Dhirubhai Ambani
Institute of Information and Communication Technology

Software Requirements Specification

for

Where Should You Live?

by Group – 34

Table of Contents

1. Introduction

1.1 About Project

The need for this project is simple: people need a better way to make informed decisions about where to live. 99acres provides valuable information on property valuations, but it falls short on providing a complete picture of a neighborhood - and let's be real, you want to know more than just how much your future home is worth. People need to know about school district quality, crime rates, local amenities, and more. Furthermore, relying solely on real estate listings doesn't give the whole story. That's where our project comes in - we're going to provide a one-stop-shop for all things "where to live" and let the people provide their two cents on the neighborhoods they know and love (or loathe).

1.2 Needs for the Project

1. **Access to property information**
    - People who need to find a place to live or do business don't necessarily have to go there in person to receive the information.
    - On a single app, they may find all the information about neighborhoods.
2. **Time saving**
    - It takes a lot of effort to gather information about numerous locations before selecting the best one.
    - This app gives users a location score based on their preferences, enabling them to choose more effectively and quickly.
3. **Budget friendly**
    - To receive the information you need and to find a place that fits your budget, you will need a broker. The cost is raised by this third party's involvement.
    - However, with the use of this app, a user can purchase a home without the assistance of a broker, making it more affordable.
4. **Security**
    - Due to the rising crime rate, it is difficult for users to verify all the property's paperwork and trust the owner.
    - This app will only include the verified locations, thus increasing the security of the user.
5. **Convenience**
    - This app will be accessible round-the-clock and from anywhere, making it easier for users to look for their dream home at any time or location.
6. **Lack of User Feedback**:
    - Existing solutions do not provide the ability for users to share their experiences and provide feedback on different neighborhoods.
    - This makes it challenging for people to get a sense of the pros and cons of different areas and make informed decisions.

1.3 Intended Audience

The intended audience for this srs document is:

- Developers
- Testers
- Users
    - Admin
    - General User

1.4 Features

1. **User-account**
    - User accounts are commonly used in software development projects to facilitate collaboration between developers, project managers, and stakeholders.
    - Here in our project the user account will ask for various details like username, user-email, password to create an account on our project.
2. **Wishlist**
    - This list will contain all the neighbourhoods that one particular user has selected and kept aside before coming to final conclusion.
3. **User rating**
    - Users can rate neighbourhoods on various parameters such as the location, amenities, price, and quality of construction, among others.
    - The rating feature is an important tool that helps users make informed decisions by providing valuable insights and feedback from other users.
    - Users can provide feedback on various aspects of the service, such as the ease of use of the platform, the accuracy of property listings, etc.
4. **Filters**
    - Allows users to narrow down a large set of data to a more manageable subset based on specific criteria such as education, safety, Health care etc.
    - They can also help to improve the efficiency and effectiveness of the application by reducing the amount of data that needs to be processed and displayed.
    - The choice of implementation will depend on the specific requirements of the application of the users.
5. **Recommendation**
    - Provides users with personalized suggestions or recommendations based on their previous search histories.
    - This feature can help to improve the overall user experience by reducing the time and effort required to find new and relevant content.

2. Functional & Non-Functional Requirements

2.1 Functional Requirements

1. **User Registration and Login**: A system that enables users to create an account and log in to access the application.
2. **Google Registration**: A feature that streamlines the registration process by using Google logins.
3. **Profile Management**: Users must be able to update their profile information at any time, including their contact details, first name, last name, and occupation.
4. **Personalized Recommendations**: The system should be able to provide personalized recommendations based on the user's profile information, such as their Searched location city and other preferences.
5. **City Selection**: A feature that allows users to select a city by searching for it.
6. **City Statistics**: A system that displays basic statistics about the selected city, such as education, crime rate, cost of living, etc.
7. **User Wishlist**: The system must provide users with a dashboard displaying their liked neighborhoods.
8. **Neighbourhood Browse**: A feature that enables users to view different neighbourhoods in the selected city and see their good and bad points.
9. **Neighbourhood Save**: A feature that allows users to save preferred neighbourhoods for future reference.
10. **Search and Filter Options**: The system must provide users with various search and filter options to help them find a suitable neighbourhood .
11. **Neighborhood Rating**: The system must allow users to share their experiences by rate on different neighbourhoods.

2.2 Pending Functional Requirements

1. **Neighbourhood Addition**: Admin has the right to add new neighborhoods and update existing neighborhoods.
2. **Real Estate Integration**: An integration that allows users to connect with local real estate agents or real estate websites to purchase houses quickly and seamlessly. Additionally, users can access a range of resources and information to help them make an informed decision about their home purchase.
3. **Comparison Feature**: A feature that enables users to compare different neighborhoods side by side.
4. **Sharing Information**: A system that enables users to share neighbourhood information with friends and family.

2.3 Non-Functional Requirements

1. **User-Friendly Interface**: A system that has a user-friendly interface that is easy to navigate and understand.
2. **Device Compatibility**: The system must be compatible with a variety of devices, including desktops, laptops, and mobile devices.
3. **Responsive Design**: A system that has a responsive design that provides optimal viewing on different mobile devices.
4. **Maintainability**: The system must be designed to be easily maintainable, allowing for bug fixes and updates to be made as needed.

These requirements will provide a clear and comprehensive guide for the development of the project solution and ensure that it meets the needs of users and provides all the necessary information to help them make informed decisions about where to live.

2.4 Pending Non-Functional Requirements

1. **Reliability**: The system must be reliable and always provide accurate information.
2. **Scalability**: The system must be designed to be scalable, allowing for future improvements and additions to be made as needed.
3. **Fast Loading**: A feature that ensures fast and efficient loading of city and neighbourhood information. The system must provide quick and reliable search results.
4. **Security**: The system must protect the privacy and security of user data, including user accounts and personal information.

3. Use Case

3.1 User As Actor

Sign Up

Login

Select City

View City

Rate

Filter

Wishlist

3.2 Admin As Actor

Edit Score of city

4. User Stories

4.1 User Stories for Functional Requirements:

1. As a **user**, I want to be able to **create a new account by providing a username, email, and password**, so that I can **access the application.**

   Back of card:

   - Title: **User Account Creation**
   - Acceptance Criteria:
     - User should be able to create an account by providing a username, valid email, and secure password.
     - Upon successful account creation, the user should be directed to app Home page.
     - In case of an error, the user should receive an appropriate error message.
2. As a **user**, **I want to be able to log in my account,** so that I can **access my saved neighbourhood and account details securely.**

   Back of card:

   - Title: **User Login**
   - Acceptance Criteria:
     - User should be able to log in with their username and password.
     - Upon successful login, the user should be directed to their app home page.
     - User can log in with the Google option.
     - In case of an error, the user should receive an appropriate error message.
     - User should be able to reset the password by clicking on forgot password.
3. As a **user**, I want to be able to **edit my profile details like name and occupation and etc**, so that I can **update my information as needed.**

   Back of card:

   - Title: **Profile Management**
   - Acceptance Criteria:
     - User should be able to access their profile and update their name and other details.
     - Upon successful update, the user should be directed to their updated profile page.
     - In case of an error, the user should receive an appropriate error message.

4. As a **user**, I want to be able access **Wishlist by adding neighbourhoods that I am interested in**, so that I can **keep track of my liked neighbourhoods.**

   Back of card:

   - Title: **Neighbourhood Wishlist**
   - Acceptance Criteria:
     - User should be able to add and remove neighbourhoods from their Wishlist.
     - User should be able to view and manage their Wishlist from their app home page.
5. As a **user**, I want to **filter neighborhoods based on specific criteria like education, safety, Health Care, etc.** so that I can **find the perfect property that meets my needs.**

   Back of Card:

   - Title: **Filters**
   - Acceptance Criteria:
     - Users should be able to access the filter feature from the dashboard or the search page.
     - Users should be able to choose from a list of filter criteria, such as education, safety, Healthcare etc.
     - Users should be able to view the filtered neighbourhoods based on their chosen criteria.
6. As a **user**, I want to **receive personalized recommendations based on my search history and preferences** so that I can **quickly find the perfect neighbourhood.**

   Back of Card:

   - Title: **Recommendation**
   - Acceptance Criteria:
     - Users should be able to access the recommendation feature from the dashboard.
     - Users should be able to view a list of recommended neighbourhoods based on their search history.
     - The recommendations should be based on factors like City location and other factors.
7. As a **user**, I want to be able **to rate for neighbourhoods based on my experience and observations.**

   Back of card:

   - Title: **Neighborhood Rating**
   - Acceptance Criteria:
     - User should be able to rate neighborhoods based on specific criteria like safety, Healthcare, Education, cost of living and etc.
     - User should be able to view overall ratings.

4.2 User Stories for Pending Functional Requirements:

1. As a **user**, I want to **connect with local real estate agents or websites to purchase homes quickly and seamlessly** so that I can **find my dream home without any hassle.**

   Back of Card:

   - Title: **Real Estate Integration**
   - Acceptance Criteria:
     - Users should be able to access the real estate integration feature from the dashboard.
     - Users should be able to browse through a list of local real estate agents or websites.
     - Users should be able to contact the agents or websites directly from the application.
     - Users should be able to access resources and information, such as property listings, pricing information, and financing options.
2. As an **Admin**, I want to be able to **add new neighborhoods to the system and update existing ones** so that **users have access to the latest and most accurate neighborhood information.**

   Back of Card:

   - Title: **Neighborhood Addition and Update**
   - Acceptance Criteria:
     - The Admin should be able to add new neighborhoods by providing necessary details such as neighborhood name, location, amenities, etc.
     - The Admin should be able to update existing neighborhood details, such as location, amenities, and pricing, if required.
     - The system should validate that all required information is provided before adding or updating a neighborhood.
     - The neighborhood data should be stored securely in the system's database.
     - The Admin should be able to view a list of all existing neighborhoods and their details.
     - The system should provide appropriate error messages in case of any failures during the neighborhood addition or update process.
     - The neighborhood information should be reflected in the search results for users to view and access.
3. As a **user**, I want to be able to **compare neighborhoods based on specific criteria like location, amenities, price, and quality of construction,** so that I can **make an informed decision about my preferred neighborhood.**

   Back of Card:

   - Title: **Compare Neighborhoods**
   - Acceptance Criteria:

        - The user should be able to select multiple neighbourhoods for comparison.
        - The system should display detailed information about each neighbourhood, including location, amenities, price, and quality of construction.
        - The user should be able to sort the neighbourhoods based on the selected criteria.
        - The system should provide an easy-to-read comparison report of the selected neighbourhoods.

4. **As a user, I want to share information about my neighborhood with my friends and family so that they can stay updated and informed about local events, news, and activities.**

   Back of Card:

    - Title: **Neighborhood Information Sharing System**
    - Acceptance Criteria:
        - The Neighborhood Information Sharing System allows users to share information about their neighborhood with their friends and family.
        - The system includes features for creating and managing profiles, creating and joining neighborhood groups, and sharing information such as news, events, and activities.
        - The system also includes privacy settings to control the visibility of shared information and features for commenting and discussing shared information with other members of the neighborhood group.
        - The system sends notifications to users when new information is shared in their neighborhood group.

4.3 User Stories for Non-functional Requirements:

1. As a **user**, I want to be able to **easily navigate and understand the system's interface** so that I can **quickly find the information I need.**
    - Title: **User-Friendly Interface**
    - Acceptance Criteria:
        - The interface should be easy to navigate for all types of users, including those with limited technical experience.
        - The interface should be visually appealing, with clear and concise labeling of all features and functions.
        - The interface should be consistent across all pages and screens.
2. As a **user**, I want the system to be **compatible with a variety of devices** so that I can **access it from anywhere, regardless of the device I am using.**
    - Title: **Device Compatibility**
    - Acceptance Criteria:
        - The system should be able to work on all modern browsers and devices, including mobile phones and tablets.
        - The system should be optimized for different screen sizes and resolutions.

        - The system should have a responsive design that adjusts to the user's device and screen size.

3. As a mobile user, I want to be able to access and use the system seamlessly from my mobile device, so that I can accomplish tasks efficiently even while on-the-go.
    - Title: **Responsive design**
    - Acceptance Criteria:
        - Responsive design ensures that the system adapts to different screen sizes and resolutions, providing an optimal viewing and user experience on mobile devices.
        - This non-functional requirement emphasizes the importance of creating a mobile-friendly system that can be accessed from any device, without compromising usability or functionality. By implementing responsive design, the system can provide a consistent experience across different platforms, making it easier for users to navigate and interact with the system.
        - This requirement also helps to increase user engagement and satisfaction, as users can complete tasks quickly and easily from their mobile devices, without having to switch to a desktop or laptop.
4. As a **system administrator**, I want the system to be **easily maintainable** so that we can **quickly fix bugs and make updates as needed to ensure the system's functionality and user experience.**
    - Title: **Maintainability**
    - Acceptance Criteria:
        - The system should have clear and concise documentation for all features and functions.
        - The system should be modular and use standardized code and programming practices.
        - The system should have a testing and debugging framework in place for easy bug fixes and update.

4.4 User Stories for Pending Non-Functional Requirements:

1. As a **user**, I want the system to be **reliable and provide accurate information at all times**, so that **I can trust the information I receive from the system.**
    - Title: **Reliability**
    - Acceptance Criteria:
        - The system should have a uptime of at least 99%.
        - The system should provide accurate information about neighborhoods, including location, amenities, and pricing.
        - The system should be tested for errors and bugs regularly, with any issues addressed promptly.
2. As a **system administrator**, I want **the system to be scalable** so that we can **easily add new features and improve existing ones as needed.**
    - Title: **Scalability**
    - Acceptance Criteria:
        - The system should be able to handle an increase in the number of users and data without any major performance issues.

- The system should be designed in a modular way to allow for easy addition of new features or modules.
- The system should be flexible enough to integrate with other systems or technologies as needed.

3. As a **user**, I want the system to **load quickly and provide fast search results** so that I can **find the information I need in a timely manner.**
    - Title: **Fast Loading**
    - Acceptance Criteria:
        - The system should load quickly and provide search results in under a few seconds.
        - The system should be able to handle high traffic volumes without any significant slowdowns.
4. As a **user**, I want the **system to protect my personal information** and **keep my data secure** so that I can **freely provide my preferences without worrying about privacy**.
    - Title: **Security**
    - Acceptance Criteria:
        - The system should use industry-standard encryption and security protocols to protect user data.
        - The system should have a secure login process, including strong password requirements and multi-factor authentication options.
        - The system should have measures in place to prevent unauthorized access, including intrusion detection and prevention systems.