Software Requirements Specification

for

SampleName

Version 1.0 approved

Prepared by <author>

<organization>

<date created>

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

The purpose of SRS is to describe the specification & description of our project " PROJECT NAME". It will illustrate the purpose & features of our software , the interfaces of the software & the restrictions under which it must operate. This document is meant for users of the software & also the potential developers.

1.2 Document Conventions

<Describe any standards or typographical conventions that were followed when writing this SRS, such as fonts or highlighting that have special significance. For example, state whether priorities for higher-level requirements are assumed to be inherited by detailed requirements, or whether every requirement statement is to have its own priority.>

1.3 Intended Audience and Reading Suggestions

<Describe the different types of reader that the document is intended for, such as developers, project managers, marketing staff, users, testers, and documentation writers. Describe what the rest of this SRS contains and how it is organized. Suggest a sequence for reading the document, beginning with the overview sections and proceeding through the sections that are most pertinent to each reader type.>

1.4 Product Scope

<Provide a short description of the software being specified and its purpose, including relevant benefits, objectives, and goals. Relate the software to corporate goals or business strategies. If a separate vision and scope document is available, refer to it rather than duplicating its contents here.>

1.5 References

<List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location.>

2. Overall Description

2.1 Product Perspective

SampleName is developed for people who want to rent their houses and view properties available for renting. It uses a real-time database for up-to-date information about the listings, and the base of the app
<Describe the context and origin of the product being specified in this SRS. For example, state whether this product is a follow-on member of a product family, a replacement for certain existing systems, or a new, self-contained product. If the SRS defines a component of a larger system, relate the requirements of the larger system to the functionality of this software and identify interfaces between the two. A simple diagram that shows the major components of the overall system, subsystem interconnections, and external interfaces can be helpful.>

2.2 Product Functions

- **Register new house:** Creates a new house at the location of a cursor placed.
- **Notifications:** See if anybody is interested in renting my property
- **Remove listings:** removes one or more listings that the user has posted previously

<Summarize the major functions the product must perform or must let the user perform. Details will be provided in Section 3, so only a high level summary (such as a bullet list) is needed here. Organize the functions to make them understandable to any reader of the SRS. A picture of the major groups of related requirements and how they relate, such as a top level data flow diagram or object class diagram, is often effective.>

2.3 User Classes and Characteristics

<Identify the various user classes that you anticipate will use this product. User classes may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. Distinguish the most important user classes for this product from those who are less important to satisfy.>

2.4 Operating Environment

<Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist.>

Google Chrome
Firefox
Microsoft Edge Chromium
Internet Explorer
Safari

2.5 Design and Implementation Constraints

<Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer's organization will be responsible for maintaining the delivered software).>
We use Javascript for the website, and an open source javascript API(Mapbox) to handle the map drawing queries, and Amazon Web Services: Firestore as a database for real time communication with the application.

2.6 User Documentation

<List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.>

2.7 Assumptions and Dependencies

<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project, unless they are already documented elsewhere (for example, in the vision and scope document or the project plan).>

3. External Interface Requirements

3.1 User Interfaces

The program will be in the form of a website, on launching the website, the user will be greeted with a Login Screen.

1. Login Screen:
   //image

   On logging in, he will be redirected to the main screen.

2. Main Screen:
   //image

3. On clicking a marker:
   //image

4. House Registration Form:
   //image

5. Notification Button:
   //image

6. Search Interface:
   //image

3.2 Hardware Interfaces

The minimum system requirements for SampleName is a Dual core CPU with at least 1.0 GB of RAM. Because the Map will render components using WebGL, the system must have a graphics card or hardware acceleration enabled that supports WebGL.

3.3 Software Interfaces

SampleName requires a modern web browser that supports Javascript to run.

3.4 Communications Interfaces

SampleName requires an internet connection to send, retrieve, and update data to a Real-time Database.

4. System Features

This section exhibits the features of SampleName, including demonstrations on how they are used in a realistic setting.

4.1 Display Houses available for Renting

It is a major feature of SampleName to display up-to-date and accurate information about the houses available for renting.

The houses are displayed as Markers on a Map. On logging in, the system queries the database and populates the map with information about currently available properties available for renting.

The user must have an internet connection to communicate with the database, if not, a connection error message is displayed. (REQ_Communication_Error)

4.2 Display House Information on Clicking Marker

4.3 Send Notification to house owner if user clicks Interested

4.4 View Notifications

4.5 Filter Houses based on Rent

4.6 Filter Houses based on Number of Rooms

4.7 Functionality to Add Images of House

4.8 Responsive website with Lightweight UI

4.9 System Feature 9

4.10 System Feature 10

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The minimum system requirements for SampleName is a Dual core CPU with at least 1.0 GB of RAM. Because the Map will render components using WebGL, the system must have a graphics card or hardware acceleration enabled that supports WebGL.

5.2 Safety Requirements

Here none of this app user loses any data while using this app due to crash or some Bug.in this app there is a tracker is available where user can report bugs they have encountered so that the developers can fix in the next release.

5.3 Security Requirements

in this app dosen't have any security requirements.

5.4 Software Quality Attributes

It provides user both simple and advanced features.Its well designed and easy use to interface. its use can anyone.

5.5 Business Rules

<List any operating principles about the product, such as which individuals or roles can perform which functions under specific circumstances. These are not functional requirements in themselves, but they may imply certain functional requirements to enforce the rules.>

6. Other Requirements

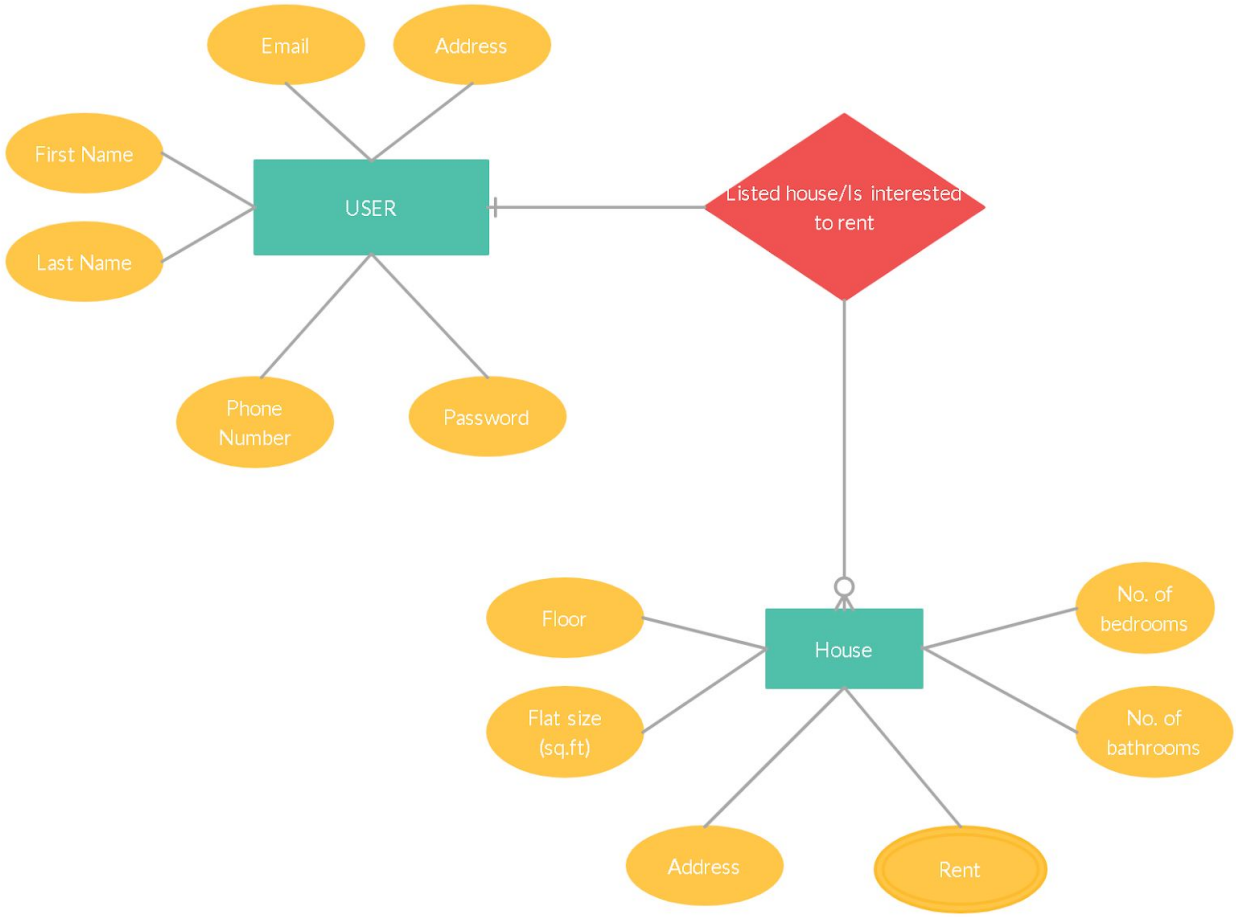
Appendix A: Glossary

References: https://www.wikipedia.org/

- Attributes: An attribute is a name paired with a domain (nowadays more commonly referred to as a type or data type). In our application, it refers to the features of the house that is listed on the map.
- Markers: A marker displays the geographical location of a house/listing that is listed on the map.
- Relationships: Relationships are a logical connection between different tables, established on the basis of interaction among these tables.
- Real-time database: A real-time database is a database system which uses real-time processing to handle workloads whose state is constantly changing. This differs from traditional databases containing persistent data, mostly unaffected by time. For example, a stock market changes very rapidly and is dynamic.
- User Interface: The user interface or *human–machine interface* is the part of the machine that handles the human–machine interaction.

Appendix B: Analysis Models

This is an entity-relationship diagram for our project.



Appendix C: To Be Determined List

This is a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.

1. System Features:
    a. System Feature 9
    b. System Feature 10

2. User Interface:
    a. Images need to be added for the working application