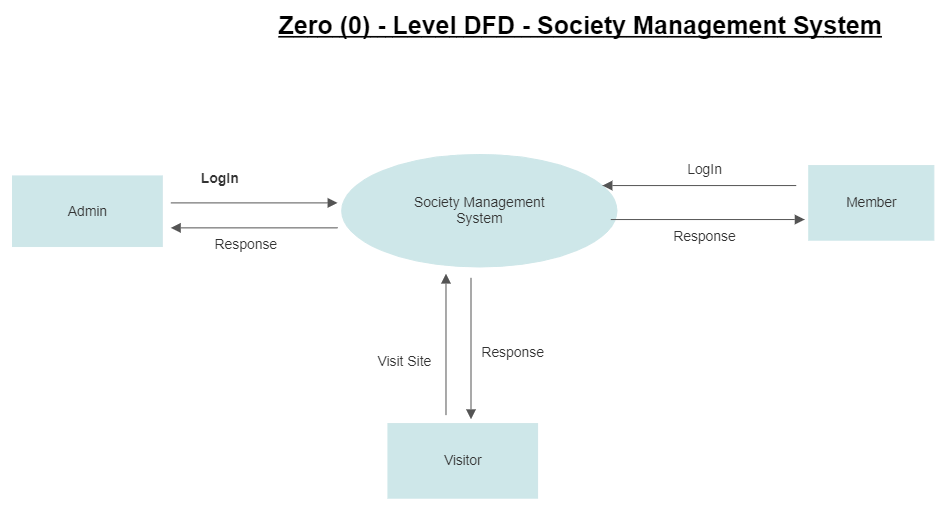


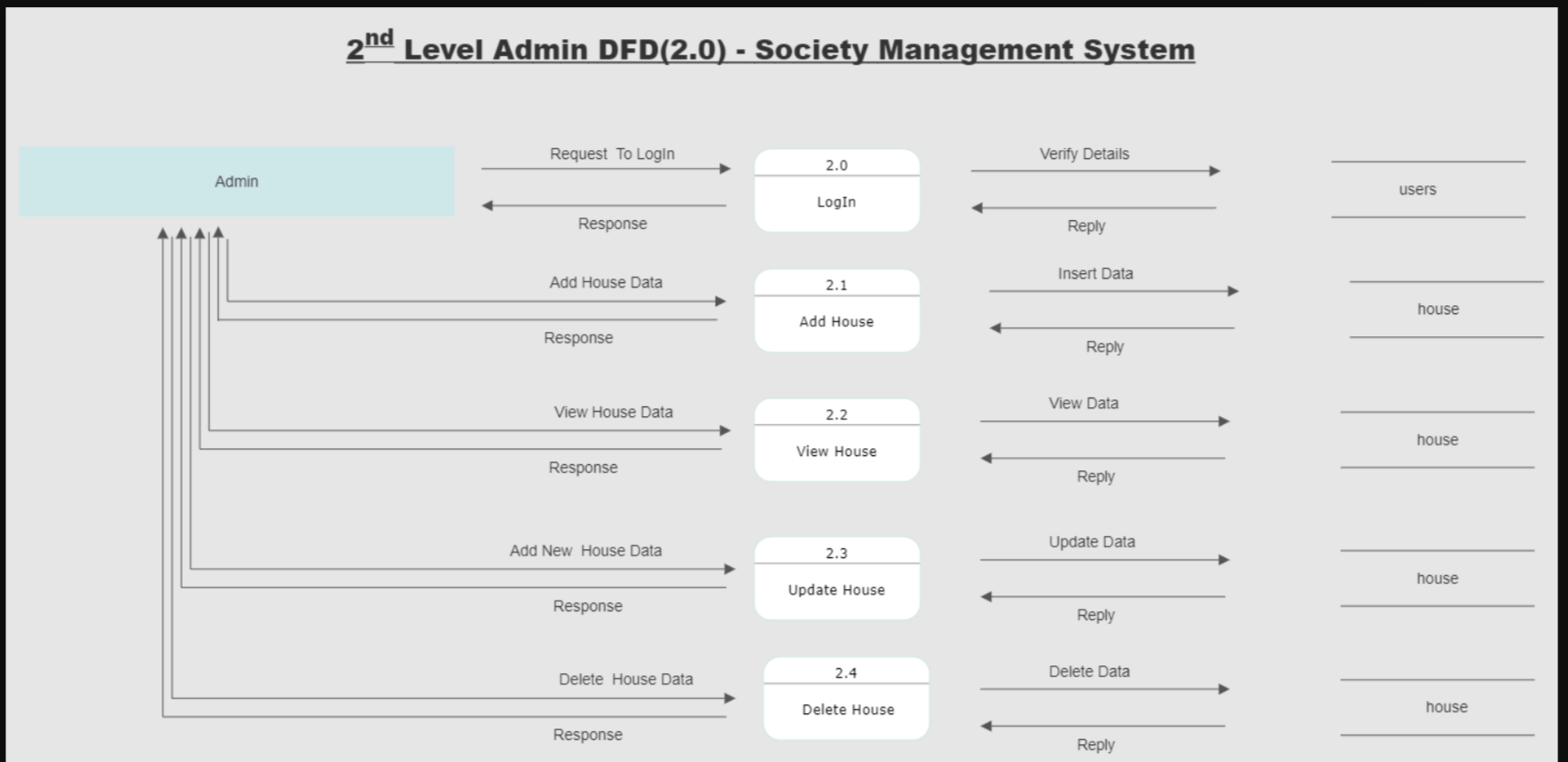
**Visitor Sequence Diagram**

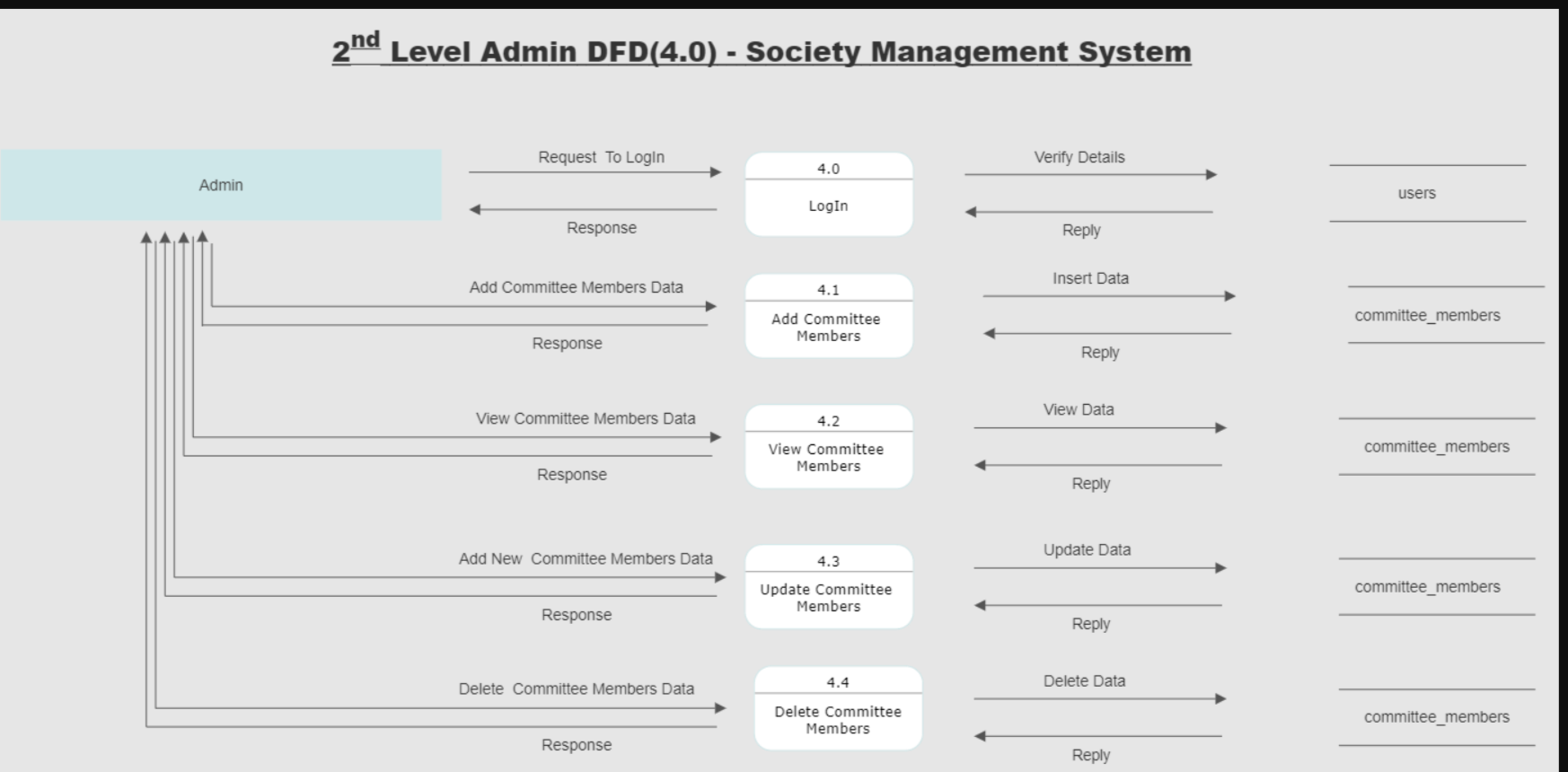
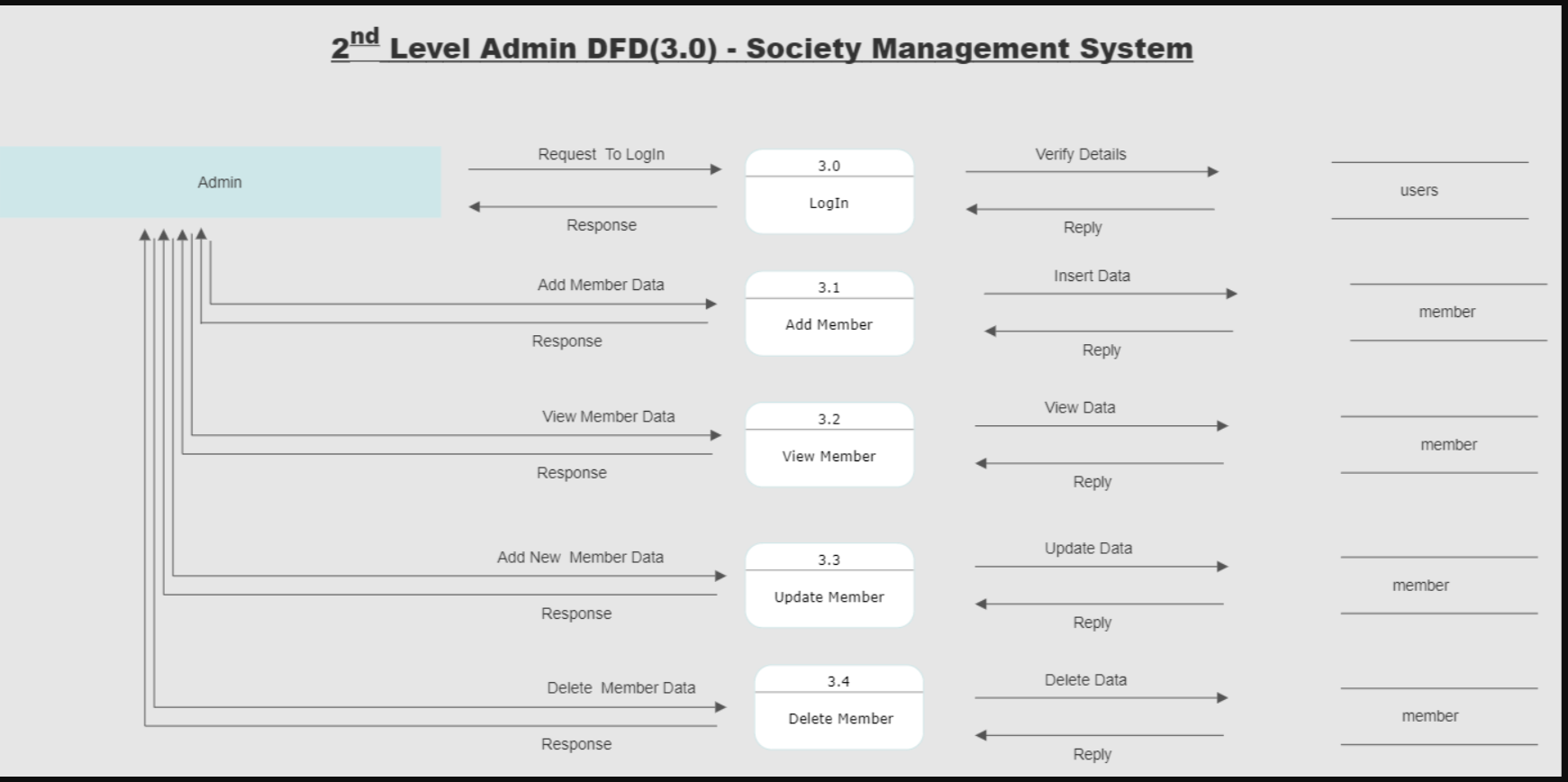


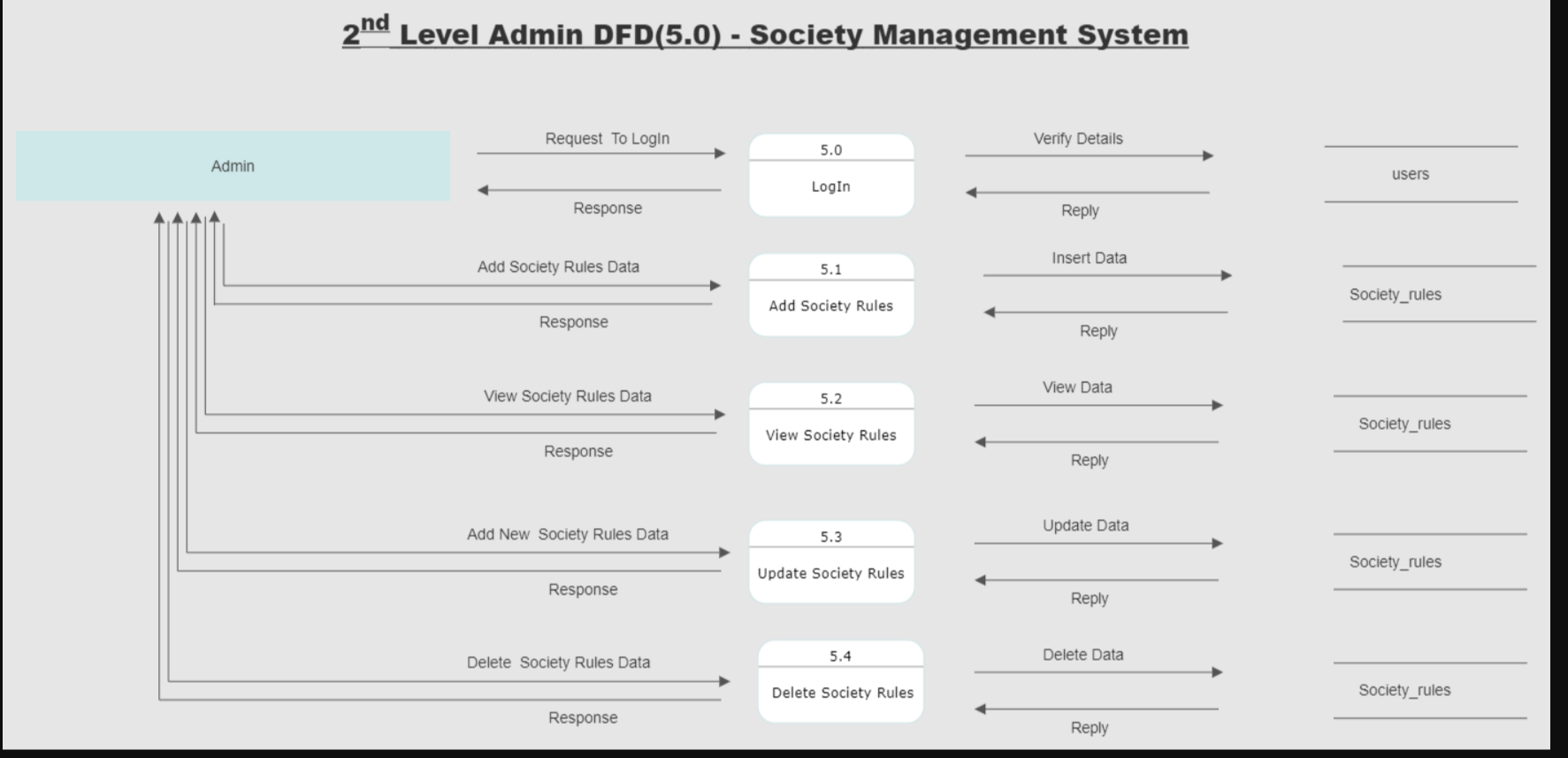
* + 1. **DFD Diagram**

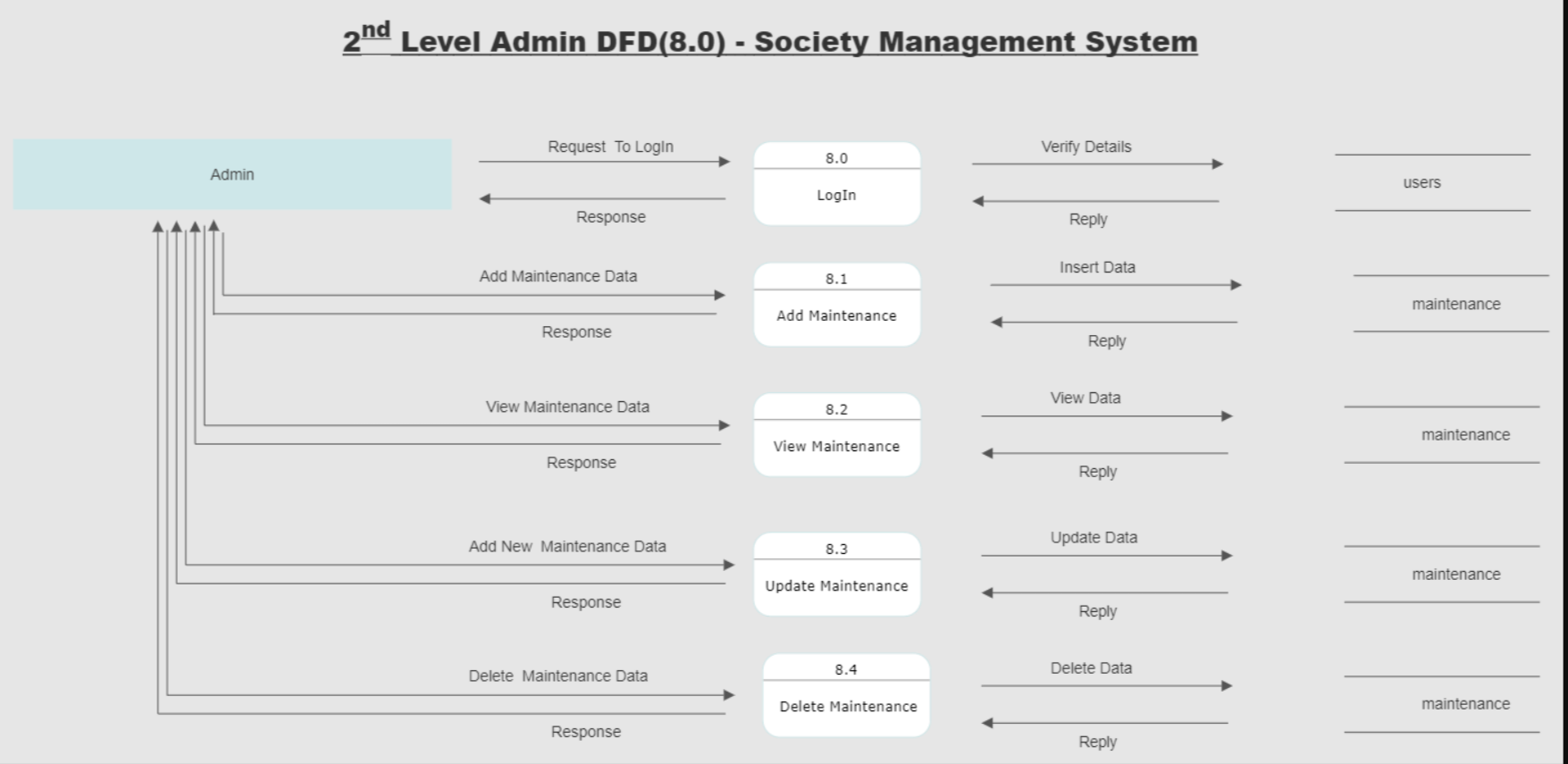
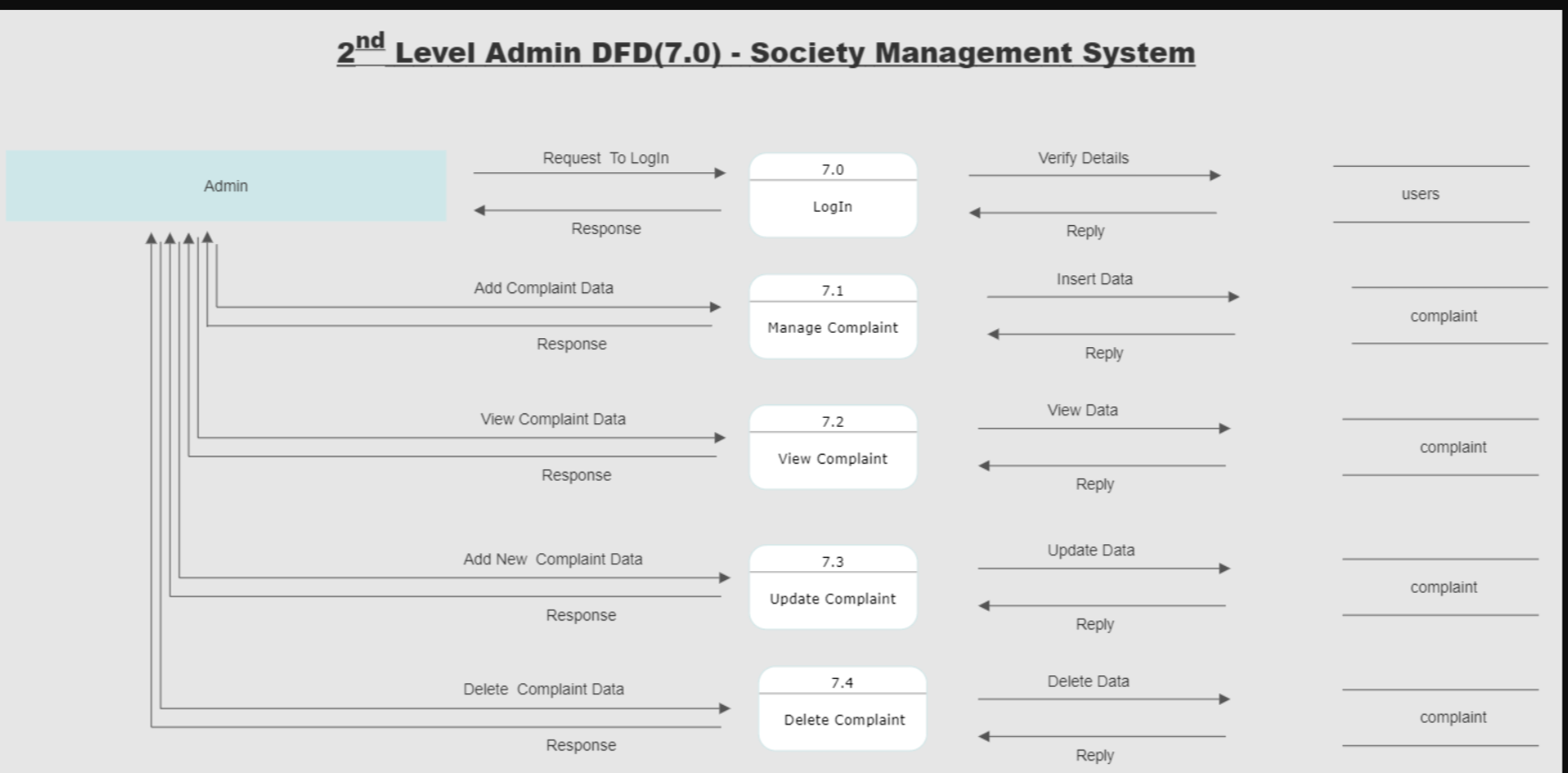
**Admin DFD**

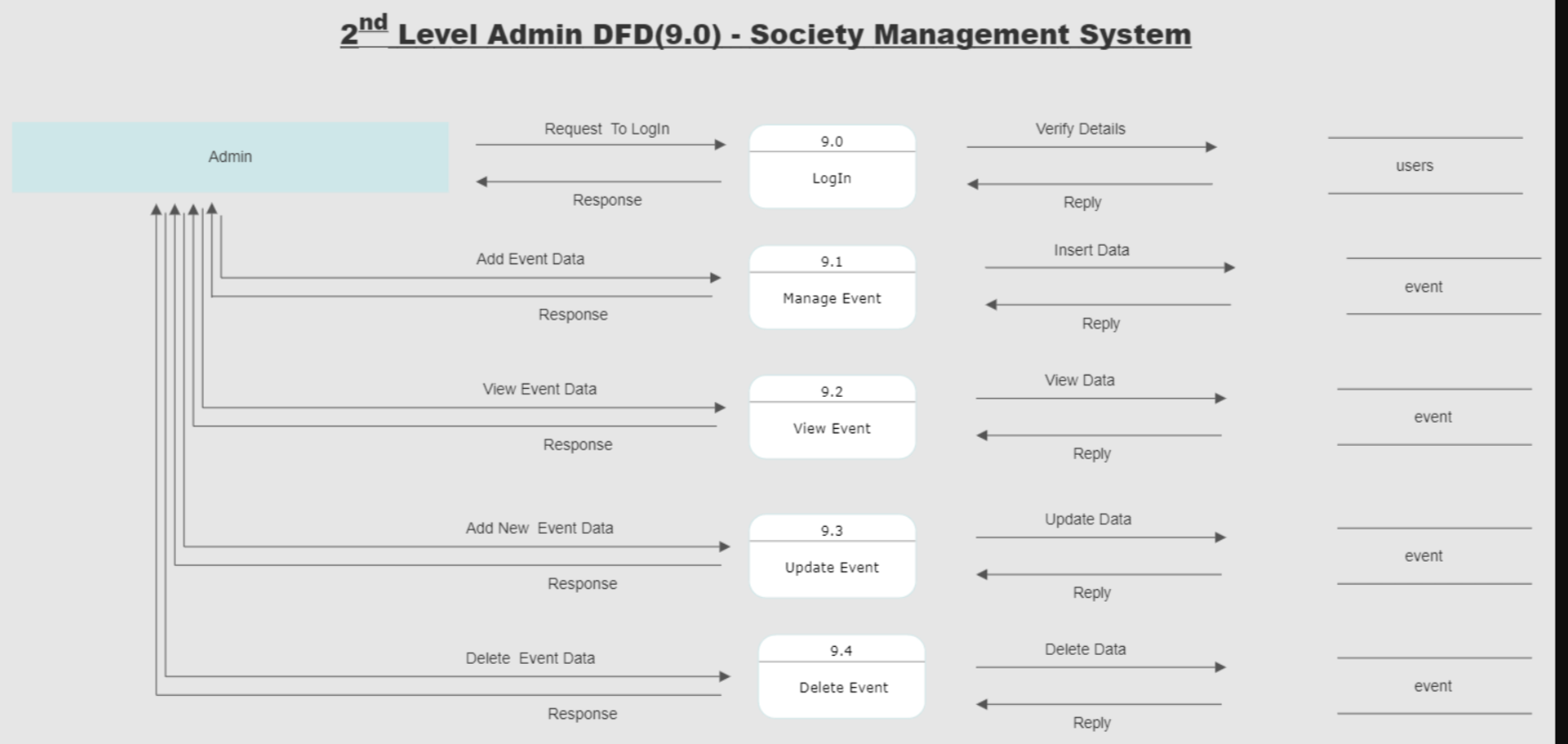


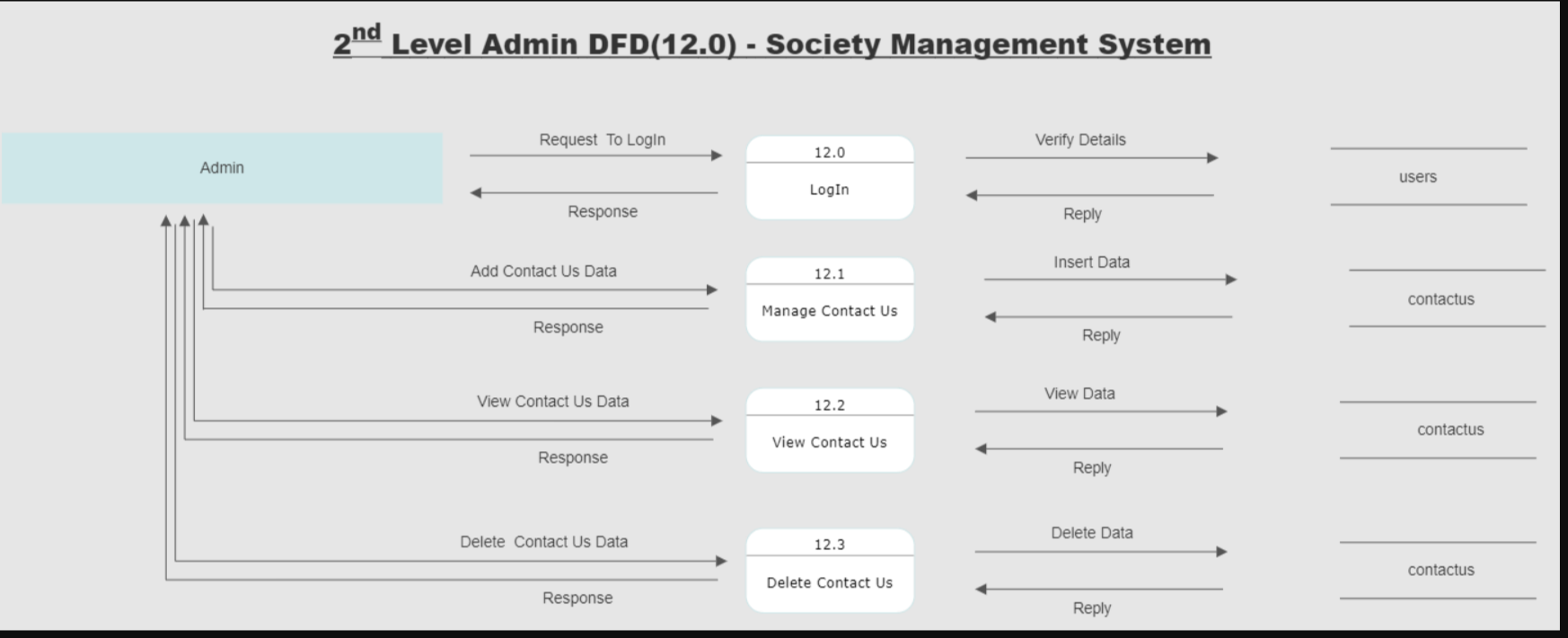
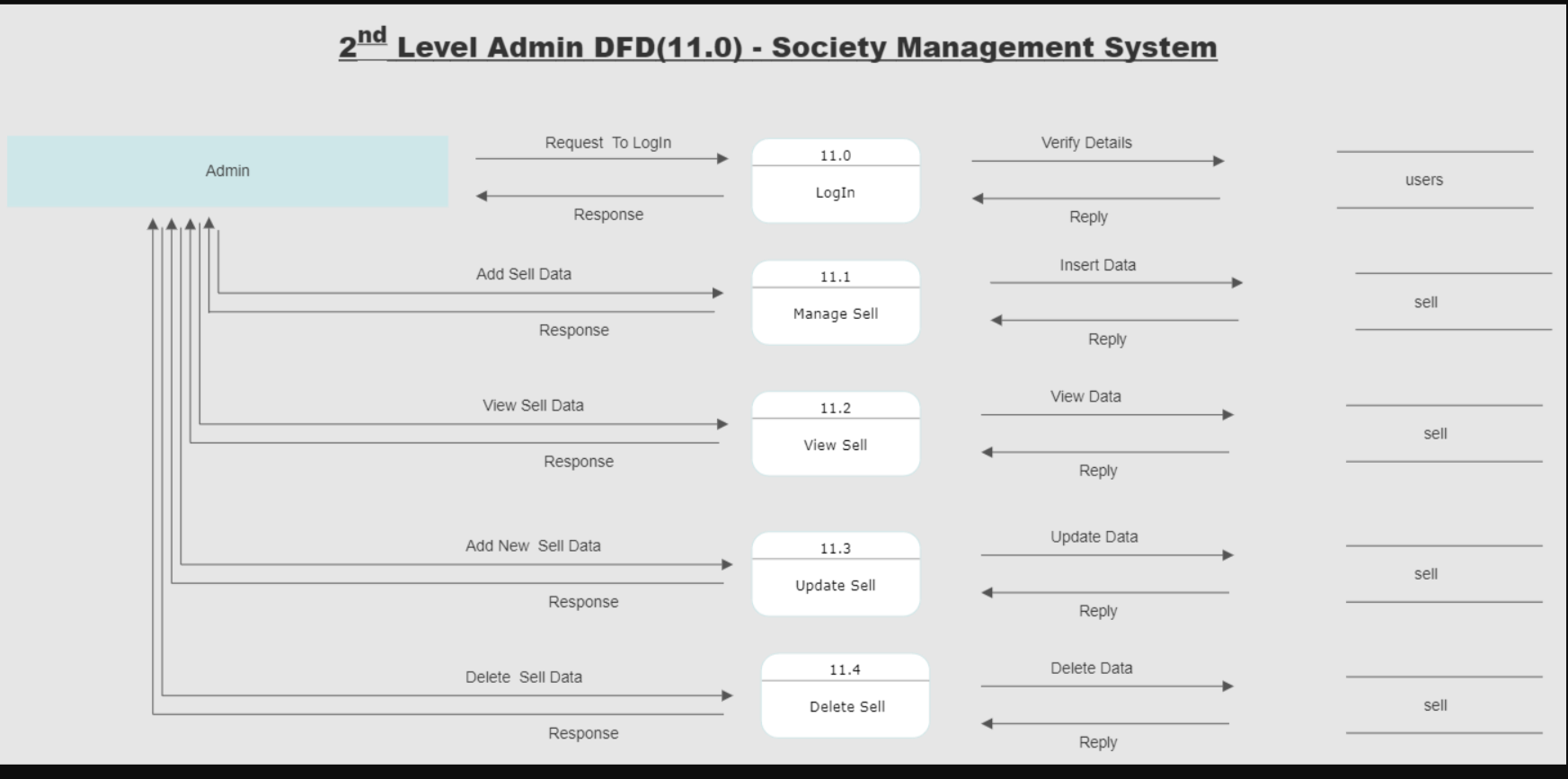




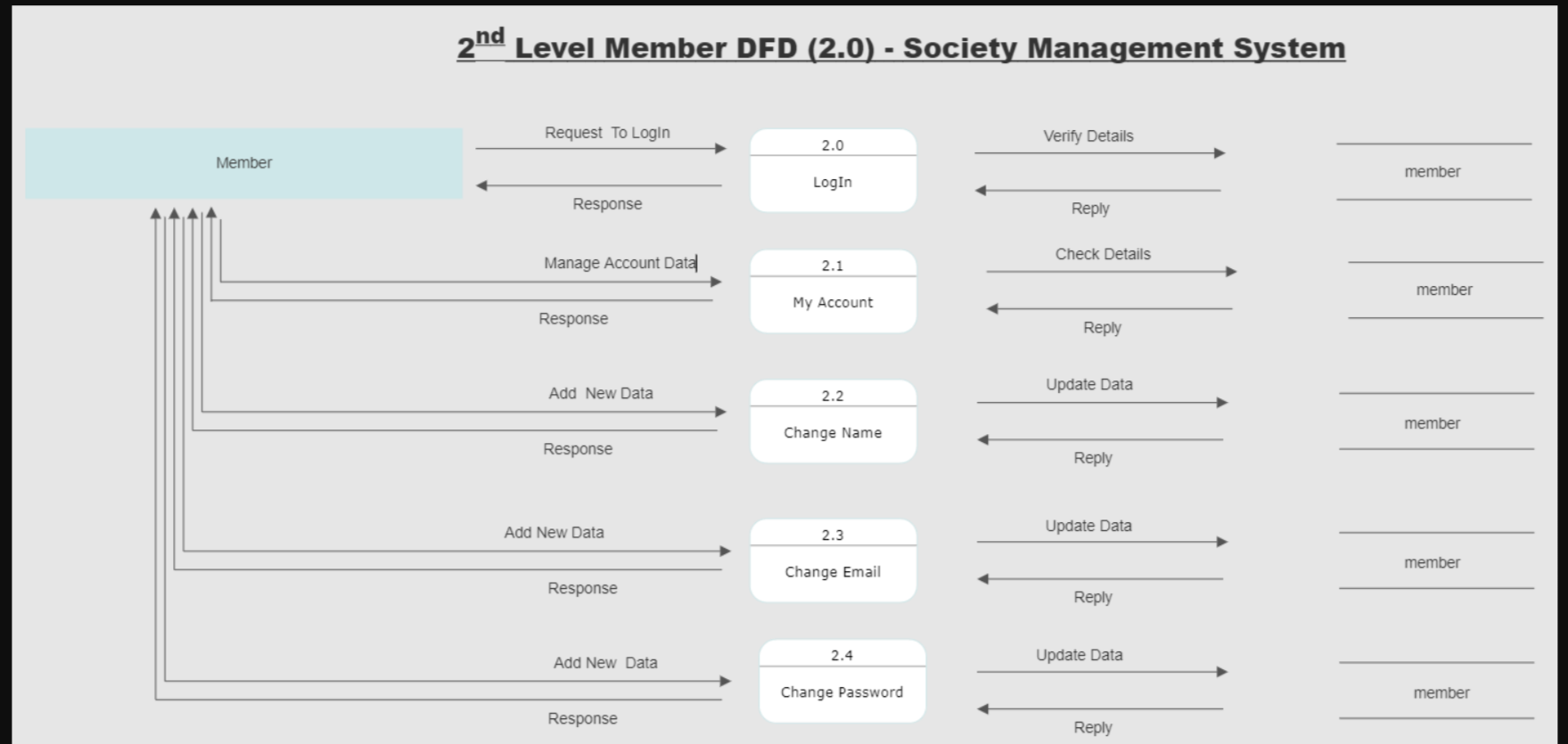


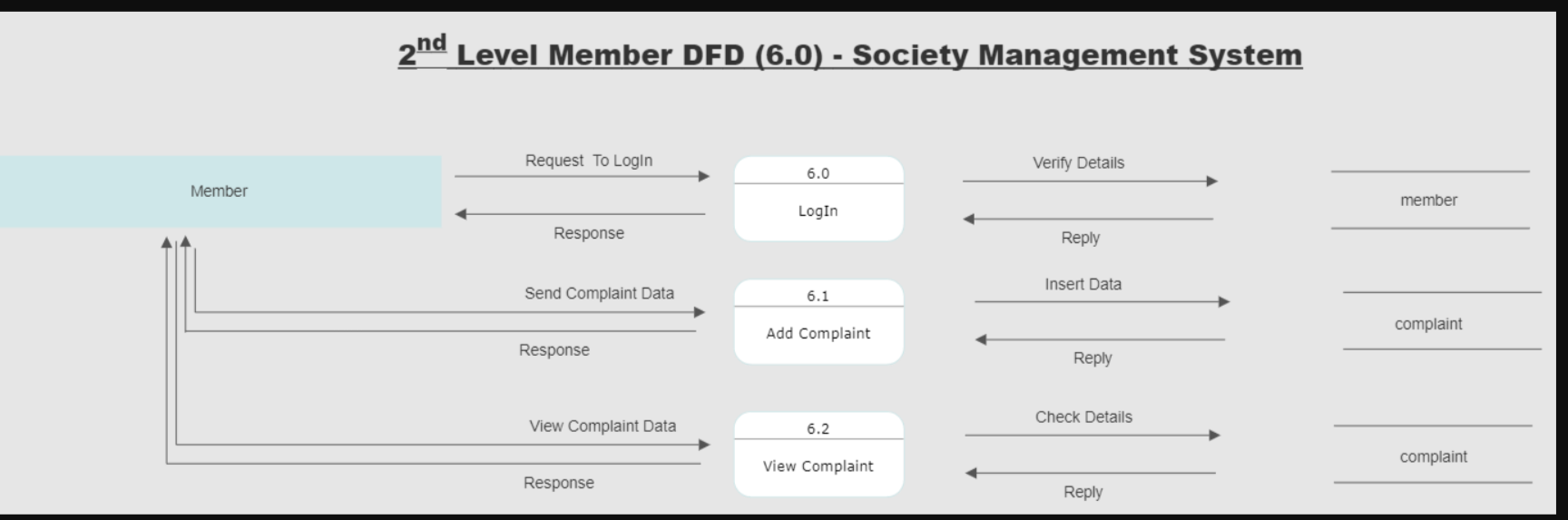


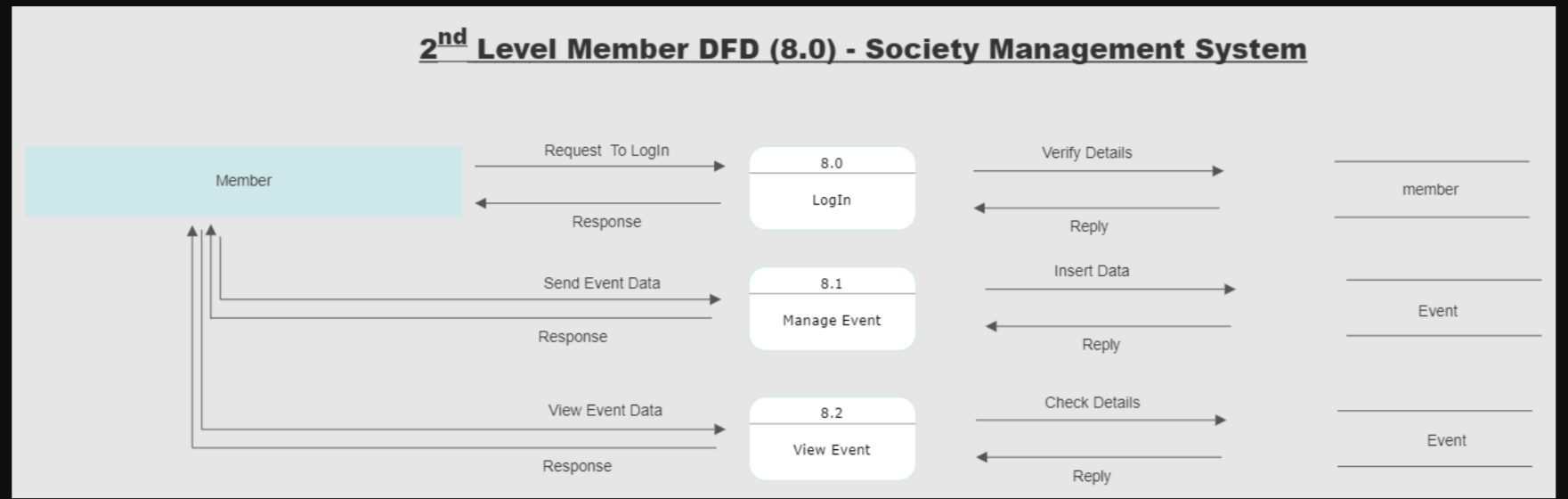


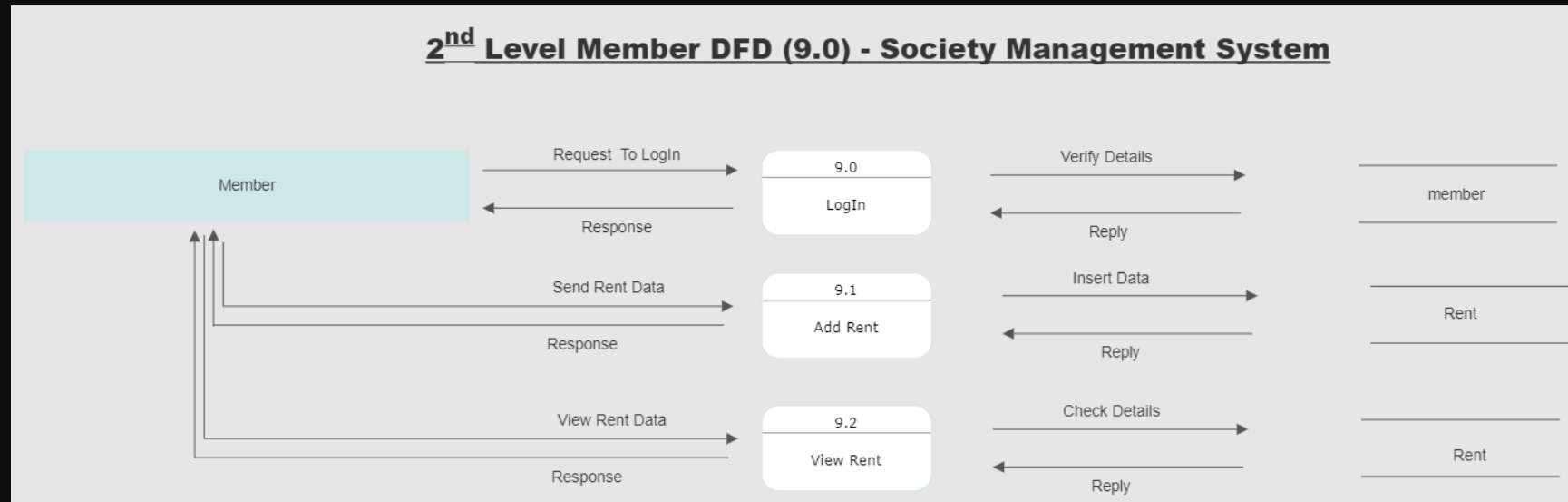


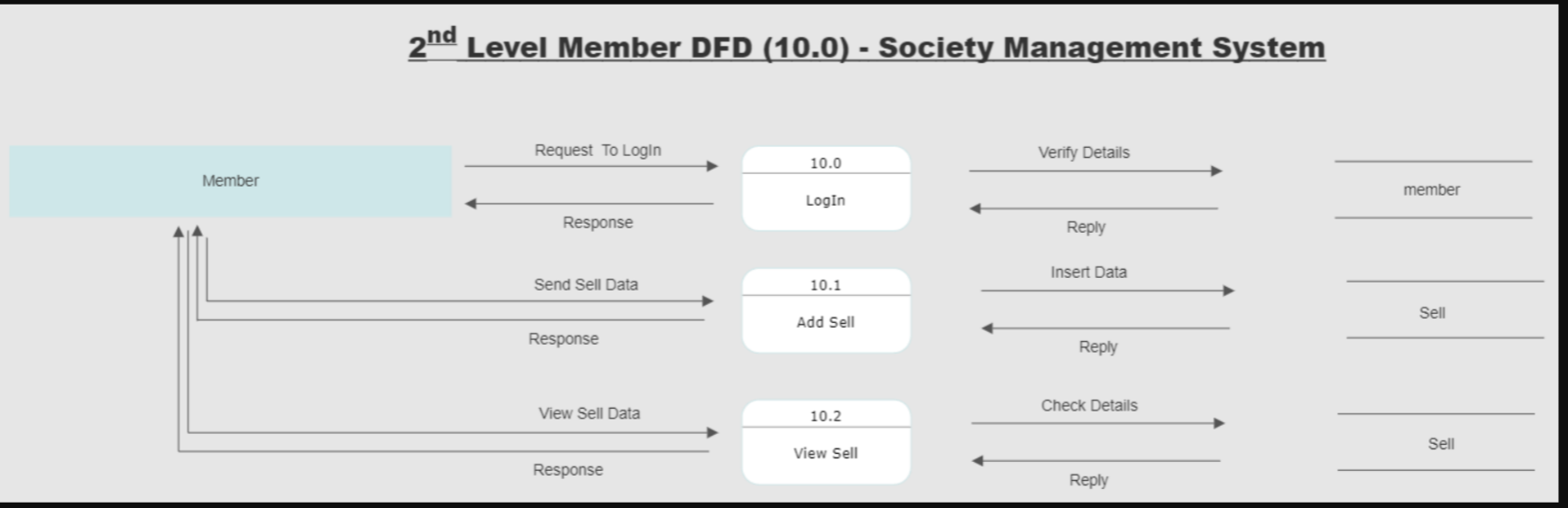
**Member DFD**



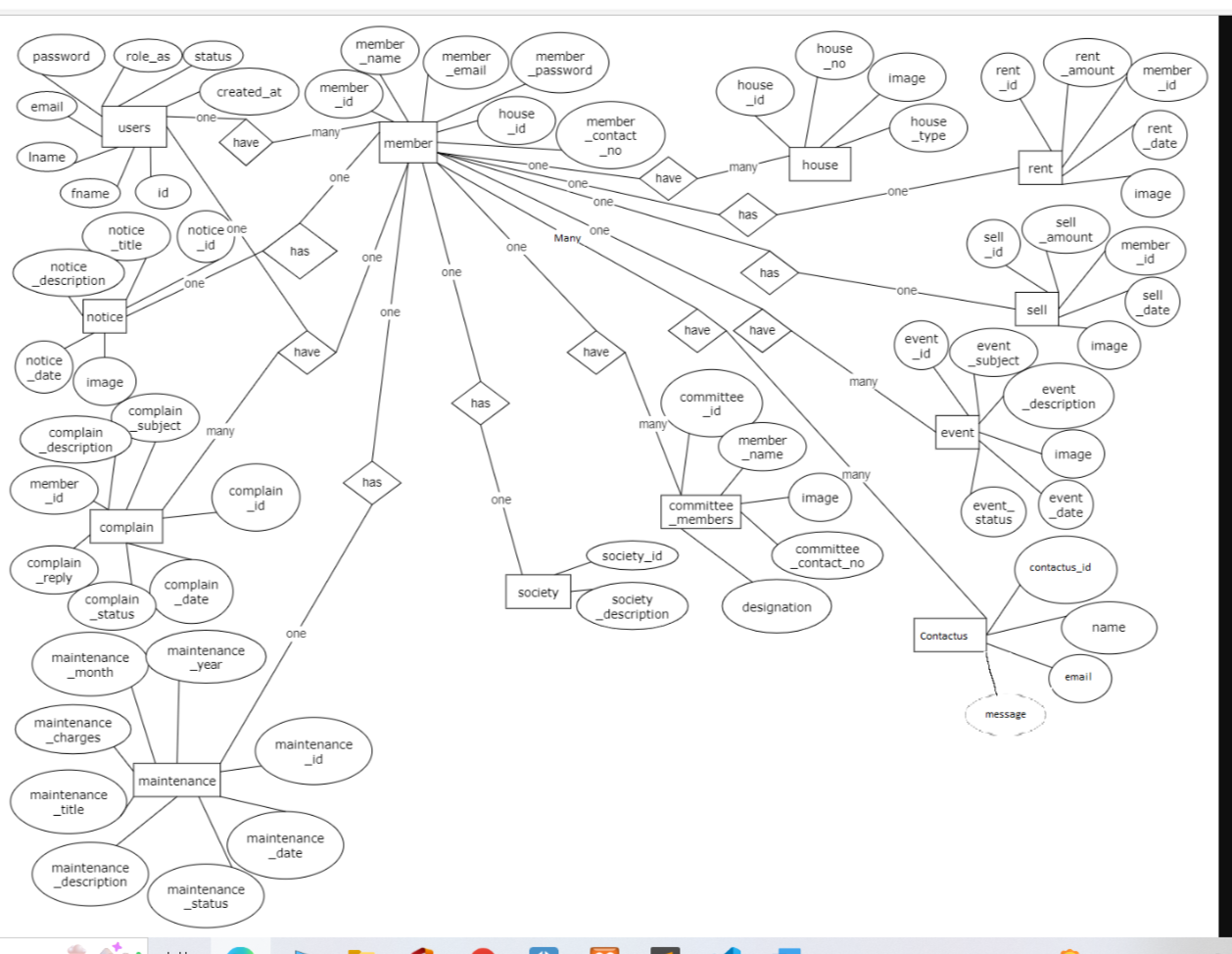








**5.3 ER Diagram**



1. **SOFTWARE DESIGN**
   1. Database Design
2. **Users**

|  |  |  |  |
| --- | --- | --- | --- |
| **Element Name** | **Element Description** | **Data types** | **Constraint** |
| id | Id | int (11) | Primary Key |
| fname | fname | varchar (200) | Not Null |
| lname | lname | varchar (200) | Not Null |
| email | email | varchar (200) | Not Null |
| password | password | varchar (200) | Not Null |
| role\_as | role\_as | tinyint (4) | As defined  Default  (0) |
| status | status | tinyint (1) | As defined  Default  (0) |
| created\_at | created\_at | timestamp | current\_timestamp () |

1. **house**

|  |  |  |  |
| --- | --- | --- | --- |
| **Element Name** | **Element Description** | **Data Types** | **Constraint** |
| house\_id | house\_id | int (11) | Primary Key |
| house\_no | House\_no | varchar (200) | Not Null |
| image | Image | varchar (200) | Not Null |
| House\_type | House\_type | varchar (200) | Not Null |

1. **member**

|  |  |  |  |
| --- | --- | --- | --- |
| **Element Name** | **Element Description** | **Data types** | **Constraint** |
| member\_id | member\_id | int (11) | Primary Key |
| member\_name | member\_name | varchar (200) | Not Null |
| member\_email | member\_email | varchar (200) | Not Null |
| member\_password | member\_password | varchar (200) | Not Null |
| member\_contact\_no | member\_contact\_no | varchar (200) | Not Null |
| house\_id | house\_id | int (11) | Foreign key |

1. **rent**

|  |  |  |  |
| --- | --- | --- | --- |
| **Element Name** | **Element Description** | **Data types** | **Constraint** |
| rent\_id | rent\_id | int (11) | Primary Key |
| rent\_amount | rent\_amount | int (11) | Not Null |
| member\_id | member\_id | int (11) | Foreign Key |
| rent\_date | rent\_date | datetime | Not Null |
| image | Image | Varchar (200) | Not Null |

1. **sell**

|  |  |  |  |
| --- | --- | --- | --- |
| **Element Name** | **Element Description** | **Data types** | **Constraint** |
| sell\_id | sell\_id | int (11) | Primary Key |
| sell\_amount | sell\_amount | int (11) | Not Null |
| member\_id | member\_id | int (11) | Foreign Key |
| sell\_date | sell\_date | datetime | Not Null |
| image | Image | varchar (200) | Not Null |

1. **notice**

|  |  |  |  |
| --- | --- | --- | --- |
| **Element Name** | **Element Description** | **Data types** | **Constraint** |
| notice\_id | notice\_id | int (11) | Primary Key |
| notice\_title | notice\_title | varchar (200) | Not Null |
| notice\_description | notice\_description | mediumtext (200) | Not Null |
| image | Image | varchar (200) | Not Null |
| notice\_date | notice\_date | datetime | Not Null |

1. **event**

|  |  |  |  |
| --- | --- | --- | --- |
| **Element Name** | **Element Description** | **Data types** | **Constraint** |
| event\_id | event\_id | int (11) | Primary Key |
| event\_subject | event\_subject | varchar (200) | Not Null |
| event\_description | event\_description | mediumtext (500) | Not Null |
| Image | Image | varchar (200) | Not Null |
| event\_date | event\_date | datetime | Not Null |
| event\_status | event\_status | varchar (200) | As defined  Default  (pending) |

1. **committee members**

|  |  |  |  |
| --- | --- | --- | --- |
| **Element Name** | **Element Description** | **Data types** | **Constraint** |
| committee\_id | committee\_id | int (11) | Primary Key |
| member\_name | member\_name | varchar (200) | Not Null |
| image | Image | varchar (200) | Not Null |
| member\_contact\_no | member\_contact\_no | int (11) | Not Null |
| designation | designation | varchar (200) | Not Null |

1. **Society\_rules**

|  |  |  |  |
| --- | --- | --- | --- |
| **Element Name** | **Element Description** | **Data types** | **Constraint** |
| society\_id | society\_id | int (11) | Primary Key |
| society\_description | society\_description | mediumtext (500) | Not Null |

**10. contact us**

|  |  |  |  |
| --- | --- | --- | --- |
| **Element Name** | **Element Description** | **Data types** | **Constraint** |
| contactus\_id | contactus \_id | int (11) | Primary Key |
| name | Name | varchar (200) | Not Null |
| email | Email | varchar (200) | Not Null |
| message | message | varchar (200) | Not Null |

**11.maintenance**

|  |  |  |  |
| --- | --- | --- | --- |
| **Element Name** | **Element Description** | **Data types** | **Constraint** |
| maintenance\_id | maintenance\_id | int (11) | Primary Key |
| maintenance\_year | maintenance\_year | datetime | Not Null |
| maintenance\_month | maintenance\_month | datetime | Not Null |
| maintenance\_charges | maintenance\_charges | int (11) | Not Null |
| maintenance\_title | maintenance\_title | varchar (200) | Not Null |
| maintenance\_description | maintenance\_ description | mediumtext (500) | Not Null |
| maintenance\_status | maintenance\_status | varchar (200) | As defined  Default  (pending) |
| maintenance\_date | maintenance\_date | datetime | Not Null |

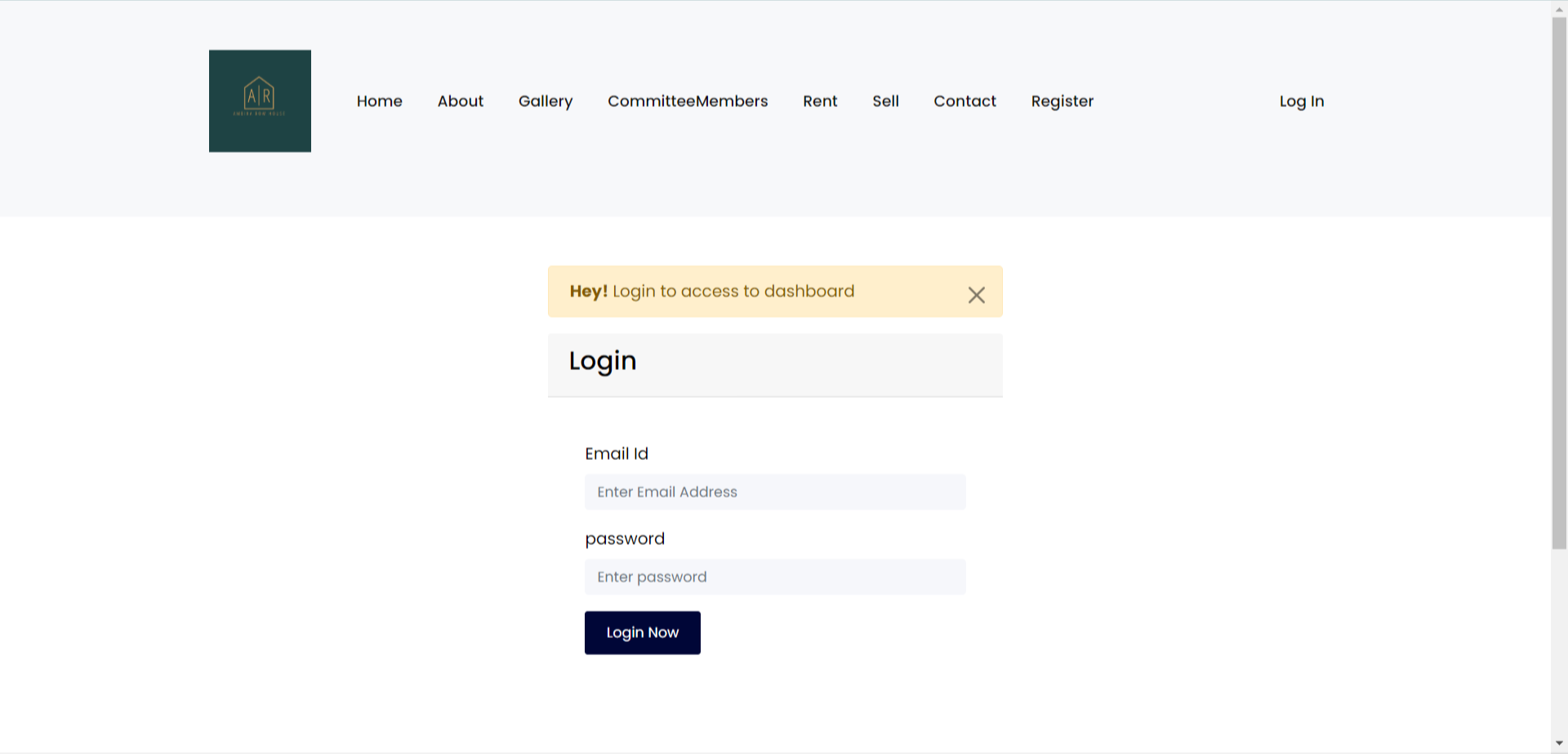
**12.complain**

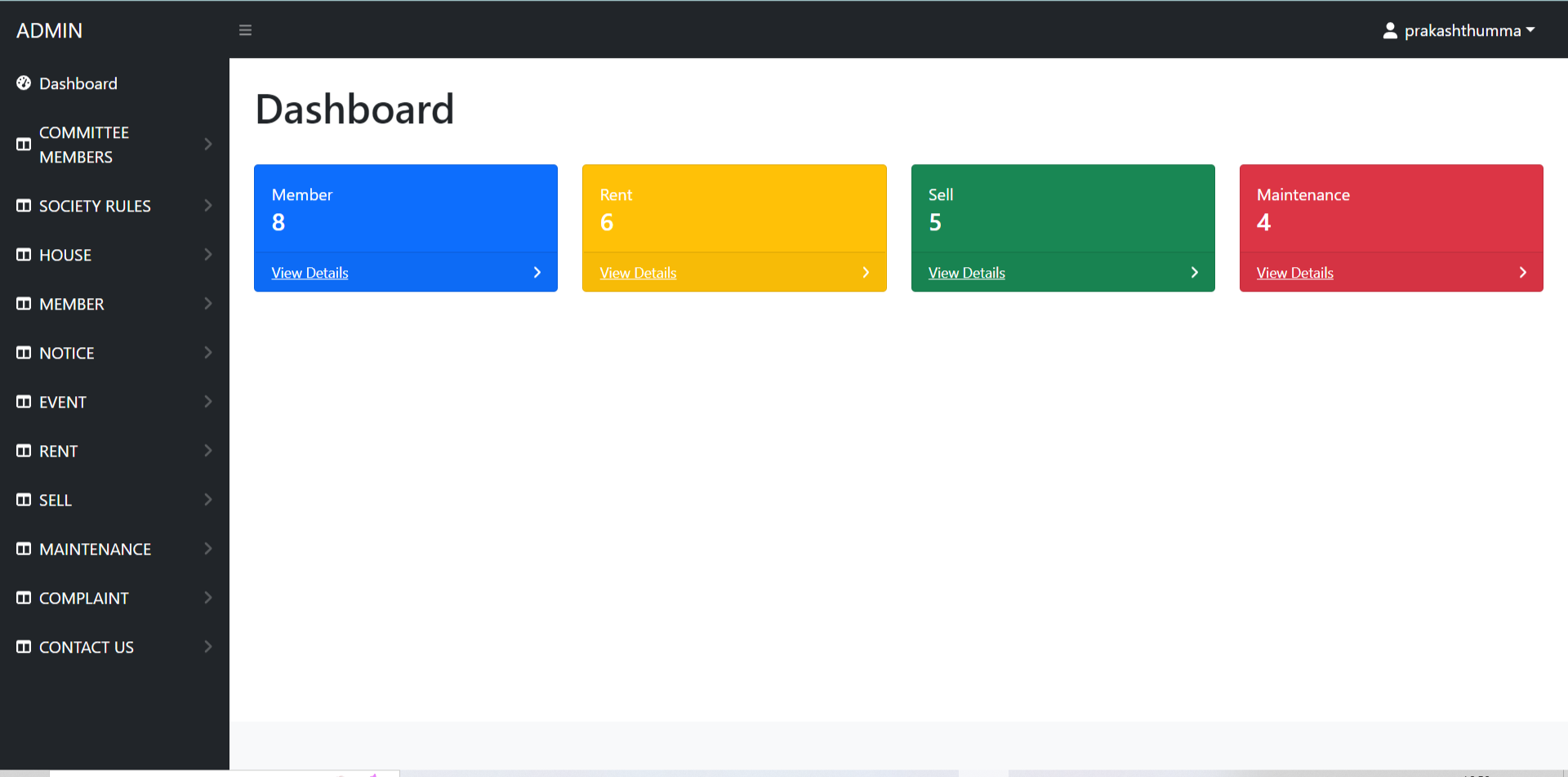
|  |  |  |  |
| --- | --- | --- | --- |
| **Element Name** | **Element Description** | **Data types** | **Constraint** |
| complain\_id | complain\_id | int (11) | Primary Key |
| complain \_subject | complain\_subject | varchar (200) | Not Null |
| complain\_description | complain\_description | mediumtext (500) | Not Null |
| member\_id | member\_id | int (11) | Foreign Key |
| complain\_reply | complain\_reply | mediumtext (500) | Not Null |
| complain\_status | complain\_status | varchar (200) | As defined  Default  (pending) |
| complain\_date | complain\_date | timestamp | current\_timestamp () |

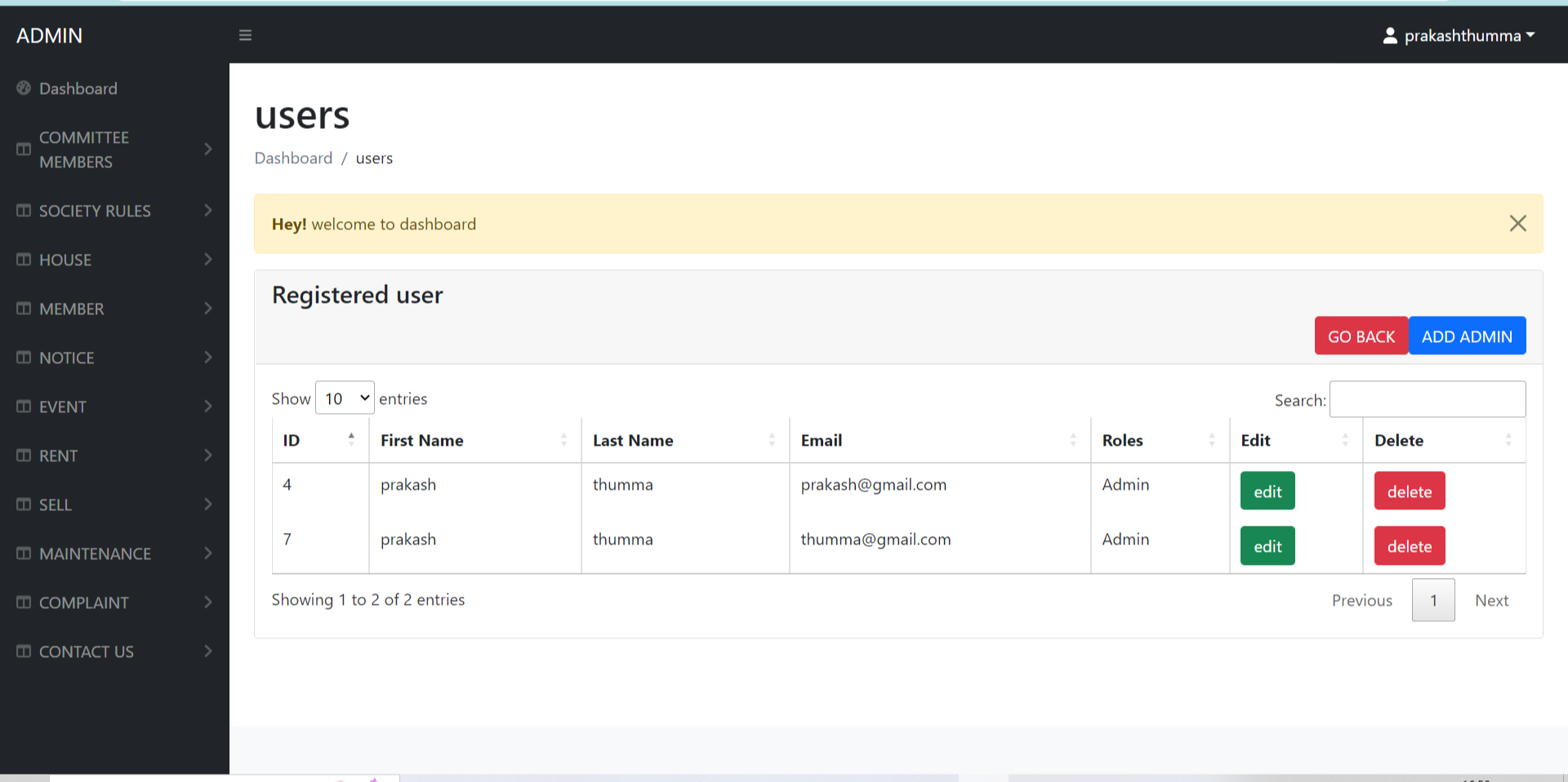
# **Interface Design**

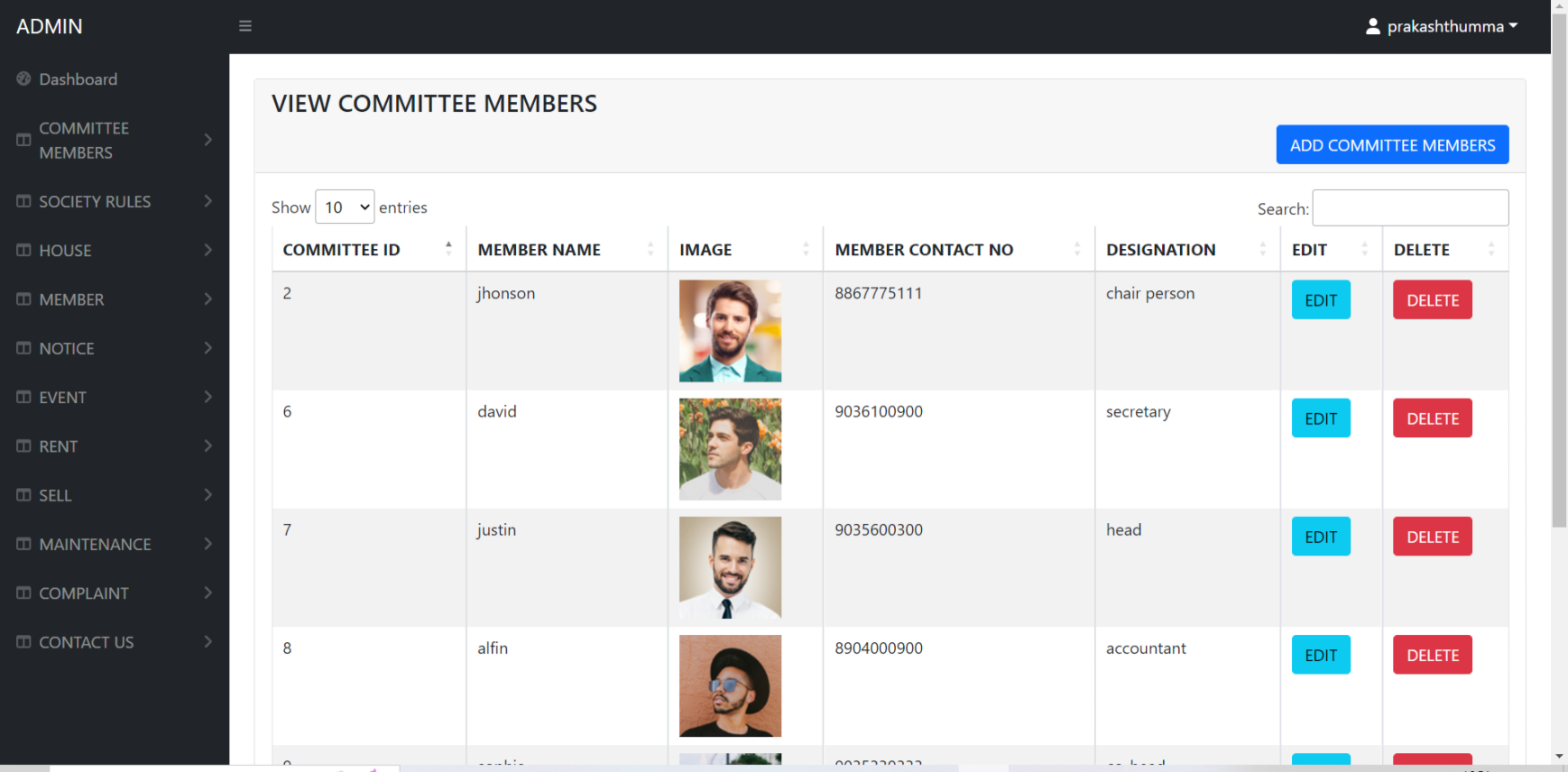
**Admin Panel**

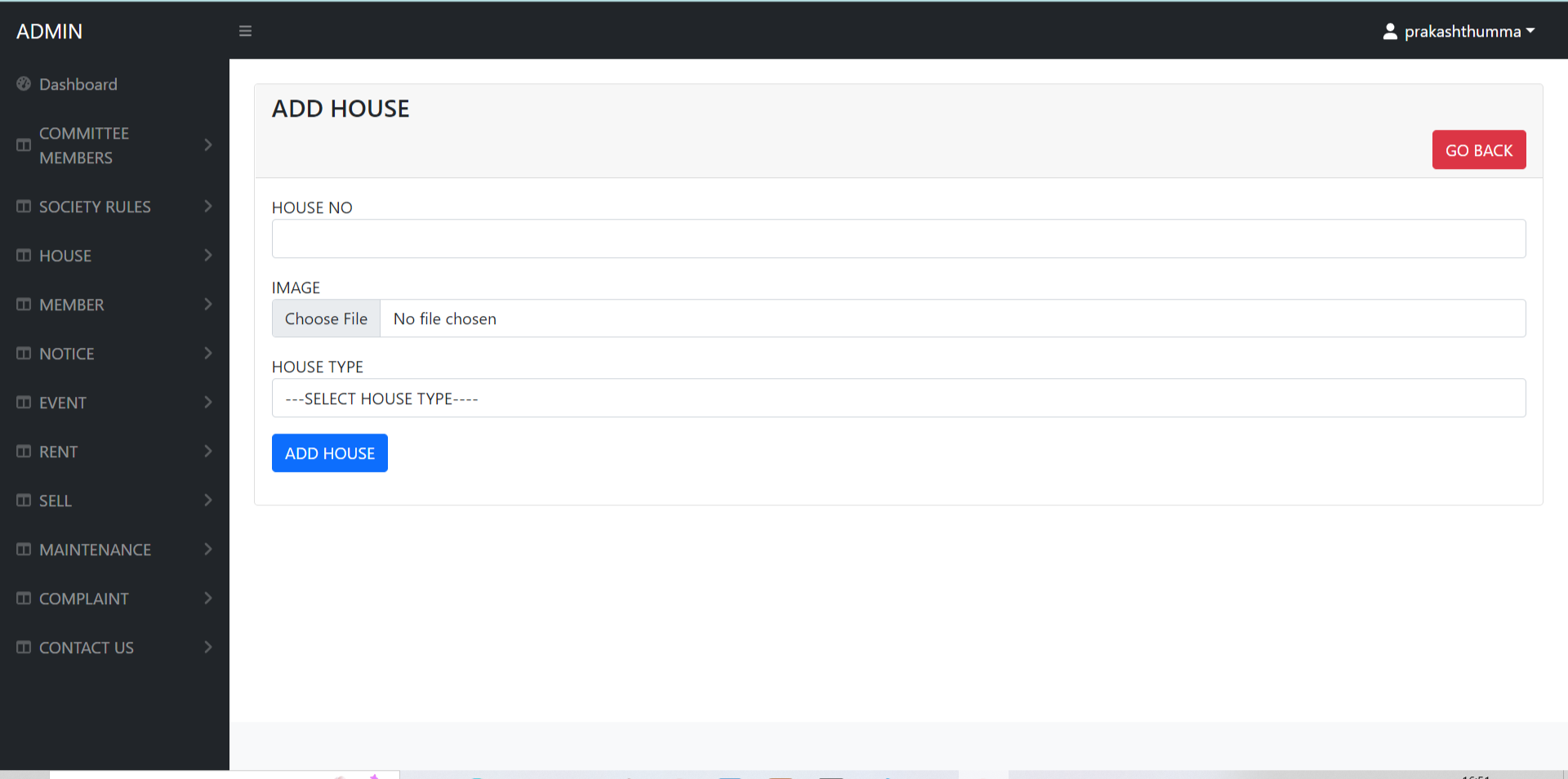
**Admin Dashboard**

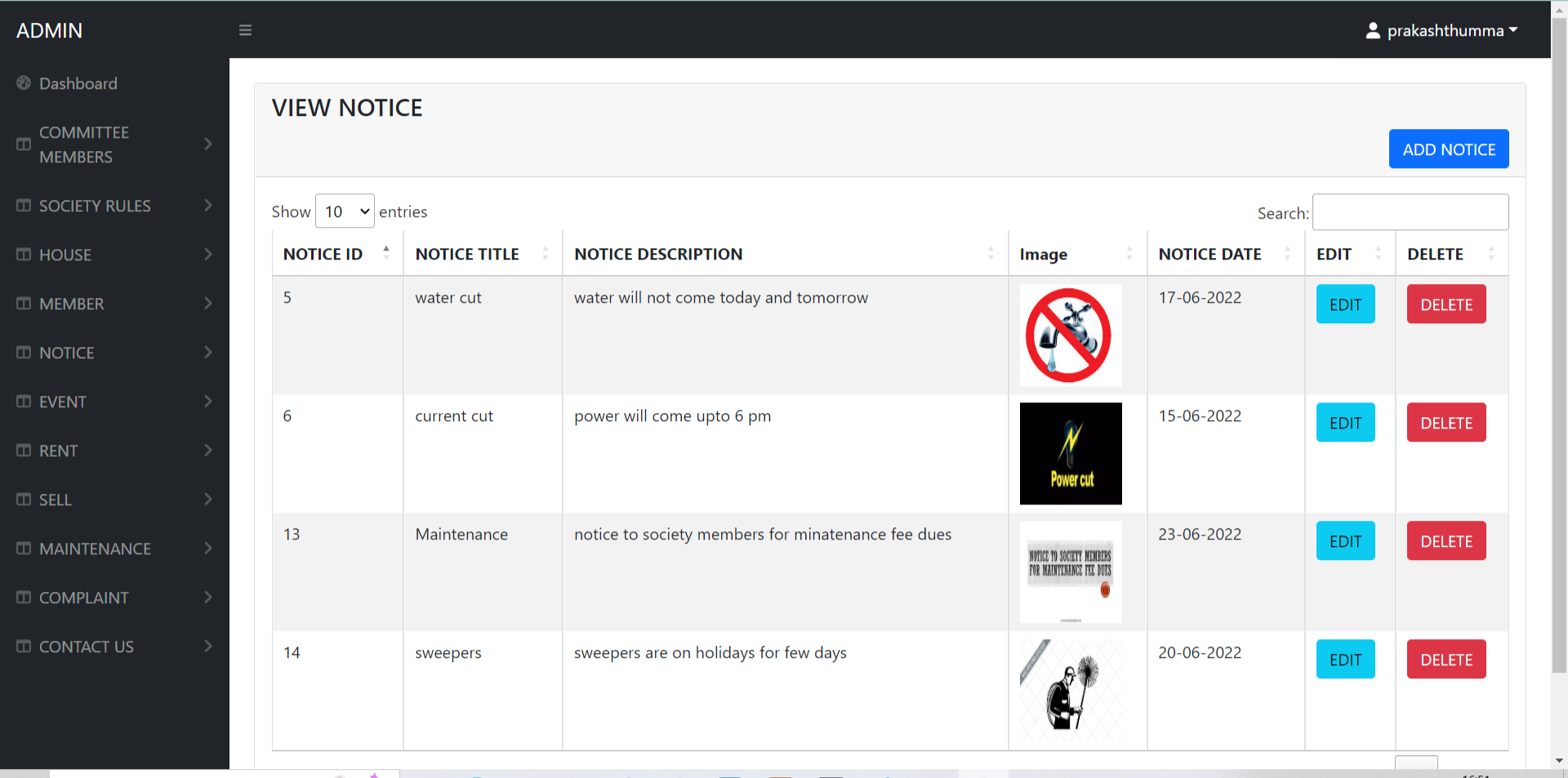


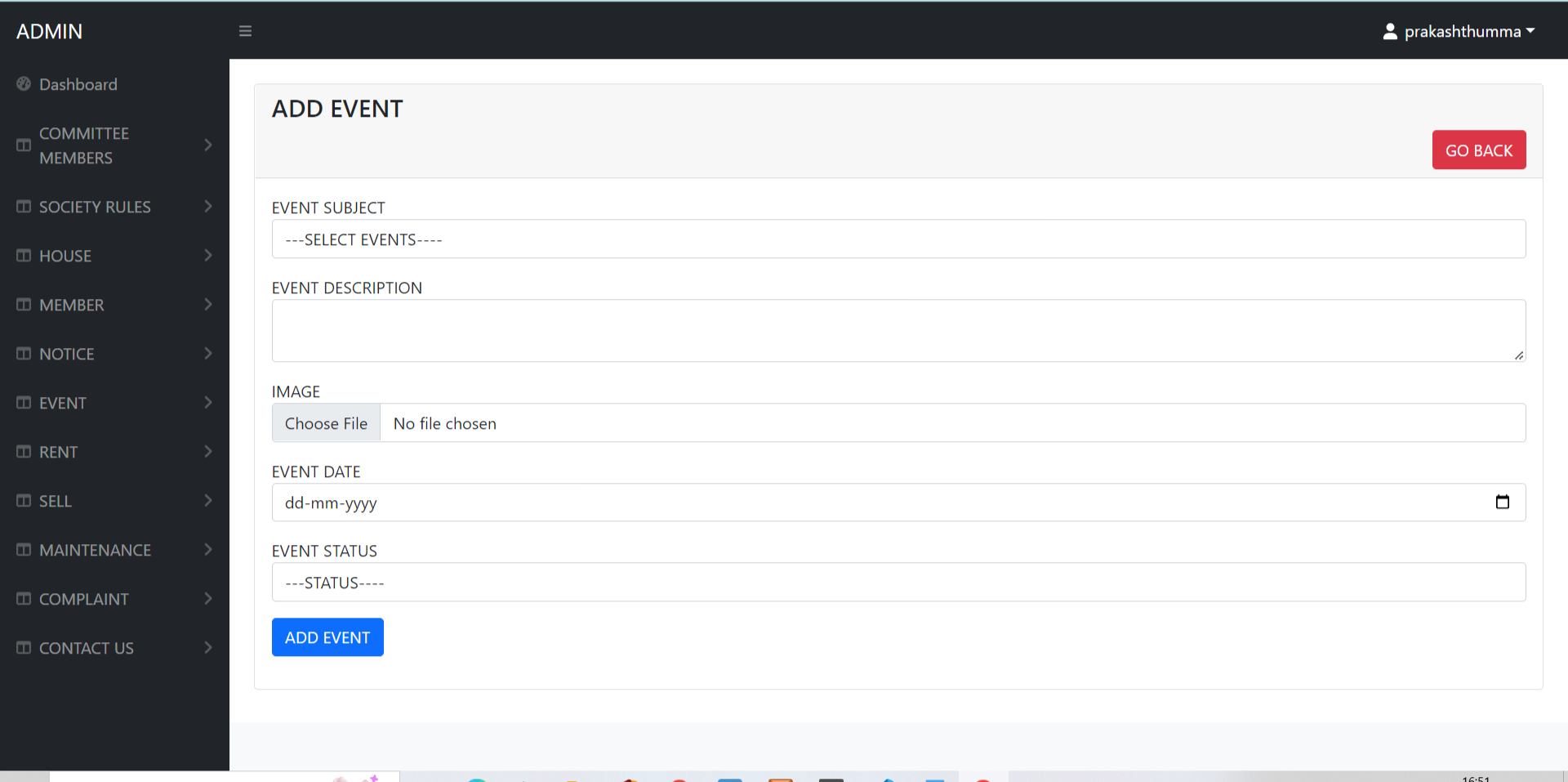


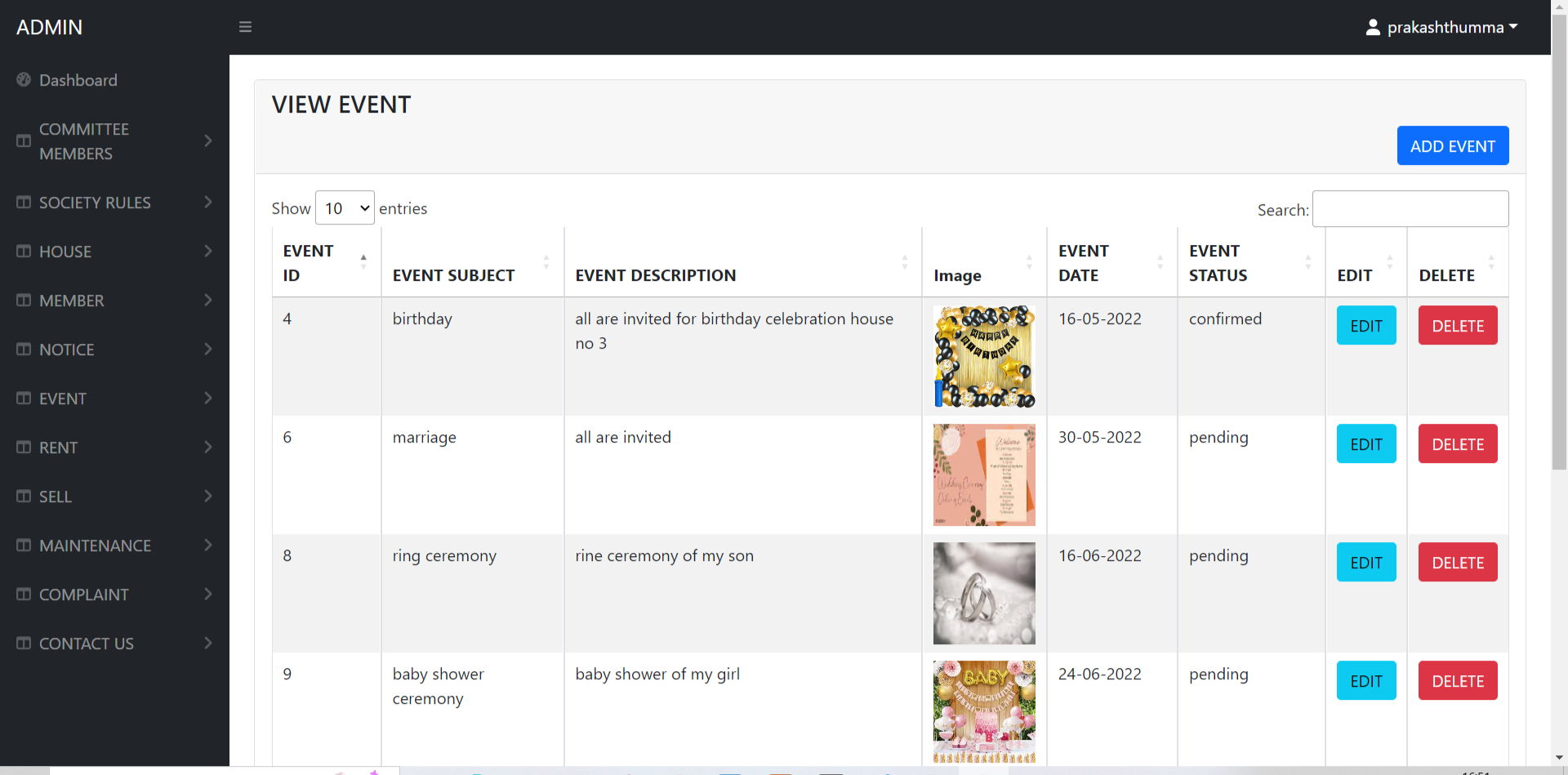


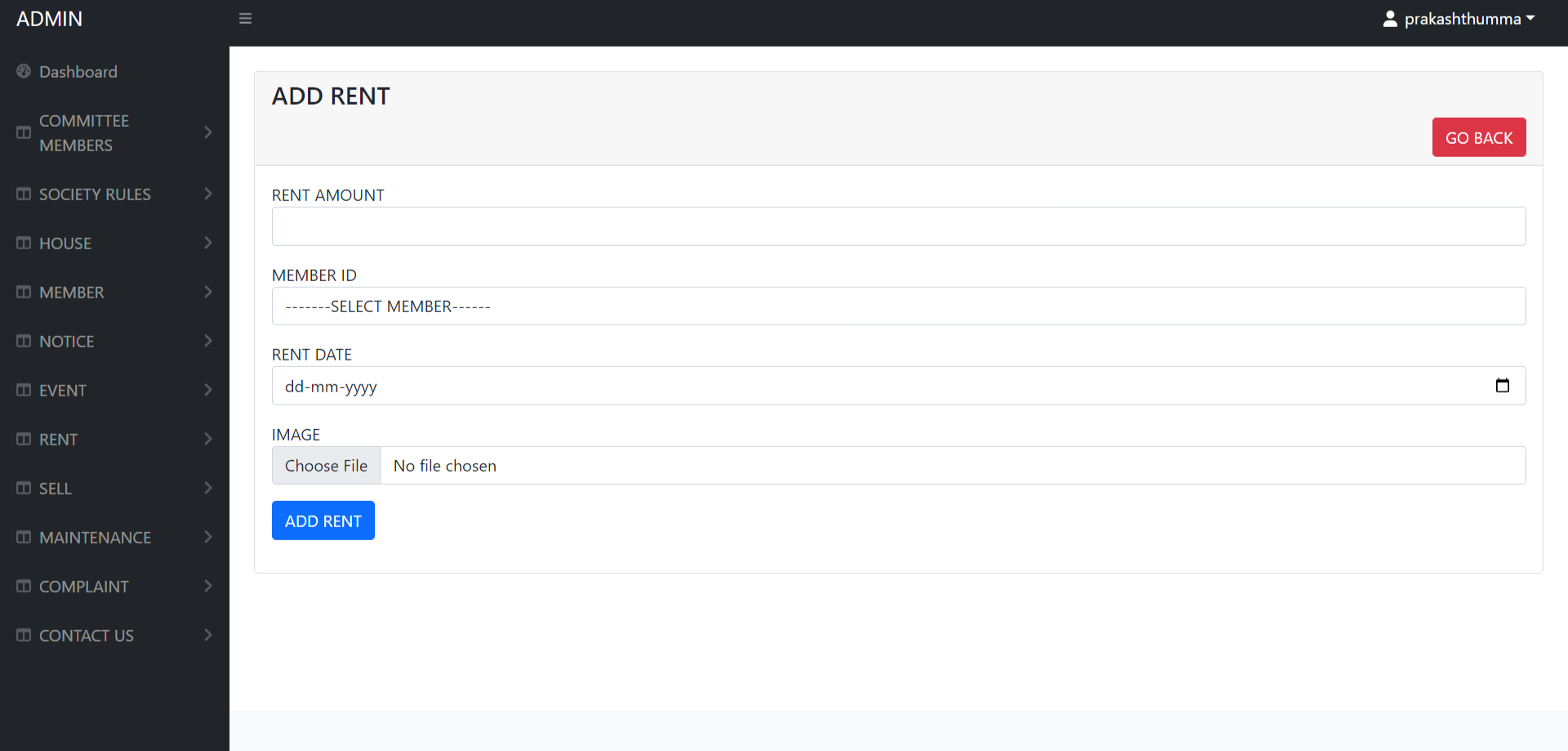


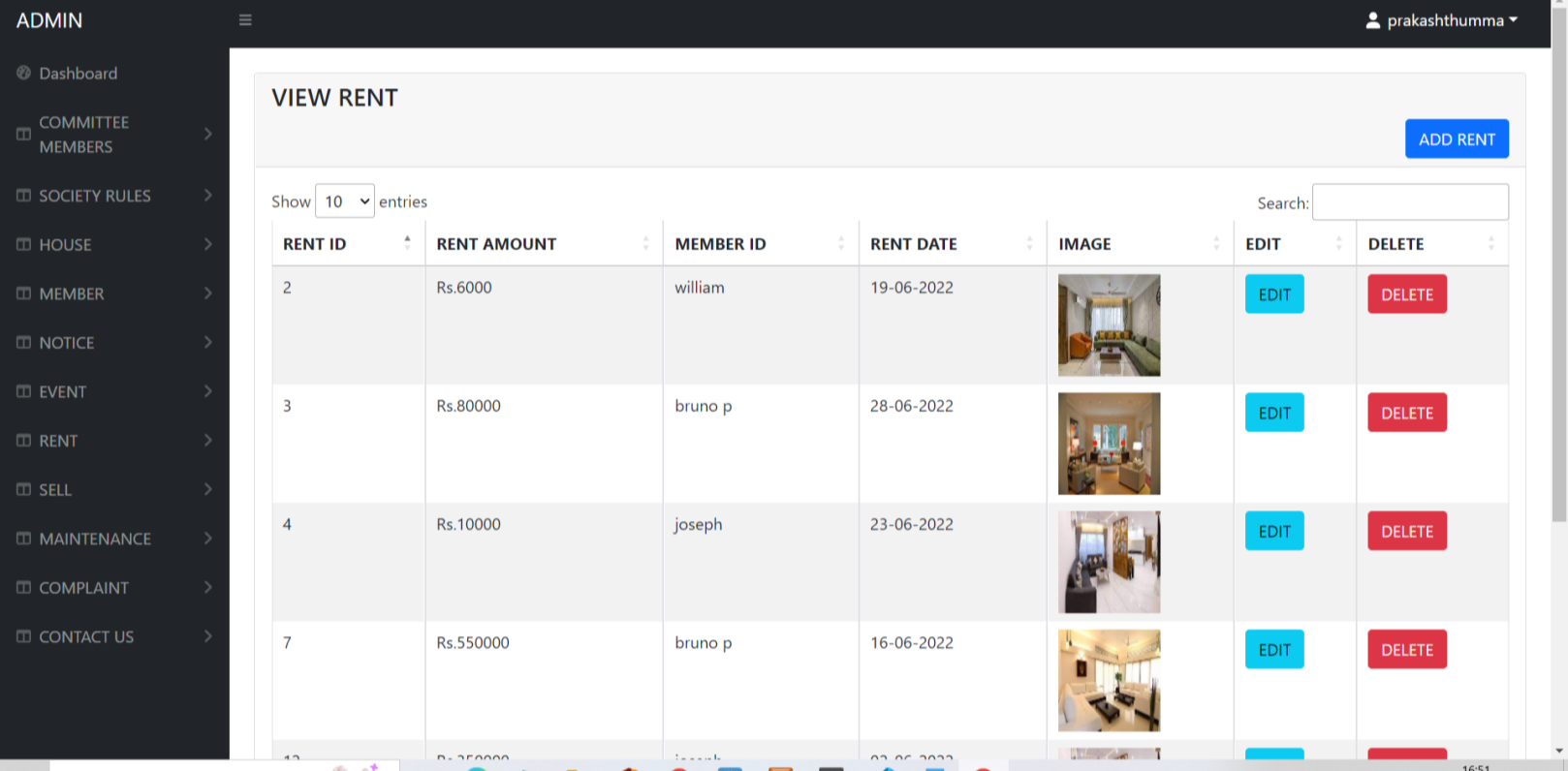


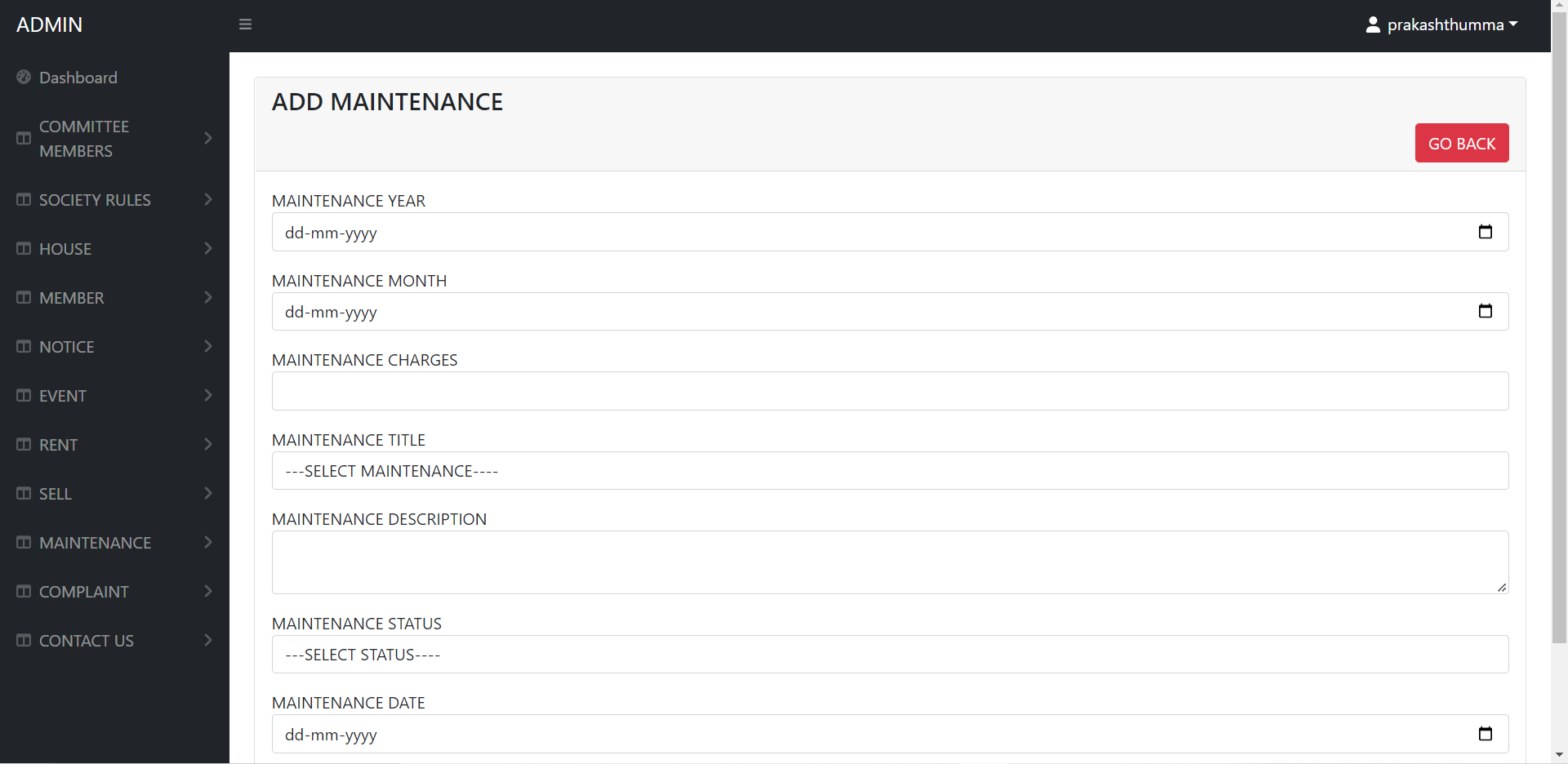


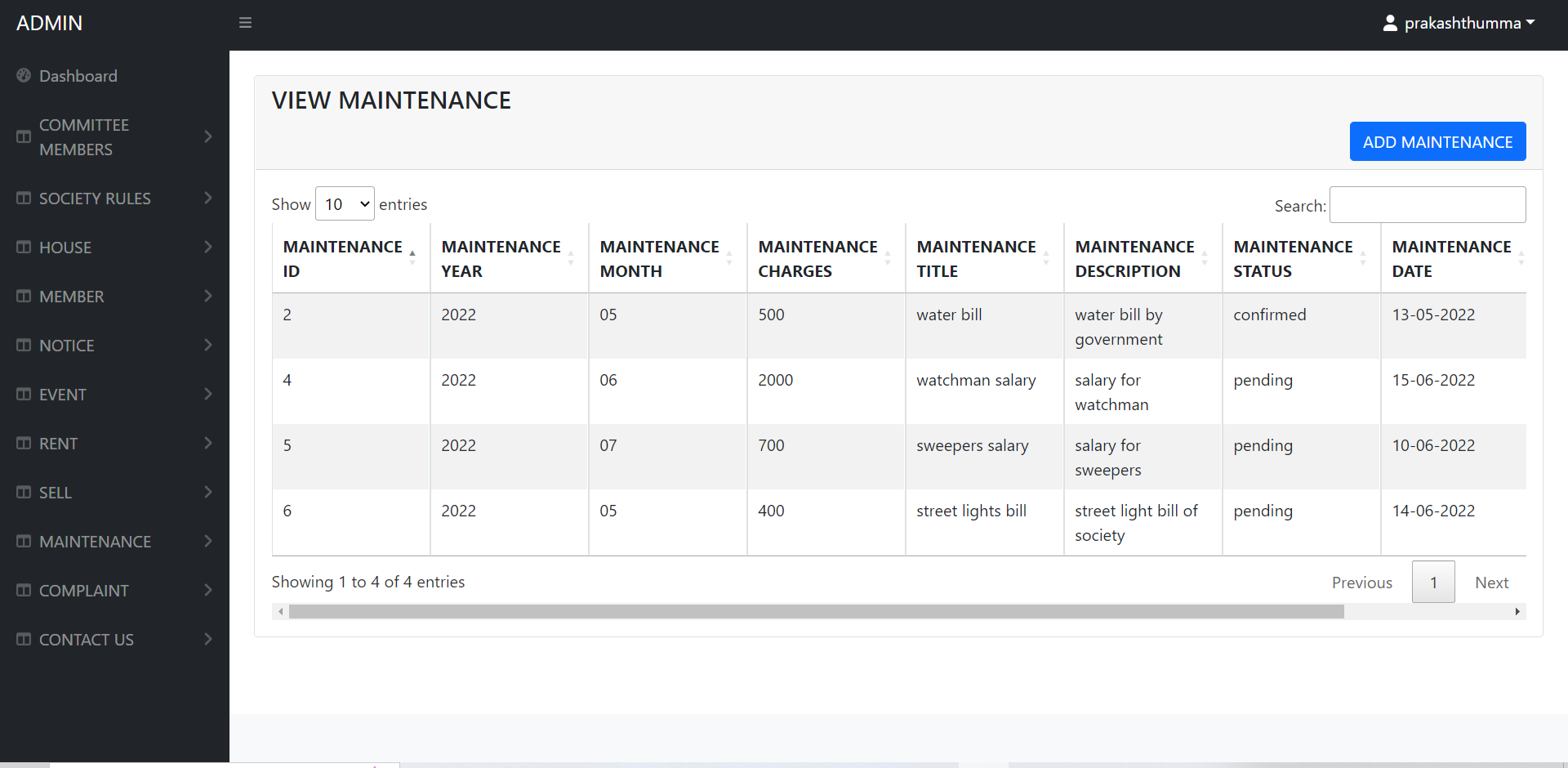


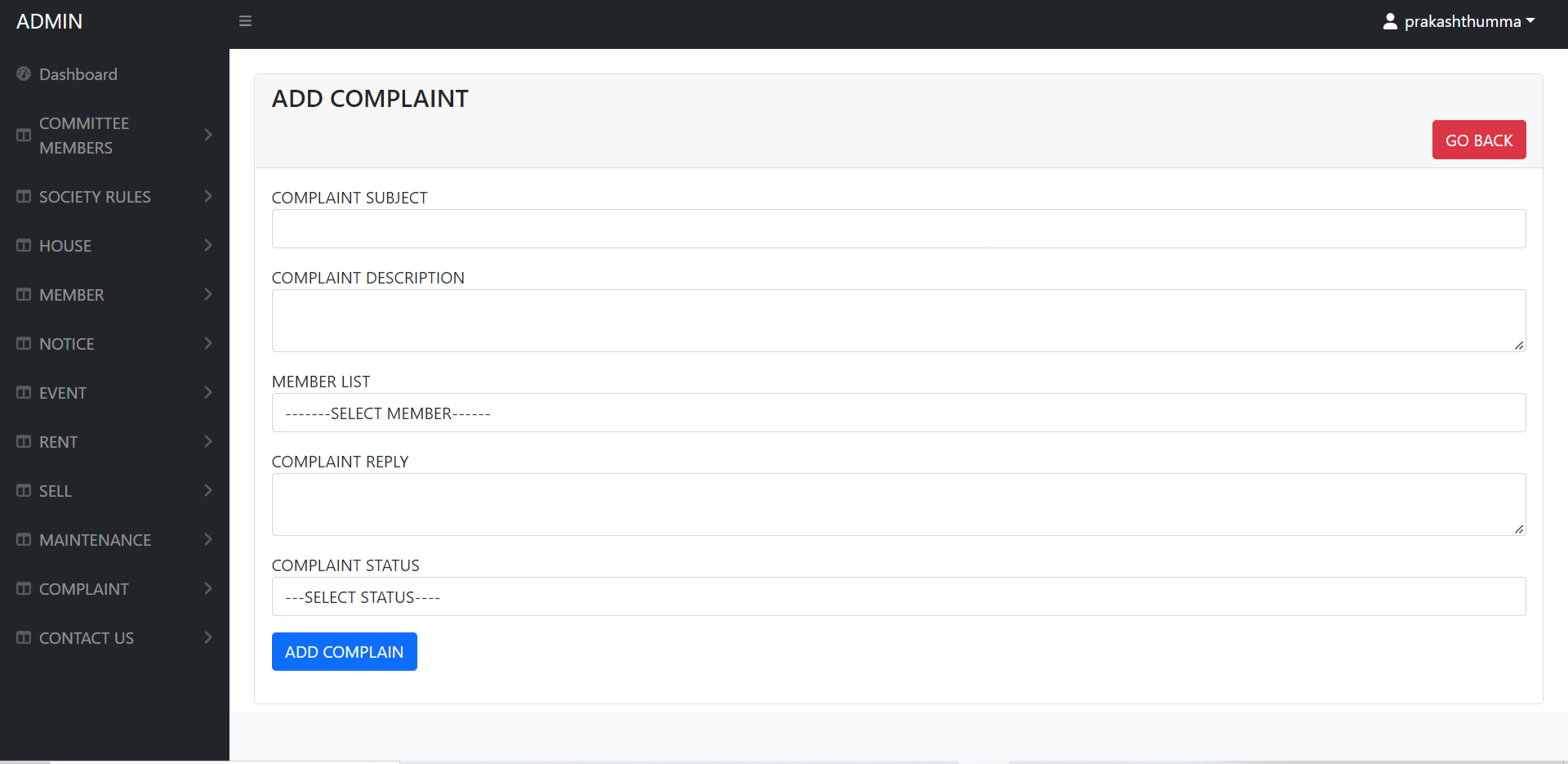


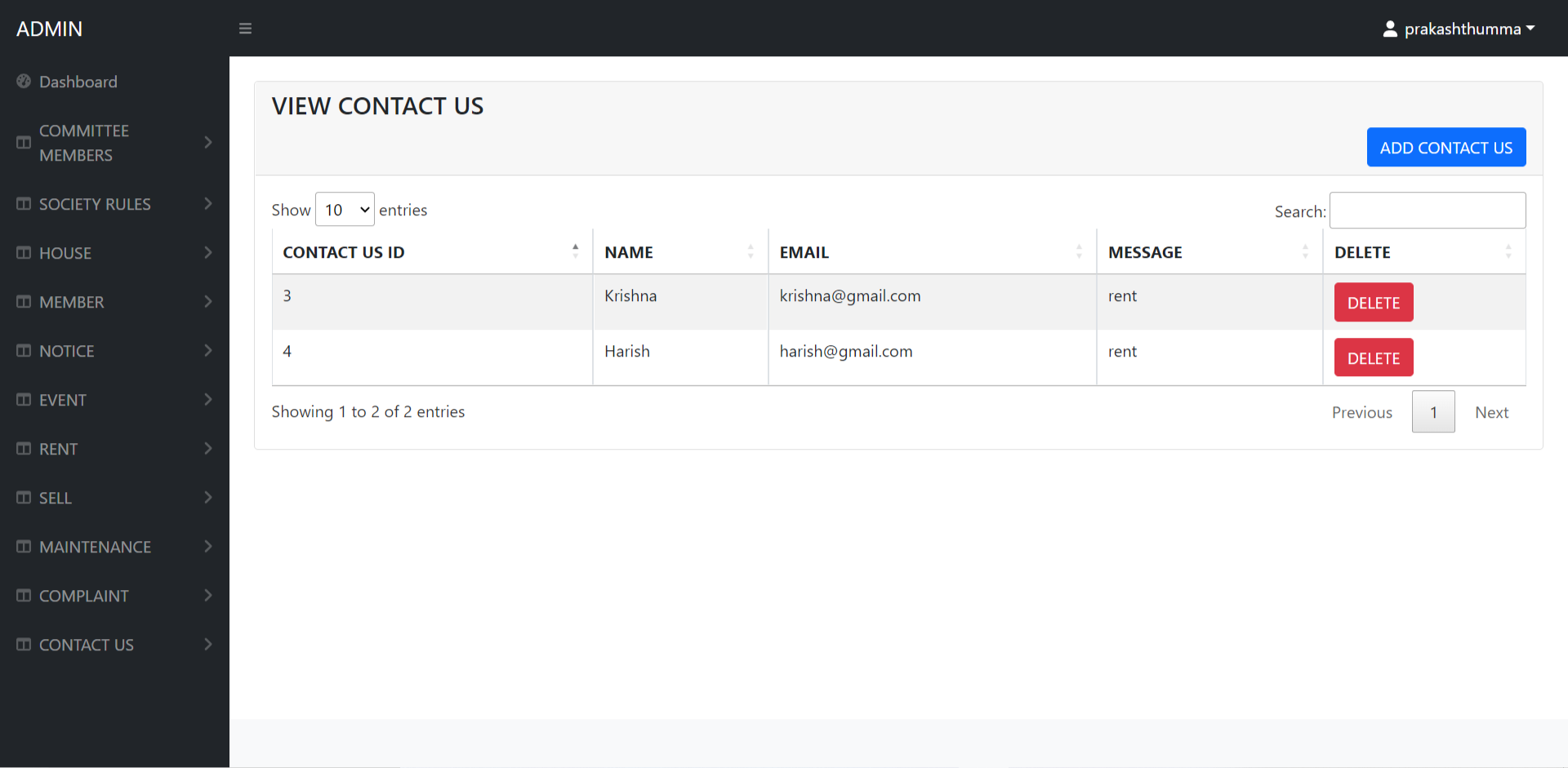
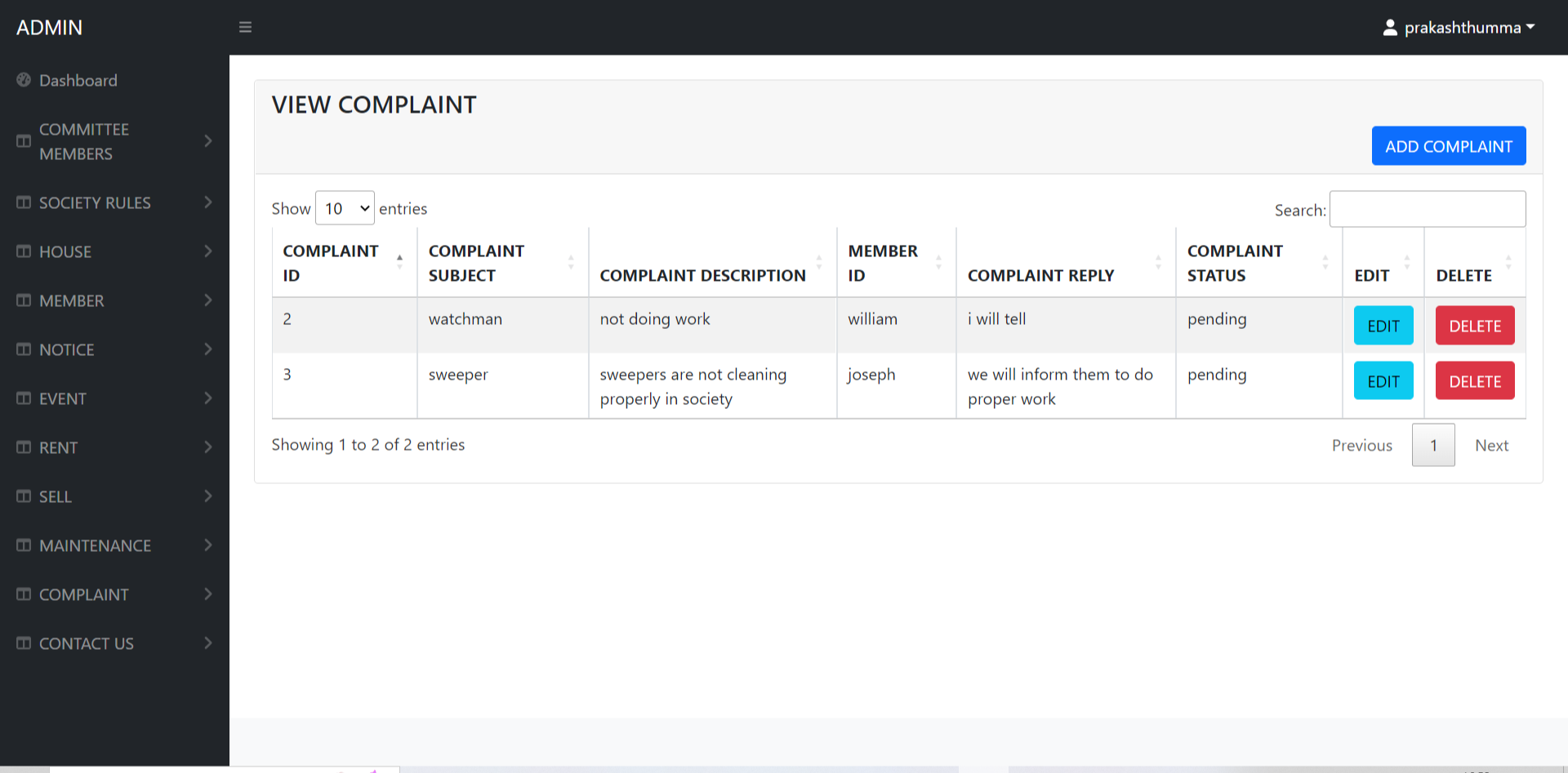






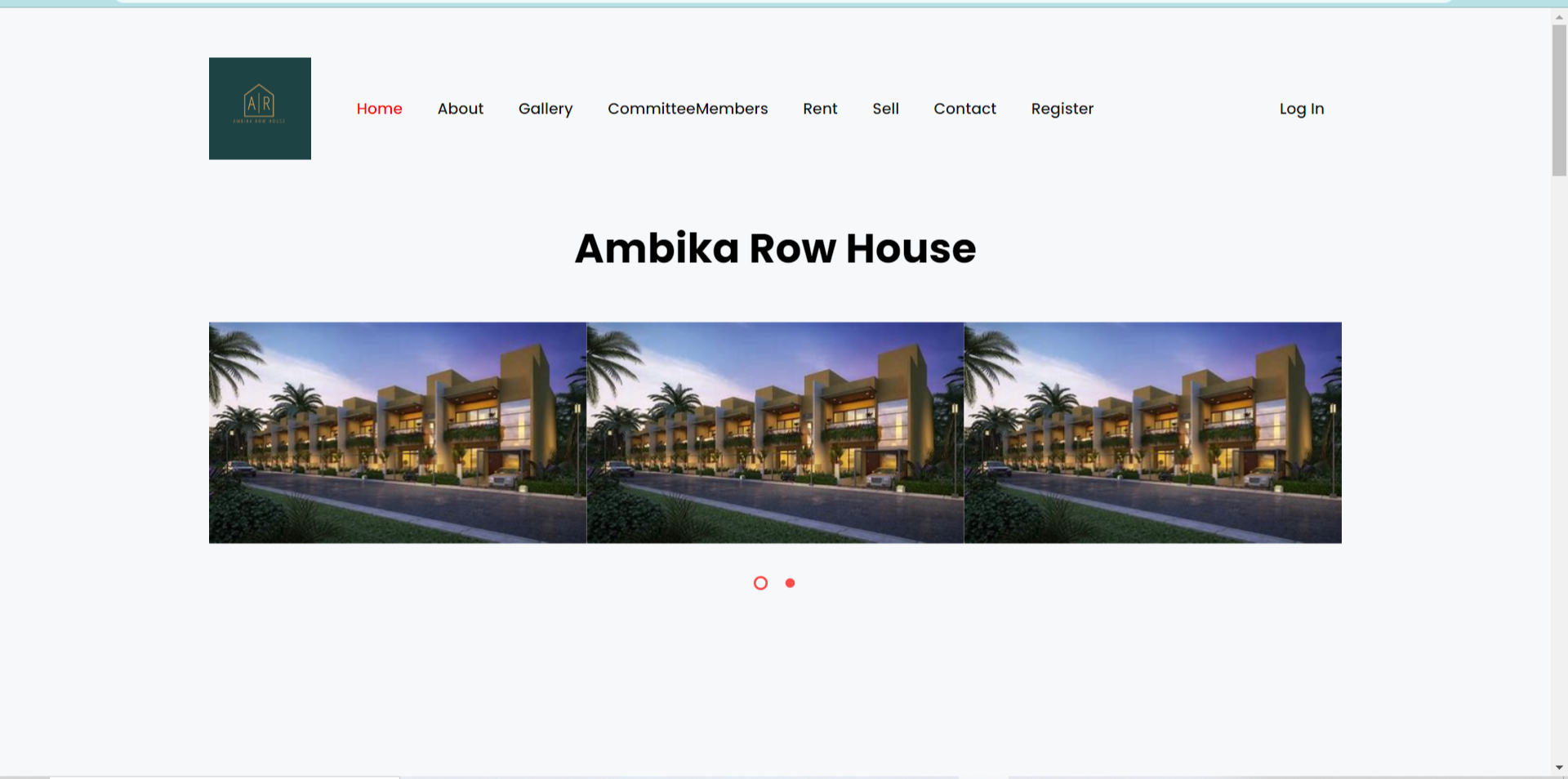


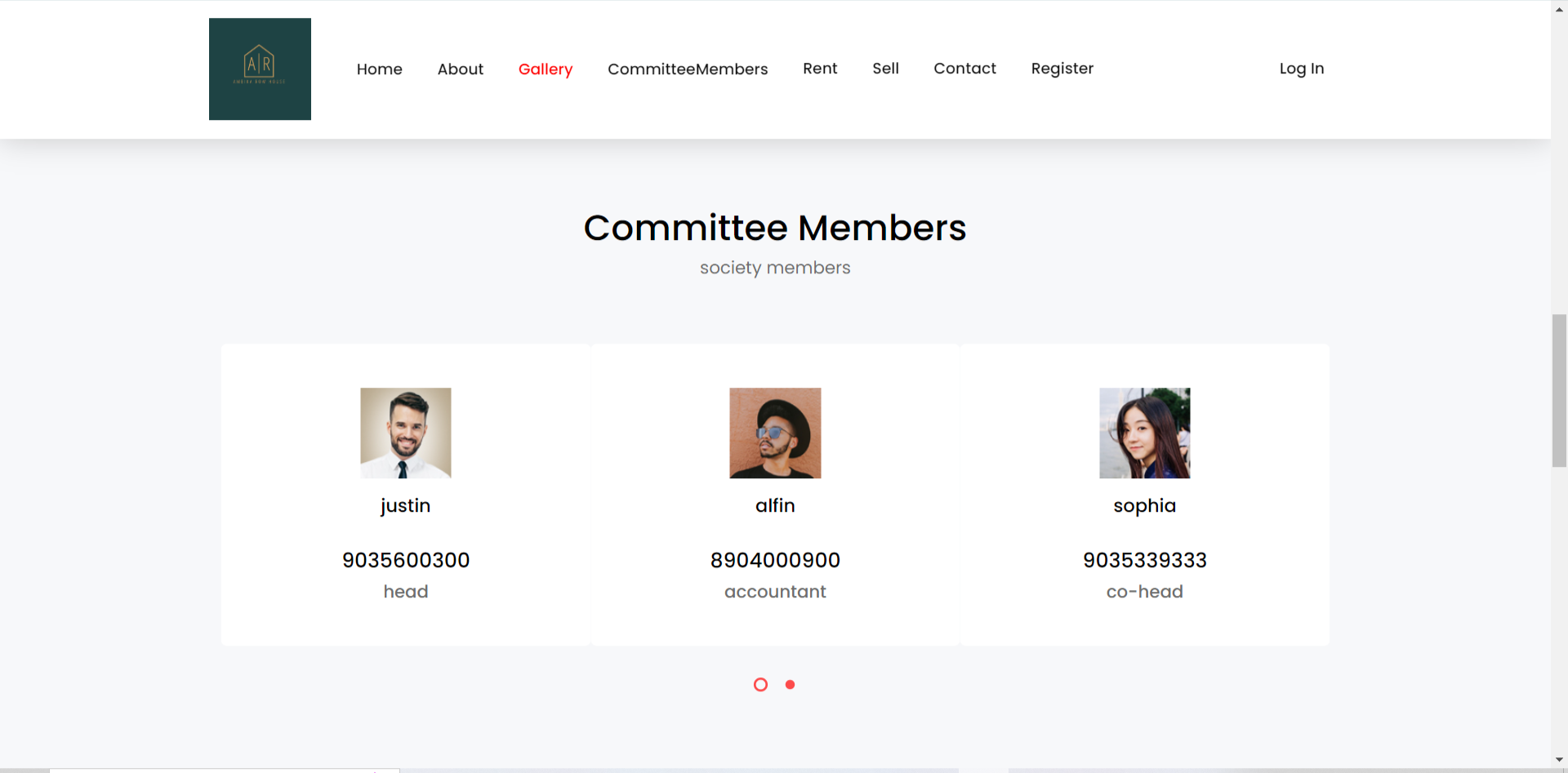
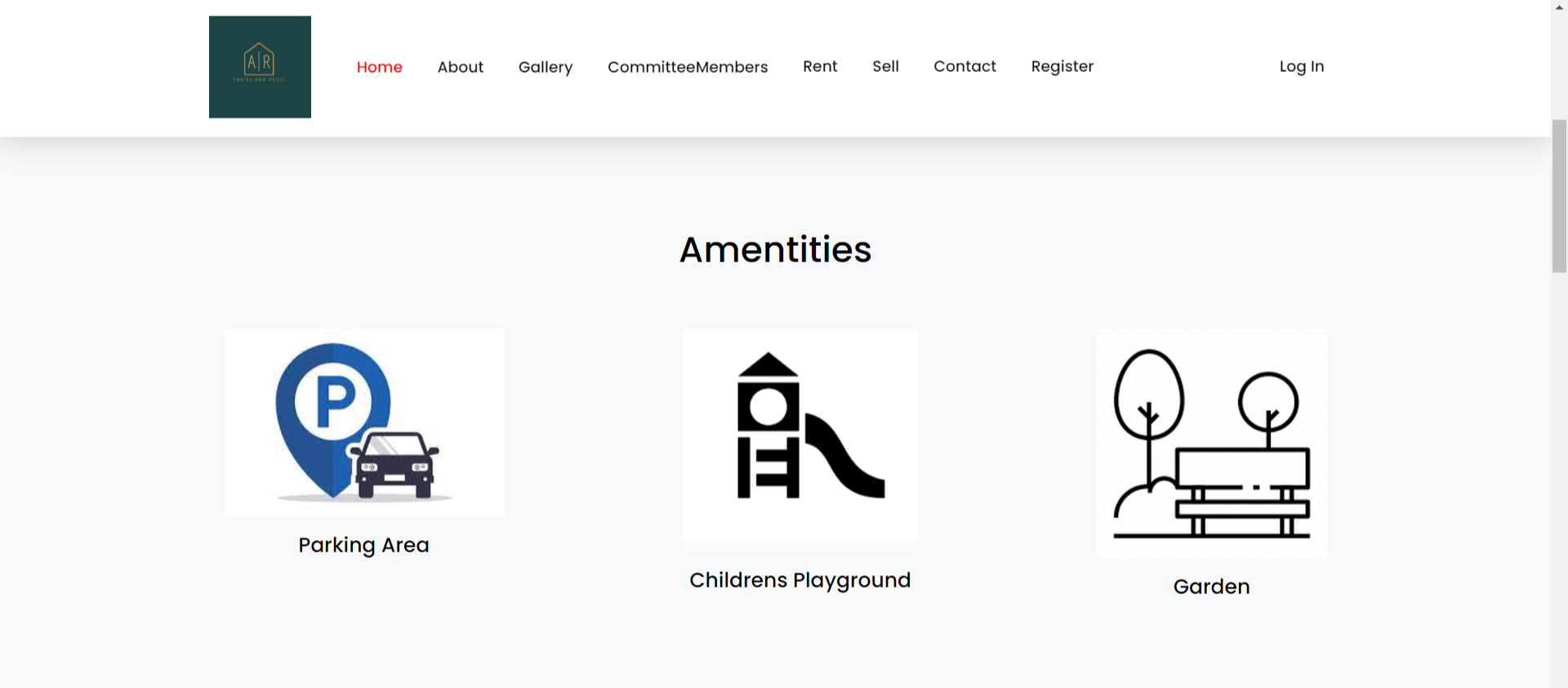


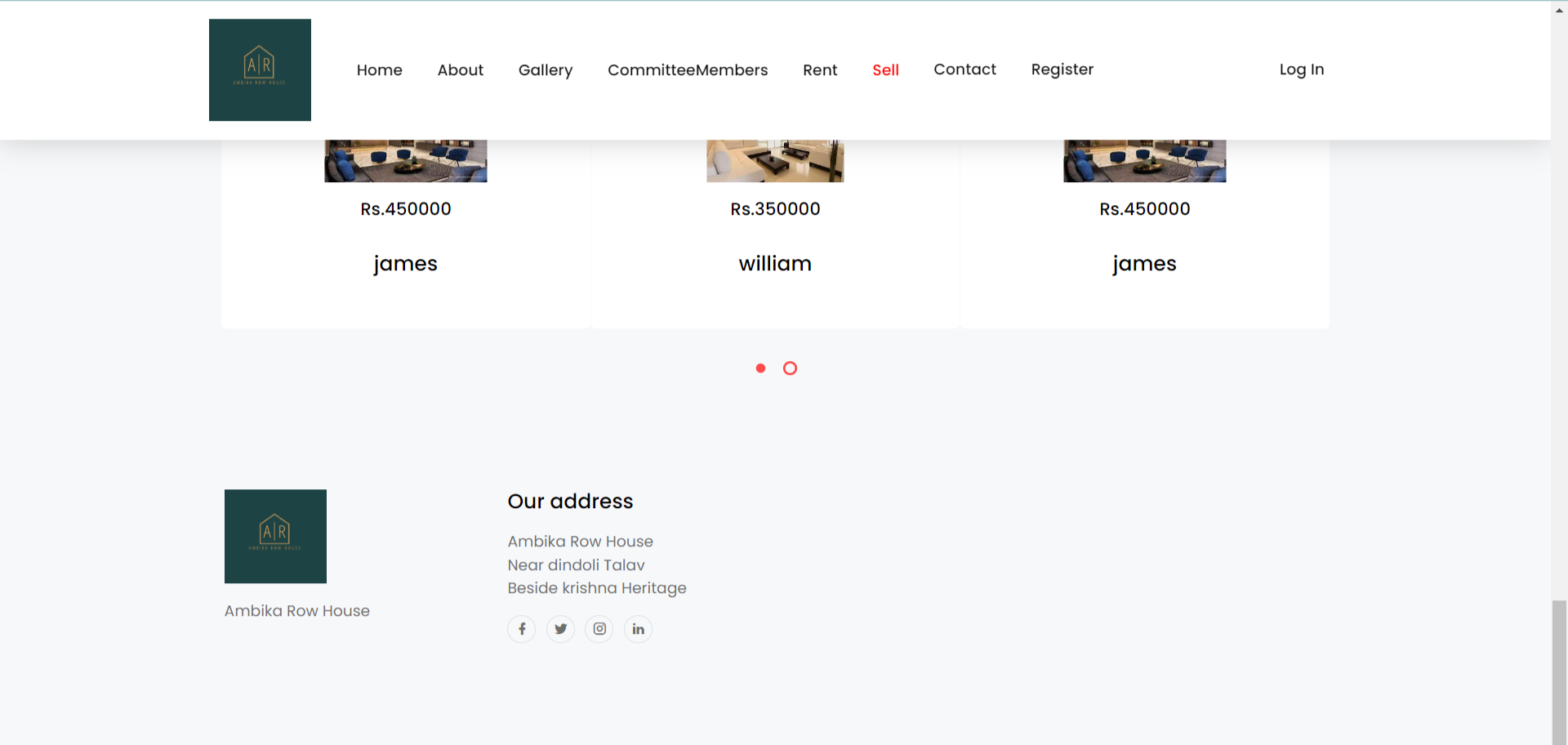
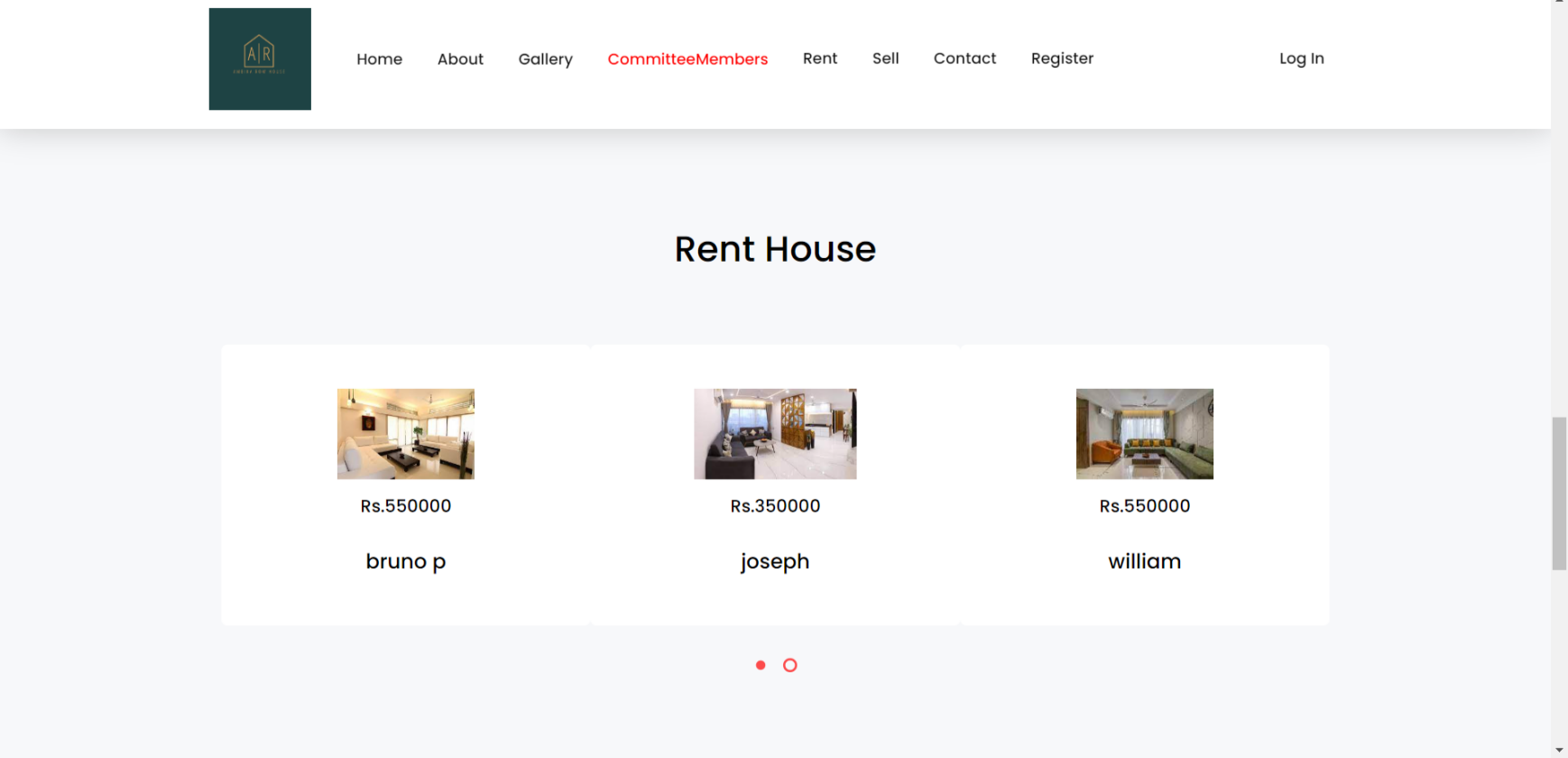


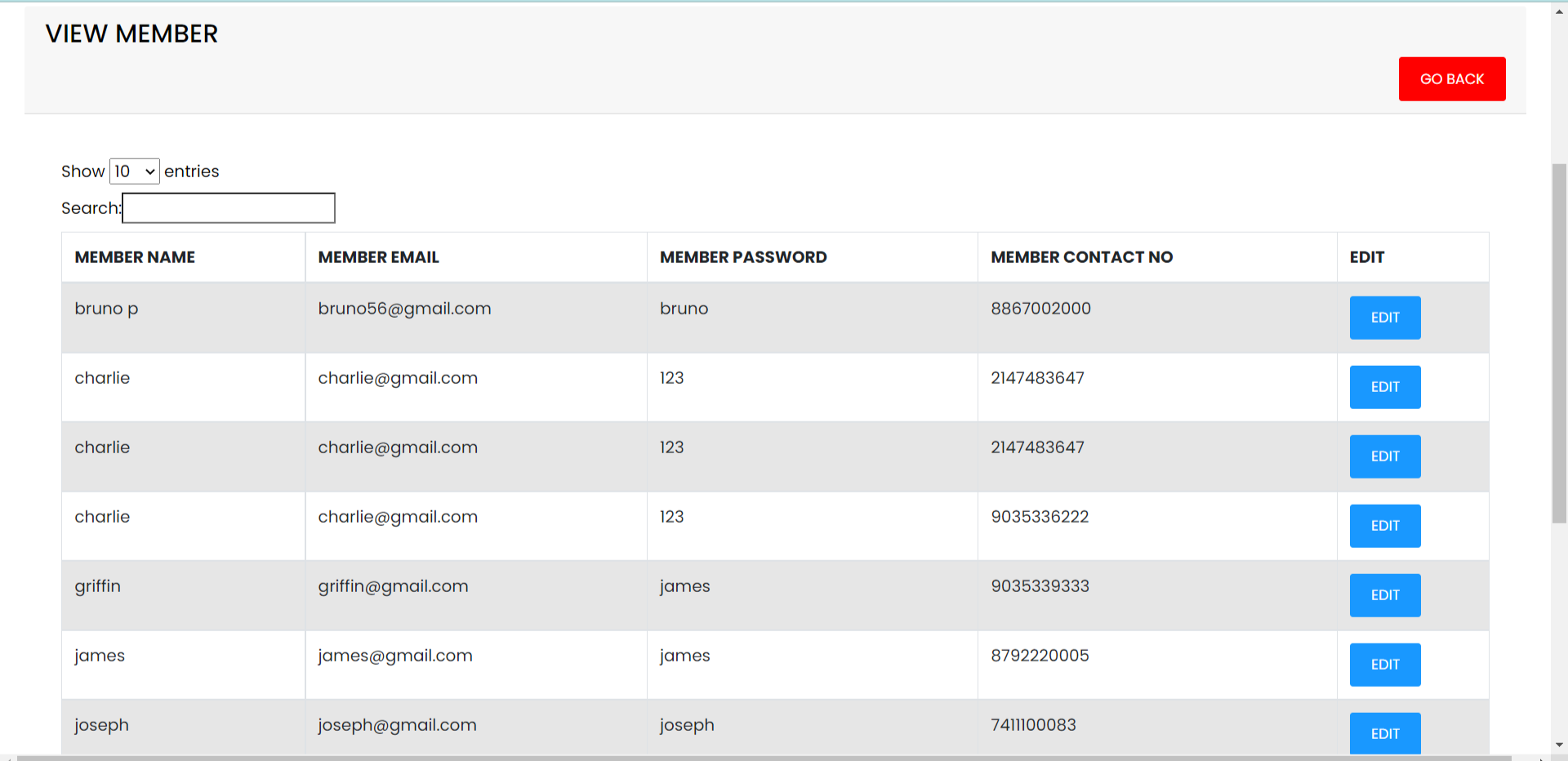
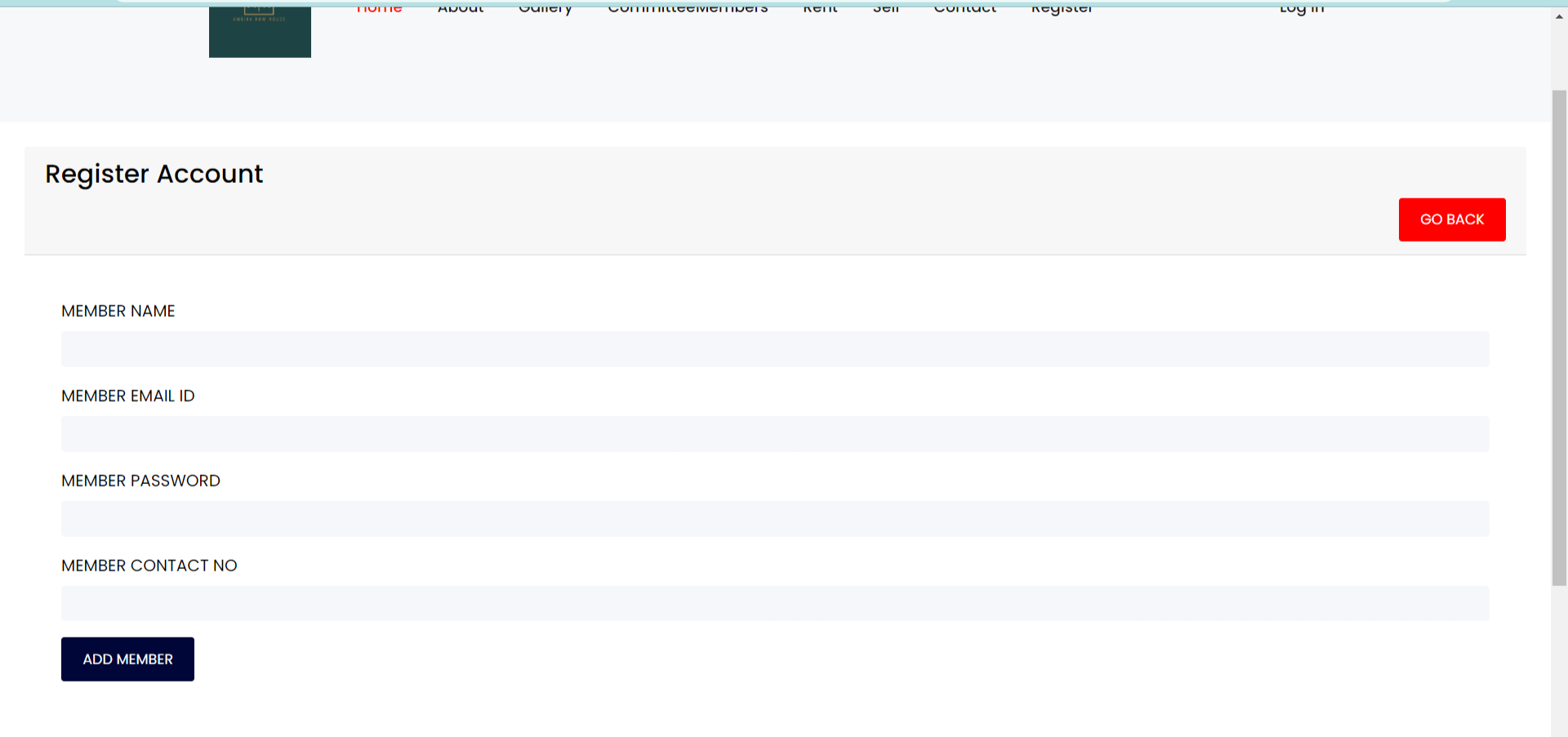
User panel

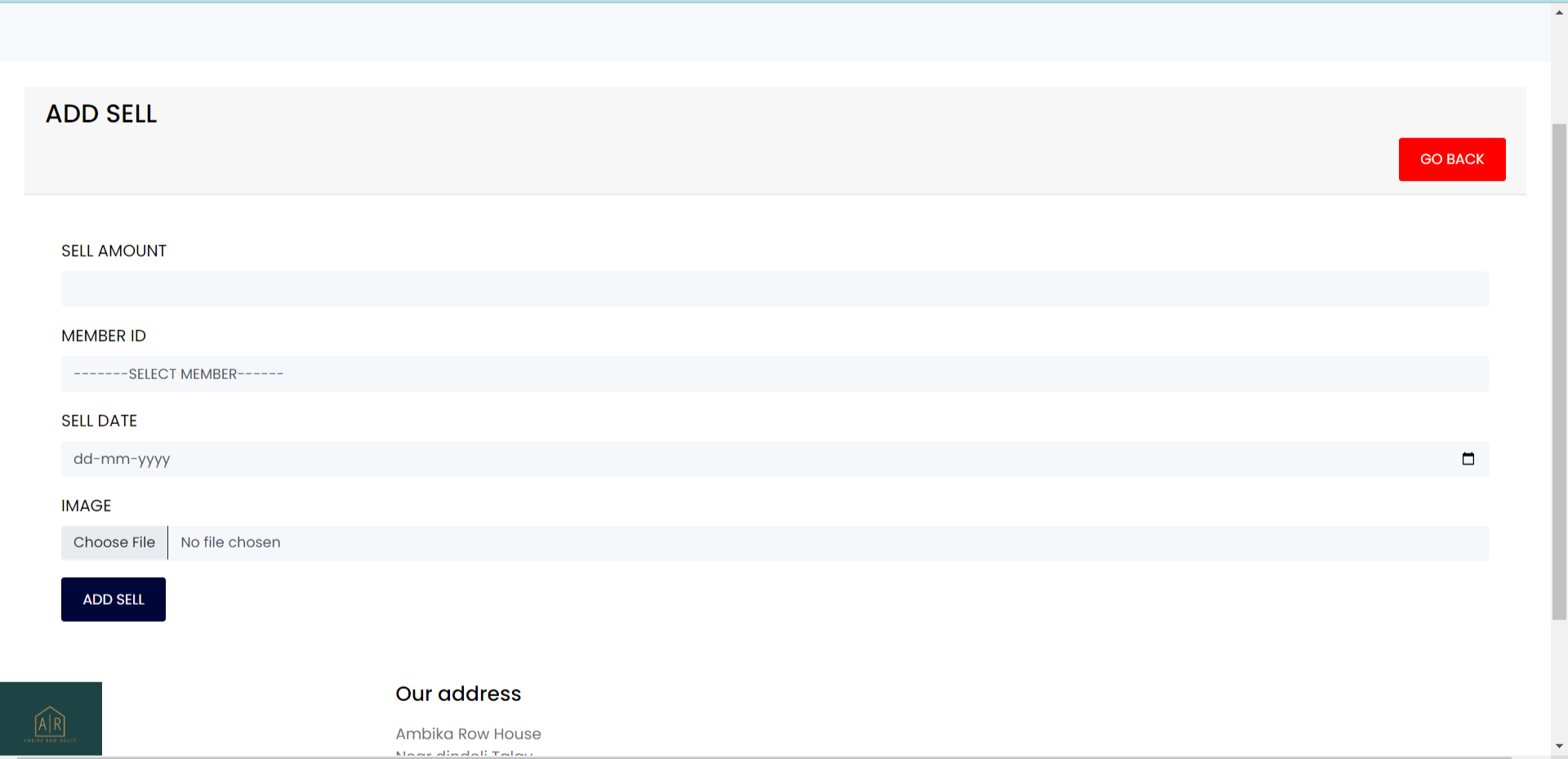
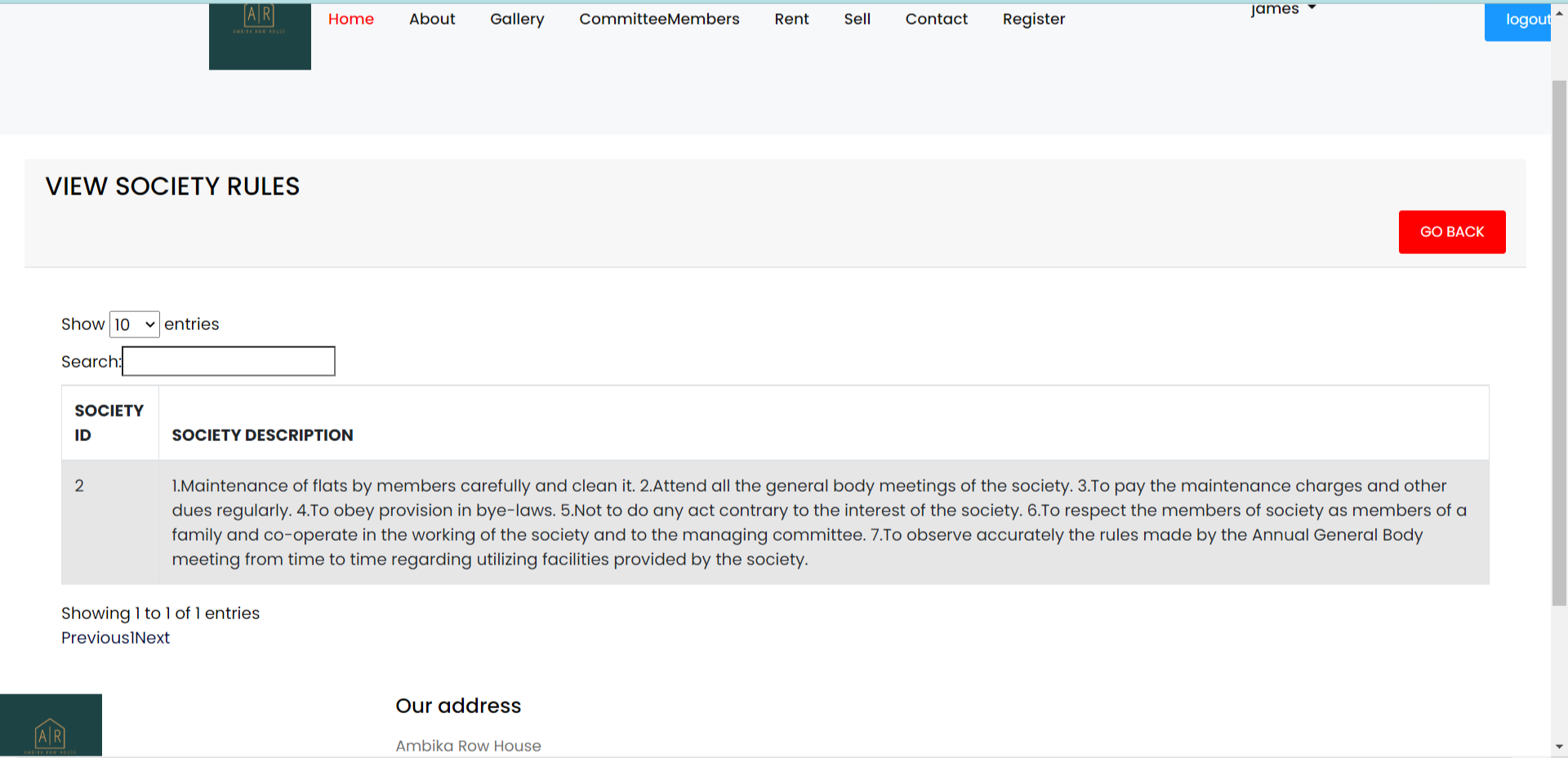
User Dashboard

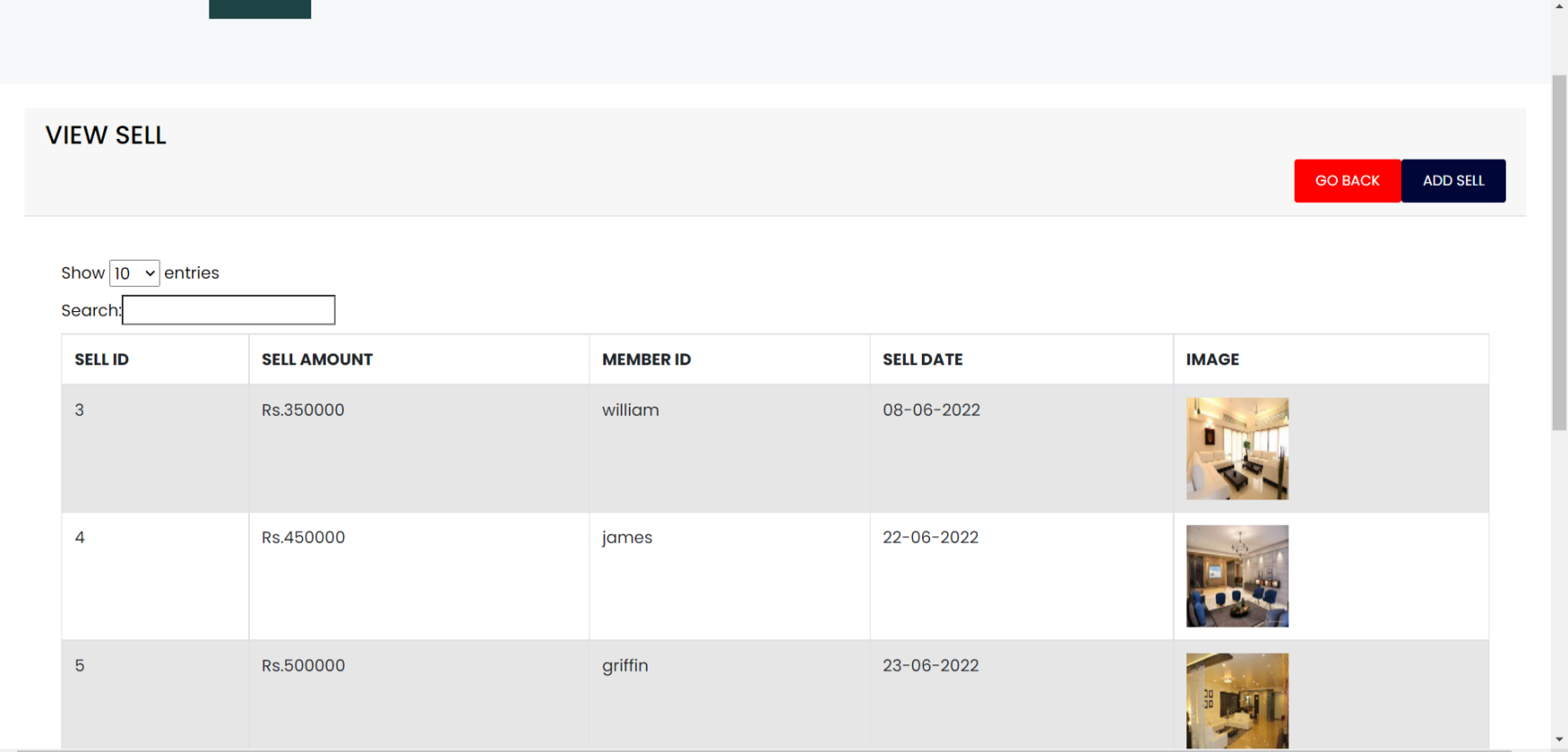












1. Testing
   1. Unit Testing
      * Login Module

Member unable to view Member details when member doesn't register or login to the application.

Member unable to add properties into application when member doesn't register or login to the application.

# Future Enhancement

* + Advance search will be implemented Properly.
  + Member can add property video.
  + member’s property can display based on admin’s approval.
  + Payment process will implement later on(rent payment).
  + Member will be notify by mail (notification) based on payment of customer's choice ex. Payment pending or Payment Successful
  + Package for seller will we include later into system.
  + New Properties and Popular Properties filter will be added for future.

**9.Reference**

 [www.google.com](http://www.google.com/)

 [www.youtube.com](http://www.youtube.com/)

 [www. getbootstrap.com](http://www.google.com/)