Software Requirements Specification

for

BharaTiya

Version 1.1 approved

Prepared by: Salman Sahel, Durdana Kamal, Shahriar Jabin Abir, Sadia Mahmood, Sifat Anwar

CSE 327 Section 10 Group 4

13th March 2021

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Shahriar Jabin Abir	3/21/2021	Some features were added for development purpose	1.1

1.   Introduction

1.1   Purpose

The purpose of SRS is to describe the specification & description of our project " BharaTiya". 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

We created this document based on the IEEE template for the System Requirement Specification Documents.

1.3   Intended Audience and Reading Suggestions

Two types of users will get this service:

1. *Users who are looking for house rent.*
2. *Users who want to rent their house.*

1.4   Product Scope

BharaTiya is an application useful for everyone. Some people are looking for houses for rent and some people are looking for proper renters to rent their houses. BharaTiya is a web application which will bring them closer.
Renter will fully fill their confusion by watching all the facilities and photos of the house with proper fare. And everything by a single click from a cell phone, laptop, or desktop, sitting inside home.

1.5   References

1. *IEEE Template for System Requirement Specification Documents:*
   https://goo.gl/nsUFwy

2. Overall Description

2.1 Product Perspective

BharaTiya 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

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

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

Google Chrome
Firefox
Microsoft Edge Chromium
Internet Explorer
Safari

2.5 Design and Implementation Constraints

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 BharaTiya 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

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

3.4   Communications Interfaces

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

4.   System Features

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

4.1   Display Houses available for Renting

It is a major feature of BharaTiya 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 available where users can report bugs they have encountered so that the developers can fix it in the next release.

5.3   Security Requirements

This app doesn't have any security requirements.

5.4   Software Quality Attributes

It provides users both simple and advanced features.Its well designed and easy use to interface.

5.5   Business Rules

this application is used to the both simple and advanced features.this application need a good start in the development process.it's use to look for a solution to find house,as well as anybody who will need help in this regards.
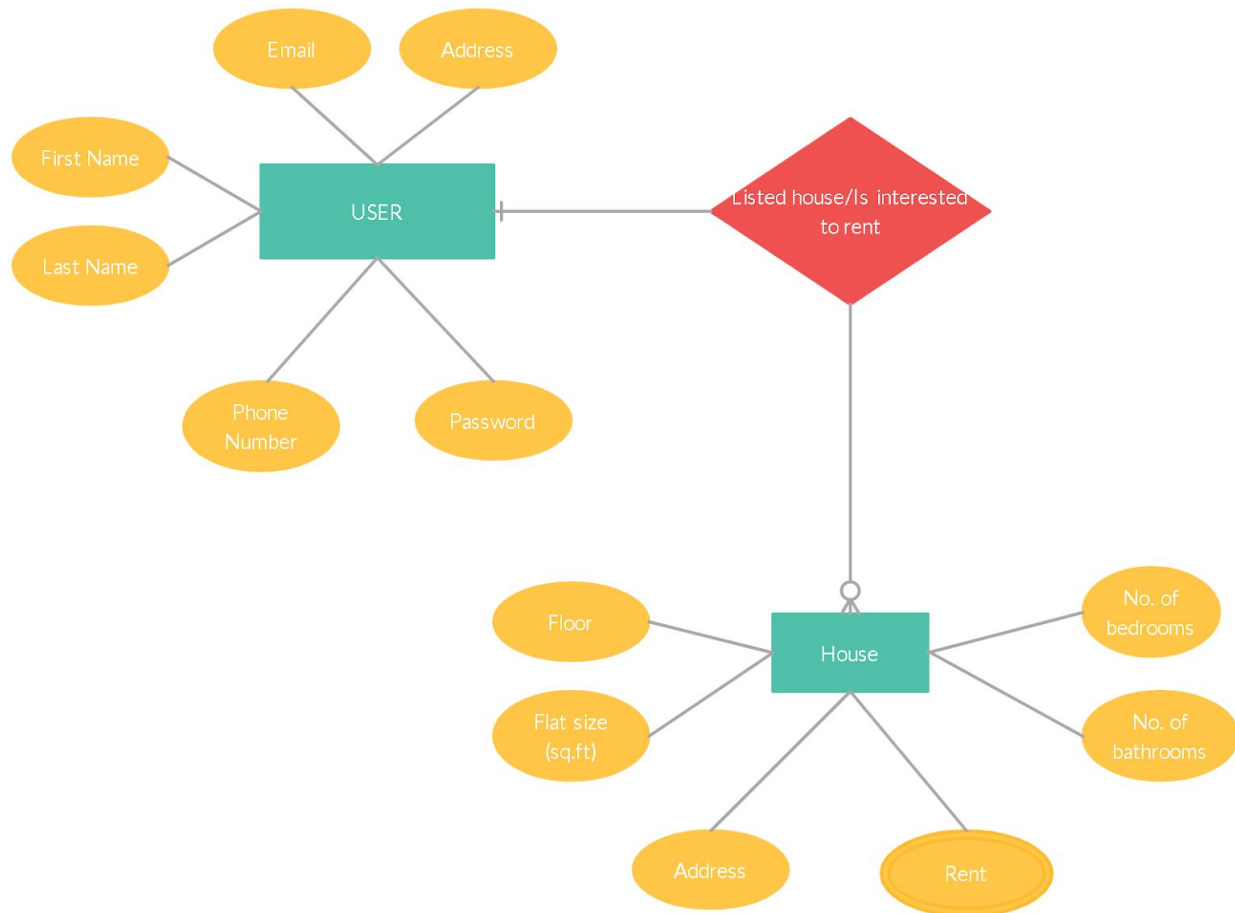
6.   Other Requirements

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