

**Project Topic: To-Let Management System**

**Documentation on SRS**

Course title: System Analysis and Design Project

Course code: SE-231

Prepared By: Submitted To:

Name: Md. Sohanur Rahman Ovi  Khalid Been Md. Badruzzaman Biplob

ID: 171-35-1912  Lecturer

Name: Jarin Tabassum

ID: 181-35-2363

**Department of Software Engineering**

**Daffodil International University**

**ABSTRACT**

We are stuck with technology when what we really want is just stuff that works. With the current paradigm shift in technological field, there is an urgent need to embrace and appreciate the power of technology. Housing sector remains vigilant to face the challenges of change by employing a new strategy that facilitates easy management of rental houses. Hence, there is need to develop a to-let management system that can simplify work for the rental managers so that all their work can be efficient and effective. The To-Let Management System is

Searching in Based on the Apartment Paying Guest, Office, House in metropolitan cities. The To-Let Management System is Based on the Owners and the Customers. The Owner is updated on the Apartment, Office details, House, Paying Guest details. The Customer is details about the Room space, Room rent and the Address Details also. The To-Let Management System is best Suitable the owners because time save and the only contact and the eligible person and there is no need to explain the room details on the speak. The To-Let Management System is best application in the city place The customer contact and the easily search and the suitable place of Apartment, Office, PG, House and based the Money, Limit Person is based on the suitable house. The To-Let Management System is save the time also. The To-Let Management System is used to easily identify the suitable place in Save time, cost also. The To-Let Management System is best way to search the house, Apartment office, Paying Guest. Hence, this system is best applicable for the above reasons making House rental an easy process through an online system.

**TABLE OF COTENTS**

|  |  |  |
| --- | --- | --- |
| SE NO | PROJECT TOPICS | PAGES |
| 1. | Project Proposal | 04-06 |
| 2. | SRS  **1.**   **Introduction**               1.1 Purpose               1.2 Project Scope               1.3 Glossary               1.4 References               1.5 Overview  **2.**   **User Classes and Characteristics**  **3.**   **Requirement Specification**              3.1     Functional Requirements              3.2     Performance Requirements              3.3     Dependability Requirements              3.4     Maintainability and Supportability              3.5     Security Requirements  3.6 Look and Feel Requirements  3.7 Non-Functional Requirements | 07-18 |
| 3. | Use-Case Diagram | 19 |
| 4. | Use-Case Description | 20-31 |
| 5. | Activity Diagram | 32-38 |
| 6. | Sequence Diagram | 39 |
| 7. | Entity Relationship Diagram | 40 |

**1.Project proposal:**

**Background Study:**

This is such a system which can be provide home rent facility to the students, bachelors and other peoples. The “To-Let Management System” is such a system, which can provide the facilities from any places with very low cost. This system is very much efficient and effective because it is fully automated.

Our aim to build a complete “To-Let Management System”. This system will ensure that people can get their house without going outside.

**Objectives:**

To-let Management System will be a web based online system which will provide home rent facilities to people, specially to the students and bachelors. As the number of web surfers are increasing day by day, a web based solution will be more helpful than physically searching the house for rent. The system will have different level of user. The one who is not registered as a user will only be able to view the basic home page.

The users of the System will be-

1. Admin
2. House owner
3. Tenant/ Renter

**Problem Statement:**

There are lots of problem in manual house rent system. Such as:

* People have to search their house by going outside and looking their required area.
* Physically going outside for house rent is a waste of time and waste of money.
* In this modern digitalized world we want to do everything easily and without time waste.

People cannot easily find rental information such as a house, apartment, sublet, bachelor house

**2.Software Requirements Specification:**

**1. Introduction:**

The introduction of the Software Requirements Specification (SRS) provides an overview of the entire SRS with purpose, scope, definitions, acronyms, abbreviations, references and overview of the SRS. The aim of this document is to gather and analyze and give an in-depth insight of the complete To-let Management System by defining the problem statement in detail. Nevertheless, it also concentrates on the capabilities required by stakeholders and their needs while defining high-level product features. The detailed requirements of the To-let Management System are provided in this document.

**1.1 Purpose:**

The purpose of the document is to collect and analyze all assorted ideas that have come up to define the system, its requirements with renter and house owner. Also, we shall predict and sort out how we hope this system will be used in order to gain a better understanding of the project, outline concepts that may be developed later,

and document ideas that are being considered, but may be discarded as the system develops. In short, the purpose of this SRS document is to provide a detailed

overview of our software product, its parameters and goals. This document describes the system and its associate’s members and its user interface, hardware and software requirements. It defines how admin, owners' and renter see the system and its functionality. Nonetheless, it helps any designer and developer to assist in software delivery lifecycle (SDLC) processes

**1.2 Project** **Scope:**

Primarily, the scope pertains to the house rent features for making To-let Management System. It focuses on the system, the stakeholders , which allow renters and house owners to be connected in an online platform and keep relationship between them.

This SRS is also aimed at specifying requirements of software to be developed but it can also be applied to assist in the selection relation between the owner and renters. The standard can be used to create software requirements specifications directly or can be used as a model for defining the system requirements.

**1.3 Glossary**

This subsection contains definitions of all the terms, acronyms, and abbreviations used in the document. Terms and concepts from the application domain are defined.

* SRS – System Requirement Specification
* SDLC – Software Development Life Cycle
* UI – User Interface

**1.4 References**

IEEE. IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.

**1.5 Overview**

This system is designed to be a facility for maintaining relationship between the house owner and renter such as search house, view house, send request is available to any renter as well as command and control systems. The rest of the SRS examines the specifications of the To-let Management System in detail.

**2. User Classes and Characteristics**

There are three types of users in this system. They are The Admin, Owner and Renter.

**Admin:** Admin of this system can login, Manage features and users, create rent posts, see renters requests and also can delete any post.  The admin is also able to initially setup the system, add new users, and set their authorization level.

**Owner:** The house owner can sign up, login, create new post, see requests, review, confirm any renters booking ad also delete any post they want.

**Renter:** Renter can login, search house, view house, send request and confirm any house for booking .

**3. Requirement Specification**

The complete requirement specification based on the elicitation process is described in this section.

**3.1 Functional Requirements**

The Functional Requirements Specification is designed to be read by a general audience. Readers should understand the system, but no particular technical knowledge should be required to understand the document.

|  |  |
| --- | --- |
| **FR-01** | **Sign in** |
| **Description** | This module helps Owner and Admin to register. Admin is able to maintain all the information of renter or owner |
| **Stakeholders** | Owner, Admin |

|  |  |
| --- | --- |
| **FR-02** | **Login** |
| **Description** | Using this module admin, renter and house can login to the interface |
| **Stakeholders** | Admin, Renter, Owner |

|  |  |
| --- | --- |
| **FR-03** | **Manage Feature** |
| **Description** | This module helps Admin to manage all the features of the system |
| **Stakeholders** | Admin |

|  |  |
| --- | --- |
| **FR-04** | **Manage User** |
| **Description** | This module helps  Admin to manage documents in the system |
| **Stakeholders** | Admin |

|  |  |
| --- | --- |
| **FR-05** | **Create Post** |
| **Description** | Owner and admin can create new rent post using this module |
| **Stakeholders** | Owner, Admin |

|  |  |
| --- | --- |
| **FR-06** | **Search house** |
| **Description** | Renter can search new house for rent |
| **Stakeholders** | Renter |

|  |  |
| --- | --- |
| **FR-07** | **View house** |
| **Description** | Renter can view houses using this module |
| **Stakeholders** | Renter |

|  |  |
| --- | --- |
| **FR-08** | **Send request** |
| **Description** | Renter can send request for new house |
| **Stakeholders** | Renter |

|  |  |
| --- | --- |
| **FR-09** | **See request** |
| **Description** | Using this module Admin and owner can see the requests |
| **Stakeholders** | Admin, owner |

|  |  |
| --- | --- |
| **FR-10** | **Rivew** |
| **Description** | Using this module owner and renter can give review |
| **Stakeholders** | Owner, Renter |

|  |  |
| --- | --- |
| **FR-11** | **Booking** |
| **Description** | Owner and renter can confirm their booking by using this module |
| **Stakeholders** | Owner, Renter |

|  |  |
| --- | --- |
| **FR-12** | **Delete post** |
| **Description** | Using this module Admin and owner can delete any post from the system if they want |
| **Stakeholders** | Admin, Owner |

**3.2 Performance Requirements**

A requirement that specifies a performance characteristic that a system or system component must possess; for example, speed, accuracy, frequency.

**3.3 Dependability Requirements** ​

The flexibility of current frameworks encourage system architects to enable reconfiguration mechanisms that refocus the available, safe resources to support the most critical services rather than over-provisioning to build failure-proof system. Therefore, these requirements are essentials.

**3.4 Maintainability and Supportability**

Supportability is the degree to which system design characteristics and planned logistics resources meet system requirements. Supportability is the capability of a total system design to support operations and readiness needs throughout the life-cycle of a system at an affordable cost.

 ​

**3.5 Security Requirements**

There are no access requirements beside those that have been outlined in the below:

1. The software must validate all user input to ensure it does not exceed the size specified for that type of input
2. The server must authenticate every request accessing the restricted Web pages
3. After authenticating the browser, the server must determine whether that browser is authorized to access the requested restricted Web pages
4. The system must have security controls to protect against denial-of-service attacks
5. The system must encrypt sensitive data transmitted over the Internet between the server and the browser
6. To get access to this system or a specific module the system must provide a central authentication mechanism. In order to prevent anyone to exploit stolen all users password must be encrypted in hash process.

**3.6 Look and Feel Requirements**

The look and feel requirements describe the intended spirit, the mood, or the style of the product's appearance. These requirements specify the intention of the appearance, and are not a detailed design of an interface.

**3.7 Non-Functional Requirements**

* Performance: The proposed system that I going to be developed will be used as chief performance system for providing the whole information to the tenant and owner. System should give expected performance results and should be response within short time.
* Effectiveness: This system have to provide correct information.
* Security: This system must have the data security . There are three categories of users named as admin, renter and house-owner.
* Safety: The database may be crushed at any time due to OS failure or virus. So it is required to take the database backup.

**3.Use case Diagram:**

**4.Use Case Description**

**1.User Sign Up:**

|  |  |
| --- | --- |
| **Use case** | Sign Up |
| **Goal** | Register for access the system |
| **Preconditions** | None |
| **Success End Condition** | User registration successful |
| **Failed End Condition** | User registration unsuccessful |
| **Primary Actors**  **Secondary Actors** | Renter, House Owner  None |
| **Trigger** | New user successfully entire the system |
| **Description** | Sign up with this system for uses |
| **Alternative Flows** | Mandatory fields are missing, show  “Sign in failed” |
| **Quality Requirements** | 1. Enter first name 2. Enter last name 3. Enter email 4. Set Password 5. Click on sign up |

**2.Login**

|  |  |
| --- | --- |
| **Use case** | Login |
| **Goal** | Login for access the system |
| **Preconditions** | Sign up |
| **Success End Condition** | Login successful |
| **Failed End Condition** | Login unsuccessful |
| **Primary Actors**  **Secondary Actors** | Renter, Admin, House Owner  None |
| **Trigger** | User successfully entire the system |
| **Description** | Login with exact email and password for further access the system |
| **Alternative Flows** | Mandatory fields are missing, show  “Login failed” |
| **Quality Requirements** | 1. Enter email 2. Enter Password 3. Click on login |

**3. Manage Feature**

|  |  |
| --- | --- |
| **Use case** | Manage feature |
| **Goal** | Users easily explore the system |
| **Preconditions** | Need to login |
| **Success End Condition** | User required features are found |
| **Failed End Condition** | User required features are missing |
| **Primary Actors**  **Secondary Actors** | Admin  House Owner, Renter |
| **Trigger** | User can easily find his/her needed features |
| **Description** | Add more feature for users so that people can use this system very easily |
| **Alternative Flows** | Mandatory fields are missing, show  “Required” |
| **Quality Requirements** | 1. Add feature 2. Delete old feature 3. Update |

**4. Manage User**

|  |  |
| --- | --- |
| **Use case** | Manage user |
| **Goal** | Admin can easily manage the system users |
| **Preconditions** | Need to login |
| **Success End Condition** | Admin successfully manage user |
| **Failed End Condition** | Admin unsuccessful to manage user |
| **Primary Actors**  **Secondary Actors** | Admin  House Owner, Renter |
| **Trigger** | Admin can be easily monitoring the system users |
| **Description** | Allows the admin to: 1.Create  2.Remove  3.Edit  4.Reset any other account |
| **Alternative Flows** | Mandatory fields are missing, show  “Required” |
| **Quality Requirements** | Admin performs any action in the description. |

**5. Create Post**

|  |  |
| --- | --- |
| **Use case** | Create post |
| **Goal** | Admin/House owner can easily create post |
| **Preconditions** | Admin and House owner must login to the web system |
| **Success End Condition** | Successful to create post |
| **Failed End Condition** | Unsuccessful to create post |
| **Primary Actors**  **Secondary Actors** | Admin, House owner  Renter |
| **Trigger** | All the system user can view the post |
| **Description** | Adding a house information to be rented |
| **Alternative Flows** | Mandatory fields are missing, show  “Required” |
| **Quality Requirements** | Goes to the create post choice and adds the house information. |

**6. Search House**

|  |  |
| --- | --- |
| **Use case** | Search House |
| **Goal** | Renter can easily search for his required house |
| **Preconditions** | The user should open To-let management system website |
| **Success End Condition** | Successful to search house |
| **Failed End Condition** | Unsuccessful to search house |
| **Primary Actors**  **Secondary Actors** | Renter  Admin, House owner |
| **Trigger** | Renter will find his required search house |
| **Description** | Allows renter to search for house |
| **Alternative Flows** | Mandatory fields are missing, show  “Required” |
| **Quality Requirements** | Inputting in option of:  1.Price  2.Region  3.Area, ……….For searching |

**7. View House**

|  |  |
| --- | --- |
| **Use case** | View House |
| **Goal** | Renter can easily view houses |
| **Preconditions** | The user should open To-let management system website |
| **Success End Condition** | Successful to view house |
| **Failed End Condition** | Unsuccessful to view house |
| **Primary Actors**  **Secondary Actors** | Renter  Admin, House owner |
| **Trigger** | Renter will saw all the posted houses by the house owner |
| **Description** | Allows users to show details of house |
| **Alternative Flows** | None |
| **Quality Requirements** | Clicking on post users can see the house |

**8. Send Request**

|  |  |
| --- | --- |
| **Use case** | Send Request |
| **Goal** | Renter can easily send request |
| **Preconditions** | The user should open To-let management system website |
| **Success End Condition** | Successful to send request |
| **Failed End Condition** | Unsuccessful to send request |
| **Primary Actors**  **Secondary Actors** | Renter  House owner |
| **Trigger** | Renter will send a request for his required house, to the house owner |
| **Description** | Allows renter to apply for house for rent |
| **Alternative Flows** | Mandatory fields are missing, show  “Required” |
| **Quality Requirements** | Fill a form for the house rental |

**9. See Request**

|  |  |
| --- | --- |
| **Use case** | See Request |
| **Goal** | House owner can easily see request |
| **Preconditions** | The user should open To-let management system website |
| **Success End Condition** | Successful to see request |
| **Failed End Condition** | Unsuccessful to see request |
| **Primary Actors**  **Secondary Actors** | House owner  Renter |
| **Trigger** | House owner will see a request for his rental house, from the renter |
| **Description** | Receive a request form renter for rent house |
| **Alternative Flows** | None |
| **Quality Requirements** | The owner gets the request from the renter |

**10. Review**

|  |  |
| --- | --- |
| **Use case** | Review |
| **Goal** | House owner can easily give a review |
| **Preconditions** | Renter must send a request to the house owner |
| **Success End Condition** | Successful to review |
| **Failed End Condition** | Unsuccessful to review |
| **Primary Actors**  **Secondary Actors** | House owner  Renter |
| **Trigger** | House owner will see a request for his rental house, from the renter |
| **Description** | After seeing a request the house owner give a review to the renter |
| **Alternative Flows** | None |
| **Quality Requirements** | House owner must create a post for rent house |

**11. Booking**

|  |  |
| --- | --- |
| **Use case** | Booking |
| **Goal** | System will show the booked house |
| **Preconditions** | None |
| **Success End Condition** | Successful to booking |
| **Failed End Condition** | Unsuccessful to booking |
| **Primary Actors**  **Secondary Actors** | House owner, Renter  None |
| **Trigger** | The renter booked his required house from the house owner |
| **Description** | Committing the house to be rented that is viewed on the website |
| **Alternative Flows** | None |
| **Quality Requirements** | The renter gets the house notification for confirmation |

**12. Delete Post**

|  |  |
| --- | --- |
| **Use case** | Delete Post |
| **Goal** | Admin/House Owner delete the post |
| **Preconditions** | After booking or change of mind. |
| **Success End Condition** | Successful to delete a post |
| **Failed End Condition** | Unsuccessful to delete a post |
| **Primary Actors**  **Secondary Actors** | Admin, House owner  None |
| **Trigger** | System users can’t find the post |
| **Description** | Removing a house information from house list |
| **Alternative Flows** | None |
| **Quality Requirements** | After booking the house, house-owner or admin can delete the post from the system. |

**5.Activity Diagram:**

**Activity diagram for user signup:**

Enter all information

Enter all information

Is it valid?

Invalid fields

no

Yes

Signup successful

Open login page

**Activity diagram for Login:**

User is registered

Enter username & password

Invalid username & password

No

Is it correct?

Yes

Login Successful

**Activity diagram for Manage User & Manage Feature:**

Manage

Manage user

Manage feature

No

Is it ok?

Is it ok

No

Yes

Yes

Successful

**Activity Diagram for Create Post & Delete Post:**

Post

Create Post

Delete post

Is it created?

Is it deleted?

No

No

Yes

Yes

Successful

**Activity Diagram for House Booking System:**

Search House

Select area

No

Is it available?

Yes

View house

Is it available?

No

Yes

Send Request

See Request

No

Is it accepted?

Yes

Review

Is it final?

No

Yes

House booking successful

**Activity diagram for Logout:**

Click Log Out

Display logout message!

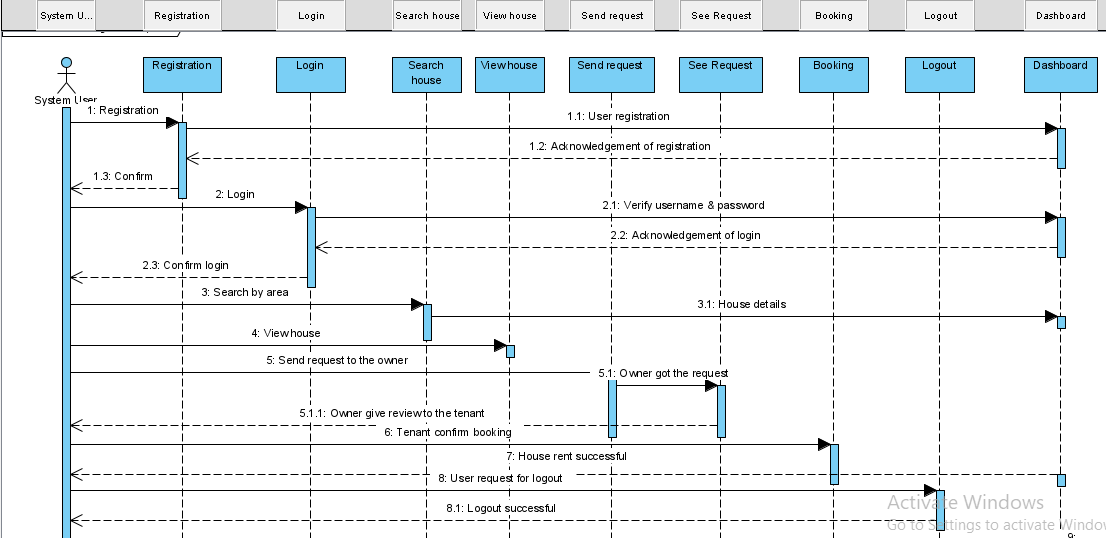
No

Continue

Yes

Logout Successful

**6.Sequence Diagram:**



**7.Entity Relationship Diagram:**

**House-Owner**

**Renter**

**House**

**Add**

**Registration**

**Give-Details**

**Rent**