☰

Print

Software Requirements Specification

For

'BASHABHARA.COM'

Prepared By:

Name	ID
MD Yeasin Arafat	2022320642
Habibur Rahaman Fahim	1911753642
Mosroor Mofiz Arman	1921079642
Khadiza Akter Luna	1430452042

Table of Content

1. Introduction

1.1 Purpose:

The software aims to give potential buyers and sellers of real estate a platform to transact in real estate purchases, sales, and rentals. Brokers typically assist buyers and sellers in finding the ideal buyer or seller for a property. Yet both the buyer and the seller must pay money. To avoid this expense, a software A remedy is required to assist folks.

1.2 Document Conventions:

For headlines, the font "Times New Roman" in size 14 is used, with font sizes 12 and 14 for paragraphs and descriptions, respectively, as well as for subheadings. The font "Cambria Math" is also used.

1.3 Intended Audience and Reading Suggestions:

The developer, users, project manager, front desk agent, and content writer can all benefit from this SRS document. The SRS paper includes all the conditions and advantages about the project as well as the procedures that will aid in its development.

1.4 Product Scope:

The Rental website offers its consumers a sizable facility. It offers us a methodical strategy to handle our unused products, which can benefit them and be utilized to manage users and its rented products. Consumers (who uploaded the goods) and Administrator will use their product.

1.5 References:

SRS IEEE standard format.

2. Overall Description

2.1 Product Perspective:

This product is a rental house management system called "BASABHARA.COM." It is a part of the software engineering course assessment at North South University's Spring academic semester of 2023. The faculty member proposed the project idea while acting as a "client" with the student group members who performed the tasks as an imaginary software firm. The product is a new self-contained system that is neither a member of a product family nor a replacement for another system.

2.2 Product Functions:

The functionality of the product includes:

- Using the rental house management system eliminates the need for a third party to rent a house.
- Users can search for houses in different areas and locations without traveling to the location.
- Direct communication between the householders and renters of the house.
- Clarifications can be done easily.
- Saves time for both parties.
- Multiple available houses and renters to choose from.

2.3 User Classes and Characteristics:

The users of the application can be classified into two types. The ones who want to give a house for rent (house owners) and the ones who want to rent a house. There is no strict distinction between the two types of users. Both users can access all the functionality of the application.

2.4 Operating Environment:

The software will operate on web apps and android devices. The operating environment should have access to a microphone for voice recording. It is recommended not to use any old Android OS version and use a relatively modern web browser.

2.5 Design and Implementation Contraints:

Constraints include:

- The time allotted for this project is, at most, six months.
- The application's front end will be made using HTML, CSS, and JavaScript.
- Python will be used as the language for the application's back-end, and PostgreSQL will be used for the application's database.
- The website will be in the English language. Users who do not know English will need help using the website.

2.6 User Documentation:

The software comes with a user manual and a short video for non-technical individuals that explains "how to install and use the software."

2.7 Assumptions and Dependencies:

The software assumes that the user's devices will have access to the internet that can connect to the internet, have a google account, can use a web app and android app, and have access to necessary devices to use those interfaces. It is also assumed that the user has access to a microphone that can record voices. This software relies on APIs such as Google OAuth API and external ML speech-to-text APIs, So the software's one of the core differentiators relies on the external ML model's performance, and it is assumed that the ML model's performances are reasonable. The project could be affected if these dependencies are incorrect, are not shared, or are changed.

3. External Interface Requirements

3.1 User Interfaces

WebApp:

- Front-end: HTML5, CSS3, Bootstrap, JavaScript.
- Back-end: Python, SQLite, Django framework.

Mobile App:

- Java

Hardware Interfaces

The primary hardware requirement for the system is a computer or a mobile phone. The system will require microphone permission to assign and give a voice command and an internet connection to call API.

3.3 Software Interfaces

This system will use ORM to map relational cardinalities. It will use a Speech-to-text machine learning model API to detect questions and answers for the voice command. This system will also use Google OAuth 2.0 to log in or register to the app with google.

3.4 Communications Interfaces

The system will use Restful API to connect the front-end with the back-end. This system uses OAuth 2.0 by Google for verification and token authentication to communicate between the front-end and back-end through API.

4. System Features

4.1 Description and Priority:

Description: Users can Sign Up and Login from application or website from anywhere and anytime.

Priority:

Description: Users get live GPS location to track down the location.

Priority: Medium

Description: Flat owners will have to upload a photo of their flat for advertisement.

Priority: High

Description: Messaging system between buyer and seller.

Priority: High

Description: All users can use voice commands to create accounts or publish advertisements.

Priority: High

4.2 Stimulus/Response Sequences:

Stimulus: Admin logins with preferred ID and Password into the Admin Panel.

Response: System provides all Real-Time User specifications to the Admin.

Stimulus: User (Customer) Signs up and Login to the system using their email and other necessary information.

Response: System provides flat advertisements and details to the User.

Stimulus: User (Renter) Signs up and Login to the system with their information, also have to provide any photo of that advertisement.

Response: System publish the advertisement into the website.

Stimulus: User request for location to the system.

Response: System provides live GPS location to track the location.

Stimulus: When User can book flats from the system.

Response: The system sends a notification to User (Customer), with GPS location.

Stimulus: Renters can upload flat photo from anywhere via email.

Response: The system adds the photo for the advertisement.

4.3 Functional Requirements:

FR IDs	Feature Name	Description
FR-1	User Authentication	Users can Sign up, Login and Reset Password if they want.
FR-2	Listing	Renters can publish their advertisement with necessary information, so that buyers can search them easily.
FR-3	Messaging	Interaction between buyers and sellers is important. They can communicate with each other through messaging system.
FR-4	Photo Upload	FR-4 Photo Upload Flat owners need to upload their flat photos for advertisement.
FR-5	Voice Interaction	All users can fill up their necessary information or can search using voice interactions.
FR-6	GPS	Buyers should able to track down actual location via GPS.

5. Other Non-Functional Requirements

5.1 Performance Requirements

All circumstances will be covered by the system. It will operate nonstop for the entire calendar year. Software would be ready when promised, preventing any disruption from happening. The system's structure will be chosen in accordance with the requirements. The system's user interface (UI) will be simple and easy to understand so that users won't have any trouble identifying the functions. The software package should have flawless operation across its whole feature set.

5.2 Safety Requirements

When software crashes by default, the requirement should be used so that we have another option that still preserves data on multiple servers. Make sure you abide by international laws and rules when managing the hotel system, and get certified to ensure all safety requirements are met.

5.3 Security Requirements

To ensure the safety and security of the product's data, the system must be registered with the security-providing company and should be approved by a genuine identity. To protect the data from market hackers, a high level of network security should be implemented. Any legitimate accredited network security business, such as Fire Eye, should certify software.

5.4 Software Quality Attributes

Software should be dependable and adaptable, meaning that it can be altered on demand by the client without any issues. It should operate smoothly and be quick to maintain. To save time and reduce complexity, the data structures should be built correctly, and design patterns should be used when appropriate. Maintaining a software's portability and designing it so that it can be utilized for upcoming projects and updates are both required.

5.5 Business Rules

Users should be able to run software with a variety of individual privileges provided by the system. Each function stands alone from the others. Each user's portal needs to be unique based on their

rights.

6. Other Requirements

Here are some additional requirements relevant with the project.

- Data Requirements.
- Multi Language System.
- Feedback.
- Security.
- Legal Requirements.

Appendix A: Glossary

Customer: Who looks for flat.

Flat Owner: who advertise for flats for the customers.

Account: Customer & Renters will have their own account.

Voice Commands: Users can interact with system via voice command.

GPS: Customers can find flat through GPS.

Photo Attachment: Renters can add photo from anywhere anytime through email.

Appendix B: Analysis Models

Use-Case Diagram: