<Rental Buddy and Home services >

Software Requirements Specification

<25-04-2023>

Team Members-

Indrajeet Bhardwaj

Ekta Dewangan

T Sanjeevan

**Table of Contents**

Revision History ii

Document Approval ii

1. Introduction 1

1.1 Purpose 1

1.2 Scope 1

1.3 Definitions, Acronyms, and Abbreviations 1

1.4 References 1

1.5 Overview 1

2. General Description 2

2.1 Product Perspective 2

2.2 Product Functions 2

2.3 User Characteristics 2

2.4 General Constraints 2

2.5 Assumptions and Dependencies 2

3. Specific Requirements 2

3.1 External Interface Requirements 3

3.1.1 User Interfaces 3

3.1.2 Hardware Interfaces 3

3.1.3 Software Interfaces 3

3.1.4 Communications Interfaces 3

3.2 Functional Requirements 3

3.2.1 <Functional Requirement or Feature #1> 3

3.2.2 <Functional Requirement or Feature #2> 3

3.3 Use Cases 3

3.3.1 Use Case #1 3

3.3.2 Use Case #2 3

3.4 Classes / Objects 3

3.4.1 <Class / Object #1> 3

3.4.2 <Class / Object #2> 3

3.5 Non-Functional Requirements 4

3.5.1 Performance 4

3.5.2 Reliability 4

3.5.3 Availability 4

3.5.4 Security 4

3.5.5 Maintainability 4

3.5.6 Portability 4

3.6 Inverse Requirements 4

3.7 Design Constraints 4

3.8 Logical Database Requirements 4

3.9 Other Requirements 4

4. Analysis Models 4

4.1 Sequence Diagrams 5

4. 2Data Flow Diagrams (DFD) 5

4.3 ENTITY RELATIONSHIP DIAGRAM 5

A. Appendices 5

A.1 Appendix 1 5

A.2 Appendix 2 5

# 1. Introduction

## In this competitive world we all are facing numerous challenges and one of the major problems is finding a suitable residence in a new location with proper facilities like basic furniture’s and other household services. We need a platform that can resolve this all our problems in one platform.

## 1.1 Purpose

## The purpose of this rental website is to provide a platform for people who are looking to rent homes in new places. The website aims to provide a simple and easy-to-use interface for users to search for and find suitable rental properties, as well as other services like they can customized their home by adding furniture, vehicle, or other homes services.

## 1.2 Scope

## The rental website will allow users to search for and find rental properties based on their preferences, such as location, price range, nearby location, customization and number of rooms. The website will also provide information on other rental services, such as maintenance, Repairing, Cooking, Cleaning etc.

**1.3 Target Audience**

## The target audience for this rental website is people who are looking to rent homes in new places. This includes individuals, families, Students and businesses who require rental properties for a short or long-term basis.

## 1.4 References

This subsection should:

(1) Provide a complete list of all documents referenced elsewhere in the SRS, or in a separate, specified document.

(2) Identify each document by title, report number - if applicable - date, and publishing organization.

(3) Specify the sources from which the references can be obtained.

This information may be provided by reference to an appendix or to another document.

## 1.5 Overview

# The rental website will have a user-friendly interface that will allow customers to search for rental homes or apartments.

# The website will also have a feature that will enable users to Customize their home by own preferences and save their search criteria.

# The website will also provide property owners with a platform to manage their properties, view tenant information, and access rental reports.

# 2. General Description

This section of the SRS should describe the general factors that affect 'the product and its requirements. It should be made clear that this section does not state specific requirements; it only makes those requirements easier to understand.

## 2.1 Product Perspective

This subsection of the SRS puts the product into perspective with other related products or

projects. (See the IEEE Guide to SRS for more details).

## 2.2 Product Functions

This subsection of the SRS should provide a summary of the functions that the software will perform.

## 2.3 User Characteristics

This subsection of the SRS should describe those general characteristics of the eventual users of the product that will affect the specific requirements. (See the IEEE Guide to SRS for more details).

## 2.4 General Constraints

This subsection of the SRS should provide a general description of any other items that will

limit the developer’s options for designing the system. (See the IEEE Guide to SRS for a partial list of possible general constraints).

## 2.5 Assumptions and Dependencies

This subsection of the SRS should list each of the factors that affect the requirements stated in the SRS. These factors are not design constraints on the software but are, rather, any changes to them that can affect the requirements in the SRS. For example, an assumption might be that a specific operating system will be available on the hardware designated for the software product. If, in fact, the operating system is not available, the SRS would then have to change accordingly.

# 3. Specific Requirements

## 3.1 External Interface Requirements

### 3.1.1 User Interfaces

## The system will have a search function that allows users to search for rental properties based on location, price, customization, nearby location and other criteria.

## The system will allow property owners to post their properties for rent and manage their listings.

## The system will allow users to view details about each property, including photos, descriptions, and rental terms.

## The system will have a messaging feature that allows users to communicate with our Help center.

## The system will provide a payment gateway that allows users to make secure payments for rental services.

## The system will have a feedback and rating system that allows users to rate their experience with a particular property or property owner.

## The system will be accessible on multiple devices, including desktop computers, laptops, tablets, and smartphones.

## The system will be able to handle a large amount of traffic and data without compromising performance or security.

## The system will follow all relevant data protection and privacy regulations.

### 3.1.2 Hardware Interfaces

As mentioned in the SRS, the rental website will be implemented as a web-based system, which implies that users will access it through a web browser. Therefore, there is no specific hardware interface requirement for the system. However, the system should be compatible with commonly used web browsers such as Google Chrome, Mozilla Firefox, Safari, and Internet Explorer. It should be designed to be responsive, so it can adjust to different screen sizes and resolutions, including mobile devices.

### 3.1.3 Software Interfaces

**Front-end Requirements:**

* The website Will be built using HTML, CSS, and JavaScript.
* The website Will be mobile responsive and Will work on all devices.
* The website Will have a user-friendly interface with easy navigation.
* The website Will have a search bar to enable users to search for properties based on different criteria.
* The website Will allow users to create an account, log in, and log out.
* The website Will have a dashboard for users to manage their profiles, saved properties, and rental applications.

**Back-end Requirements:**

* The website Will use a server-side programming language such as Python.
* The website Will use MySQL for relational database management system.
* The website Will be hosted on a secure server.
* The website Will use HTTPS protocol to ensure secure communication.
* The website Will implement user authentication and authorization.
* The website Will have a system for administrators to manage properties, rental applications, and users.
* Third-party Integrations:
* The website Will integrate with a payment gateway to enable users to make payments securely.
* The website Will integrate with Google Maps API to show property locations.
* The website Will integrate with social media platforms to allow users to share properties and the website.

**Performance Requirements:**

* The website Will have a fast-loading time.
* The website Will handle a large number of users simultaneously.
* The website Will be scalable and able to handle a growing number of properties and users.
* The website Will have a backup and recovery system in case of a system failure.

## 3.2 Functional Requirements

### A search feature that will allow users to search for rental homes or apartments based on various criteria such as location, price, customization, nearby location,

### number of bedrooms, and more.

### A registration feature that will enable users to create an account, save their search criteria, and receive notifications when a new property

### that matches their criteria becomes available.

### A property management feature that will allow property owners to manage their properties, view tenant information, and access rental reports.

### A lease agreement feature that will allow property owners to generate lease *agreements for their tenants*

## 3.3 Non-Functional Requirements

Non-functional requirements may exist for the following attributes. Often these requirements must be achieved at a system-wide level rather than at a unit level. State the requirements in the following sections in measurable terms (e.g., 95% of transaction shall be processed in less than a second, system downtime may not exceed 1 minute per day, > 30-day MTBF value, etc.).

### 3.5.1 Performance

The website should be able to handle a large number of concurrent users without slowing down or crashing.

### 3.5.2 Usability

The website should have a user-friendly interface that is easy to navigate and understand.

### 3.5.3 Accessibility

The website should be accessible to users with disabilities, with support for screen readers and other assistive technologies.

### 3.5.4 Security

The website should use secure encryption methods to protect user data and payment information.

## 3.6 Inverse Requirements

1. The system must not allow unauthorized access to user data.
2. The system should avoid any features that could cause user confusion or frustration.
3. The system must not create security vulnerabilities or provide an easy target for cyber-attacks.
4. The system should avoid any features that may infringe on users' privacy.
5. The system must not crash or experience downtime that could cause loss of data or business.
6. The system should avoid any features that may negatively impact system performance or scalability.
7. The system must not violate any legal or regulatory requirements.
8. The system should avoid any features that may be considered offensive or discriminatory.
9. The system must not rely on third-party services that are known to be unreliable or insecure.
10. The system should avoid any features that may be susceptible to software bugs or errors.

## 3.7 Design Constraints

1. Platform or Technology Constraints: The system must be designed to work on a specific platform or technology, such as a particular operating system, web browser, or hardware device.
2. Performance Constraints: The system must be designed to meet specific performance criteria, such as response time, processing speed, or memory usage, to ensure efficient and effective operation.
3. Resource Constraints: The system must be designed to work within specified resource limitations, such as memory, storage, or bandwidth, which may impact the system's functionality and performance.
4. Budget Constraints: The system must be designed to adhere to a specific budget, which may affect the choice of technologies, features, and development resources.
5. Time Constraints: The system must be designed to be completed within a specific timeline or deadline, which may impact the scope and complexity of the design.
6. Compliance Constraints: The system must be designed to comply with specific industry standards, regulations, or legal requirements, such as data privacy, security, or accessibility standards.
7. User Experience (UX) Constraints: The system must be designed to provide a positive user experience, taking into consideration factors such as ease of use, intuitiveness, and accessibility for different user groups.
8. Design Guidelines or Standards: The system must be designed in accordance with established design guidelines or standards, such as branding guidelines, UI/UX design principles, or coding standards.
9. Integration Constraints: The system must be designed to integrate with specific external systems or APIs, which may require adherence to certain integration protocols or data formats.
10. Existing System Constraints: The system must be designed to work with or integrate into an existing system or infrastructure, taking into consideration any limitations or constraints imposed by the existing system.

## 3.8 Logical Database Requirements

1. Data Entities: Identify the entities (e.g., customers, products, orders) that need to be represented in the database, along with their attributes (e.g., name, address, price) and relationships (e.g., one-to-many, many-to-many) with other entities.
2. Data Integrity: Define the rules and constraints that ensure the accuracy, consistency, and integrity of the data in the database, such as primary key constraints, foreign key constraints, and data validation rules.
3. Data Retrieval: Specify the requirements for retrieving data from the database, including the types of queries, reports, and views that need to be supported, as well as any performance or response time requirements for data retrieval operations.
4. Data Storage: Define the requirements for storing data in the database, including the size, format, and structure of the data, as well as any data retention or archival requirements.
5. Data Security: Specify the requirements for securing the data in the database, such as access control mechanisms, authentication, and encryption, to ensure the confidentiality, integrity, and availability of the data.
6. Data Backup and Recovery: Define the requirements for backing up and recovering data in the database, including the frequency, scope, and storage location of database backups, as well as the procedures for restoring data in case of data loss or system failures.
7. Data Migration: Specify the requirements for migrating data from existing databases or legacy systems to the new database, including data conversion, data mapping, and data validation requirements.
8. Performance and Scalability: Define the performance and scalability requirements for the database, including the expected number of concurrent users, transaction volumes, and response times, as well as any performance tuning or optimization requirements.
9. Database Administration: Specify the requirements for managing and administering the database, including user management, backup and recovery procedures, database monitoring, and performance tuning.
10. Interoperability: Define the requirements for interoperability with other systems or databases, including data exchange formats, data integration mechanisms, and data synchronization requirements.