SRS DOCUMENT

SYSTEM REQUIREMENTS AND SPECIFICATION

**Logo

Description automatically generated**

***Find rooms and roomies***

**Prepared By:**

**IT034 – Soumyadeep Ghosh**

**IT044 – Naamsukh Jobanputra**

**Table of Contents**

Table of contents………………………………………i

1. Introduction…………………………………………………..1
2. Purpose……………………………………………….1
3. Scope………………………………………………1
4. FUNCTIONAL REQUIREMENTS

4.1 Customers……………………………………………………….. 2

4.2 Features…………………………………………………………… 2

F.0 Types of users involved…………………………………………… 3

F.1 Basic functionalities for owners…………………………….. 4

F.2 Features for users………………………………………………….. 5

**1. Introduction**

Find-my-dorm will help you to find amazing and well maintain rooms for students and directly deal with the owner and even contact the room mates for convenience. Find-my-dorm can be a mode of business for many people and a utility for many students seeking accommodation.

**2. Purpose**

In today’s life, finding good PGs near colleges have become a hectic task. We have to contact third party dealers and go to see PGs personally which is tiresome and sometimes, futile. This application will provide ease of finding PGs to the users. They can find their PGs and book them by contacting the owner very easily which will somehow cut-off the cost for hiring middlemen.

**2.1 Customers**

Majority of the customers will be college students as well as the owners who are going to rent rooms to the students.

**3. Scope**

The scope of this system is to make student’s life easy . This application help students to find their PGs or dorms quickly without paying extra money to the brokers.

**4.Functional Requirements**

**4.1 Registration**

**Description**: In order to use the functionalities of our web application, the user must be registered.

**Pre-Condition**: The user should not be already registered. It has to be a new account.

**Input**: The user has to enter details i.e. Full Name, Email, Password and Confirm Password.

**Processing**: Checks whether all the data is valid(according to rules of database).

**Output:** Successfully registered user will be redirected to login page and if any error occurs then error will be shown on the same registration page with error message.

**4.2 LOGIN**

**Description**: It allows only authorized people to access the application. When the user logs in to the system, the user has to enter login details.

**Pre-Condition**: User must be registered.

**Input**: Enter email address and password.

**Processing**: Checks whether the user is already registered and the entered email address and password are correct or not.

**Output**: If valid, then the user is given access to the application and the user profile is displayed. Otherwise an error message is displayed indicating some incorrect input. Post-Condition: User is redirected to Home Page

**4.3 Logout**

**Description**: After having finished with the work, the user can log out using the logout button.

**Pre-Condition:** User must be logged in.

**Input:** Click on logout.

**Output:** User is redirected to Log in Page.

**Basic CRUD operations for house owners**

**4.4 Insert/Update/Delete new PGs and rental properties**

**4.4.1** House Owner can create new entry for the dorm (room/rooms) they want to rent and upload the pictures as well as the details, including house features and details and the payment details.

**Pre-Condition**: User must have an account.

**Input**: To add any dorm/PGs/rental property, the user has to specify whether it is a dorm/PG/rental property or a Hostel. Also , he/she needs to specify the facilities available in their property. The owner should specify the university which is nearest to the property.

**Processing**: Checks whether all the data satisfies the predefined rule of the database.

**Output**: Property will be added and other users can see the property. **Post-Condition**: Property will be registered in the database.

**4.4.2 Update the advertisements/post of PGs or rental properties**

**Description:** House owner can change his information (update), post advertisements as well as delete his account.

Pre-Condition: Property must be registered.

**Input:** You need to provide what changes you want to make.Also they can delete the property if they want.

**Processing**: It will check the changes.

**Output**: Property details will be updated.

**Post-Condition**: Property will be updated in the database.

**4.4.2 Delete the advertisements/post of PGs or rental properties**

**Description**: After putting the property on the site, if the owner wants to delete the property then he can.

**Pre-Condition:** Property must be registered.

**Input:** Click on delete property button.

**Output:** Property will be deleted.

**Post-Condition**: Property will be deleted from the database

**4.5 Searching dorms/PG’s with the help of their university**

**Description**: User can search property with the help of the University.

**Pre-Condition**: None.

**Input**: The user should enter University name .

**Processing**: Checks whether University meets the particular requirements and fetches it from the database and shows it.

**Output**: Display all the property specified to a particular University.

**4.6 Filtering dorms/PG’s through different amenities.**

**Description**: User can add particular amenities and get the properties on the basis of the amenities specified

**Pre-Condition**: None.

**Input**: The user should select amenities.

**Processing**: Checks whether property matches the amenities from the database .

**Output**: Display all the property specified to a particular University.

**4.7 Contact details of owner**

**Description:** User can get the contact details of the owner of the property.

**Pre-Condition**: None.

**Input**: User should solve the puzzle or captcha .

**Processing**: Checks whether puzzle is solved correctly or captcha is written correctly.

**Output**: Display all contact details of the owner.

**4.7 Rate/Comment on the dorm**

**Description:** User can rate/comment on the dorm if they want to.

**Pre-Condition**: None.

**Input**: User can give stars to the dorm/PG’s and also can comment about the property .

**Output**: “Thank you for the feedback” after giving the feedback.

Post-condition : Updates the rating / comment in the database.

**4.8 Admin**

**Description**: Only admin can login here with his registered email and password, admin can see number of users and other statistics about the website.

**Pre-Condition**: Admin must be registered. Input: Enter Email and password.

**Processing**: Checks whether email and password are valid or not. **Output**: if valid, then the admin is given access to the application and he can see his dashboard.

1. **Non-Functional Requirements**

• 24\*7 availability.

• The database used here is robust, reliable and fast. So users will have to wait for the output for a very short time.

**6. Development Responsibility**

**DR 1**

We, IT034 - Soumyadeep Ghosh and IT044 - Naamsukh Jobanputra are responsible for planning, design, development, testing and maintenance of the application, including UI Designing as well as designing the database.

**DR 1.1**

**DR 1.1.1 Planning Phase**

Both members of the unit are responsible for effective planning of the project and research to find problems of the real-life users and implement the functionalities regarding the same.

**DR 1.1.2 Development Phase**

IT-044 Naamsukh Jobanputra will be responsible for designing the application on the system level and will also design the UI and frontend functionalities.

IT034 Soumyadeep Ghosh will be responsible for the raw API designing and building the routes as backend performance.

**DR 1.1.3 Testing**

We plan to test this application using Web LOAD or Selenium. Both the developers are yet to learn this technology and this domain.

**DR 1.1.4 Reviewing**

According to the ‘review’ and ‘report bug’ functionalities in the beta version, the customers that use this application will suggest changes and report bugs while using this application.

This review will be collected automatically into a Google form.

It will be the responsibility of both the developers to study the reviews and analyze the bugs and plan necessary changes.

IT 034 Soumyadeep Ghosh is responsible for documenting the next versions.

IT 044 Naamsukh Jobanputra is responsible for making changes to the repository on GITHUB or BitWarden.