**Apartment Finder**

Software Requirements Specification

Version <1.0>

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| <27-09-18> | <1.0> | SRS 1.0 | Jarifa |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. [Introduction](#_Introduction) 5

1.1 [Purpose](#_Purpose) 5

1.2 [Scope](#_Scope) 5

1.3 Definitions, Acronyms, and Abbreviations 6

1.4 References 6

1.5 Overview 6

2. [Overall](#_Overall_Description) Description 6

3. [Specific Requirements](#_Specific_Requirements) 7

3.1 [Functionality](#_Functionality) 7

3.1.1 Sell Configured to Ordered Products. 7

3.1.2 Provide comprehensive product details. 7

3.1.3 Detailed product Categorizations 7

3.1.4 Provide Search facility. 7

3.1.5 Maintain customer profile. 8

3.1.6 Provide personalized profile 8

3.1.7 Provide Customer Support. 8

3.1.8 Email confirmation. 9

3.1.9 Detailed invoice for customer. 9

3.1.10 Provide shopping cart facility. 9

3.1.11 Provide multiple shipping methods. 9

3.1.12 Online tracking of shipments 9

3.1.13 Provide online Tax Calculations 10

3.1.14 Allow multiple payment methods. 10

3.1.15 Allow online change or cancellation of order. 10

3.1.16 Allow Online Product reviews and ratings 10

3.1.17 Offer financing options. 10

3.1.18 Provide detailed sitemap. 10

3.1.19 Offer online promotions and rewards. 11

3.1.20 Online Purchase of products. 11

3.2 Usability 11

3.2.1 Graphical User Interface 11

3.2.2 Accessibility 11

3.3 Reliability & Availability 11

3.3.1 Back-end Internal Computers 11

3.3.2 Internet Service Provider 11

3.4 Performance 12

3.5 Security 12

3.5.1 Data Transfer 12

3.5.2 Data Storage 12

3.6 Supportability 13

3.6.1 Configuration Management Tool 13

3.7 Design Constraints 13

3.7.1 Standard Development Tools 13

3.7.2 Web Based Product 13

3.8 On-line User Documentation and Help System Requirements 13

3.9 Purchased Components 13

3.10 Interfaces 14

3.10.1 User Interfaces 14

3.10.2 Hardware Interfaces 14

3.10.3 Software Interfaces 14

3.10.4 Communications Interfaces 15

3.11 Licensing Requirements 15

3.12 Legal, Copyright, and Other Notices 15

3.13 Applicable Standards 15

4. Supporting Information 15

Software Requirements Specification

# Introduction

The goal of this document is to gather and analyze and give an in-depth insight of the **Apartment Finder** app. This detailed overview is created to give the reader a clear view about the system and its operations.

## Purpose

The purpose of the document is to collect and analyze all assorted ideas that have come up to define the system, its requirements with respect to consumers. Also, we shall predict and sort out how we hope this product will be developed and used in order to gain a better understanding of the project, outline concepts that may be developed later, and document ideas that are being considered, but may be discarded as the product develops.

In short, the purpose of this SRS document is to provide a detailed overview of our software product, its parameters and goals. This document describes the project's target audience and its user interface, hardware and software requirements. It defines how our client, team and audience see the product and its functionality. Nonetheless, it helps any designer and developer to assist in software delivery lifecycle (SDLC) processes.

## Scope

Primarily, the scope pertains to specifying the requirements of the system to be developed. It focuses on an in depth ins and outs of the system, which will provide necessary instruction for the developer to develop the system appropriately.

The standard that was used to develop this document is straightforward as no tool was included to perform this job.

## Definitions, Acronyms, and Abbreviations

|  |  |
| --- | --- |
| Configuration | It means a product which is available / Selected from a catalogue can be customized. |
| JWT | JSON Web Token |
| JSON | Javascript Object Notation |
| REST | Representational State Transfer |
| API | Application Programming Interface |

## Overview

The remaining sections of this document provide an in depth description, including characteristics of the users of this project, the product's functional, technical and data requirements.  General description of the project is discussed in section 2 of this document.  Section 3 gives the functional requirements, data requirements and constraints and assumptions made while designing the E-Store.  It also gives the user viewpoint of product.  Section 3 also gives the specific requirements of the product.  Section 3 also discusses the external interface requirements and gives detailed description of functional requirements. Section 4 is for supporting information.

# Overall Description

This document contains the problem statement that the current system is facing which is hampering the growth opportunities of the company. It further contains a list of the stakeholders and users of the proposed solution. It also illustrates the needs and wants of the stakeholders that were identified in the brainstorming exercise as part of the requirements workshop. It further lists and briefly describes the major features and a brief description of each of the proposed system.

The following SRS contains the detail product perspective from different stakeholders. It provides the detail product functions of E-Store with user characteristics permitted constraints, assumptions and dependencies and requirements subsets.

# Specific Requirements

The specific requirements are –

## Functionality

Introduction –

This subsection contains the requirements for the Apartment Finder solution. These requirements are organized by the features discussed in the vision document. Features from vision documents are then refined into use case diagrams and to sequence diagram to best capture the functional requirements of the system.

### Search Panel

#### A map along with an input box and a drop down list and an advanced option button shall be displayed after loading the page.

#### User shall select a position near which s/he wants to live.

#### S/He will select a value from the drop down, which will be the radius of search circle.

#### A circle will be drawn on the displayed map, using user’s input.

### Advanced Options Panel

#### ‘Advanced search option’ is a panel of search filters that will hold numerous filters.

##### Nearby will be a filter to search for nearby establishments around the house to be let.

###### Nearby will contain mosque, market, school and hospital as options.

###### All of these options will contain a drop down list of expected distances.

###### All of these options will contain preference index.

###### Preference index will work in ascending order.

##### Flat size filter will use square feet as unit.

##### Number of Bed room / bath room combination will be a drop down list.

##### Monthly rent will use a range.

##### In which floor is the flat located will be an input field.

##### Existence of lift/escalator will be a drop down list of Yes/No.

##### Existence of parking space will be a drop down list of Yes/No.

##### Security measurements will refer to the existence of security guard, which will be a drop down list of Yes/No.

##### Month of availability will be a drop down list of future months.

#### At the end of this panel, there will be search button, clicking which will initiate the search request.

### Search result list

#### The system shall display a list of search results below the advanced option panel.

### Result display modal

#### Clicking on a resultant card will pop a modal.

#### Each modal will contain an image of the house, a description of the house and some properties of the house.

#### Below the description there will be information about the advertiser.

### Add new advertisement button

#### On the lower right corner, add new advertisement button will float, clicking which will prompt login/signup modal.

### Log-in prompt

#### A login prompt will contain usual login input panels.

### Sign-up prompt

#### A sign up prompt will contain usual sign up input panels, i.e. name, username, password, repeat password.

#### Further user information i.e. user bio etc. can be modified from user settings page.

### User’s past advertisements

#### Upon successful login user will be redirected to a page which will contain a list of past advertisements by this user, sorted by their date.

#### Expired advertisements will be distinguished by a flag.

#### User can view, edit, and delete these advertisements.

#### View will be similar to 3.1.4.

#### Click on the update button will open up an update page.

#### Submit button on update will prompt update process, which will be verified against the user in back end.

#### Delete button will prompt the deletion of the specific entry.

### User details page

#### User can view his detail information by clicking profile button on upper right corner.

#### This page will contain a button on the upper right corner which will redirect the user to user update page.

### User update page

#### User update page will contain all the fields of user details.

#### When a user changes any input it will be validated twice: once on the front and once in back.

### Add new advertisement

#### Add new advertisement button will be placed on the user’s past advertisement page.

#### This button will prompt a modal with the following inputs

##### Name of the house

##### Detailed address of the house

##### Flat size

##### Number of bedrooms and bathrooms

##### Rent

##### Which floor

##### Month of availability

##### Available security measurements

##### Existence of lift/escalator

##### Pin point location on the map

##### Couple of pictures of the house

##### Flat owner’s contact address and number

## Technical Aspects

### Tech Stack

#### Git will be used for VCS.

#### GitHub will be used to maintain the project.

#### Backend of this service will be a REST API, built on ExpressJS and NodeJS.

##### Mongoose will be used for interfacing with MongoDb.

##### JWT will be used for Authentication.

##### Morgan will be used for logging.

##### Swagger will be used for API Documentation.

##### SonarQube will be used to measure code coverage and ensure code quality.

##### Mocha will be used for testing purposes.

##### Password Hashing with Salt will be implemented to ensure security.

#### Frontend will be done in HTML, CSS and Javascript.

##### An nginx server will be used to host front end codes.

##### JWT will be used to communicate with the server.

##### Standard JSON objects will be the medium of payload communication.

#### Docker will host the entire application.

##### Two custom networks will be created using docker bridge network: one for the front end and another for the back.

##### Both network will open a single port for communication.

##### Frontend network will open port 80 which will be mapped to localhost’s 80 port.

##### Backend network will open port 3000 which will be mapped to localhost’s 3000 port.

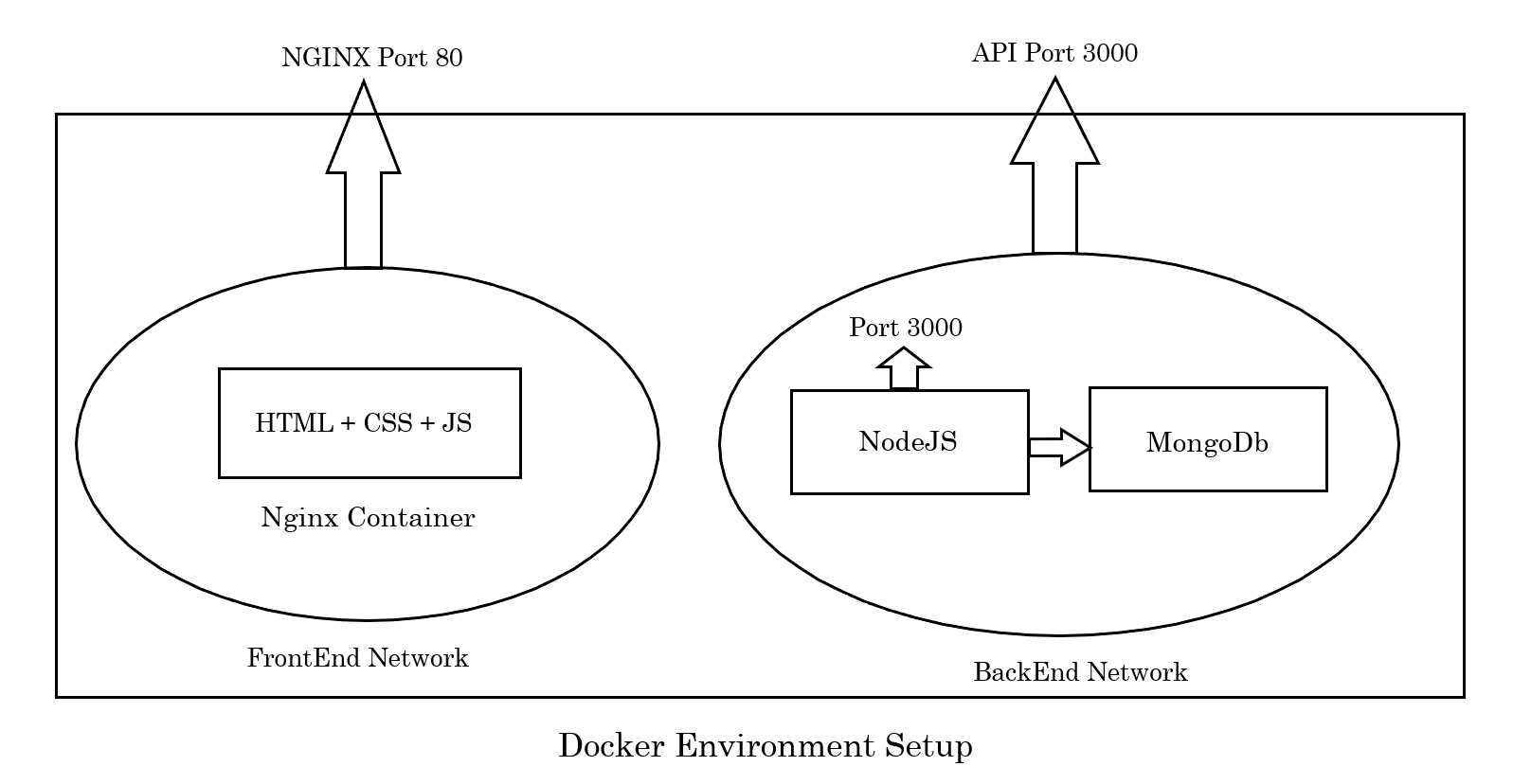
##### Frontend network will host a container of nginx, containing HTML, CSS and JS.

##### Backend network will host two containers: one with the NodeJS code and another with the mongodb.

##### MongoDb container won’t publish any port globally, it will only publish 27017 port for NodeJS to communicate.

##### MongoDb’s data will be backed up by volume mount.

##### Entire application and its networks will be maintained by Docker Compose files.



### Database

#### This application will be backed by a MongoDb Database.

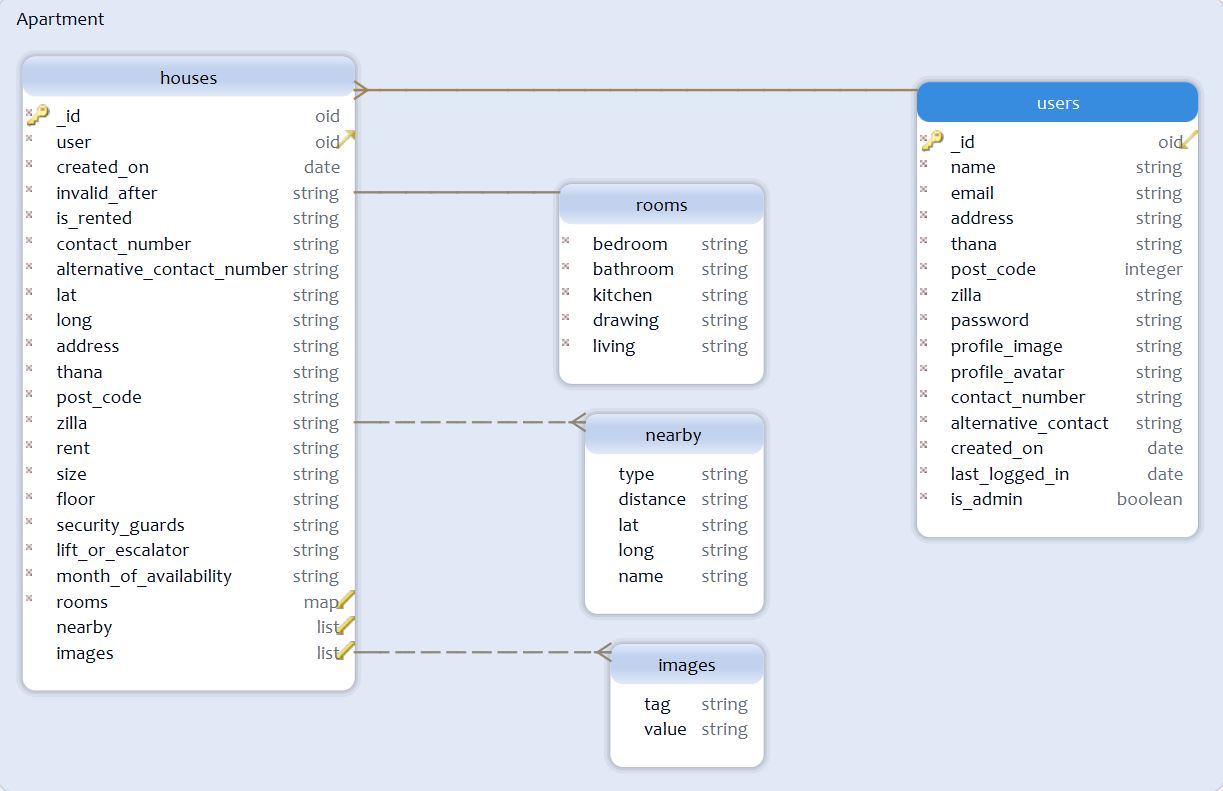
##### There will be two collections (or tables) in the database: users, houses.

##### Users will persist information regarding users.

##### Houses will record information regarding houses to be let.

##### A foreign key relationship from house to user will ensure data consistency.

##### All the information regarding nearby locations will be held in houses collection.



## Usability

### Graphical User Interface

The system shall provide a uniform look and feel between all the web pages.

The system shall provide use of icons and toolbars.

### Accessibility

The system shall provide zoom support for people with poor eye sight.

## Reliability & Availability

### Back-end Internal Computers

The system shall provide storage of all databases on redundant computers.

The system shall provide for replication of databases to off-site storage locations.

### Internet Service Provider

The system shall provide a contractual agreement with an internet service provider for T3 access with 99.9999% availability.

The system shall provide a contractual agreement with an internet service provider who can provide 99.999% availability through their network facilities onto the internet.

## Performance

The product shall be based on web and has to be run from a web server.

The product shall take initial load time depending on internet connection strength which also depends on the media from which the product is run.

The performance shall depend upon hardware components of the client/customer.

## Security

### Data Transfer

The system shall use secure sockets in all transactions that include any confidential customer information.

The system shall automatically log out all customers after a period of inactivity.

The system shall not leave any cookies on the customer’s computer containing the user’s password.

The system shall not leave any cookies on the customer’s computer containing any of the user’s confidential information.

### Data Storage

The customer’s web browser shall never display a customer’s password. It shall always be echoed with special characters representing typed characters.

The system’s back-end servers shall never display a customer’s password. The customer’s password may be reset but never shown.

The system’s back-end servers shall only be accessible to authenticated administrators.

Passwords in the system’s back-end databases shall be encrypted.

## Supportability

### Version Controlling System (VCS)

The source code developed for this system shall be maintained in VCS.

## Design Constraints

### Standard Development Tools

The system shall be built using Microsoft Visual Studio Code.

### Web Based Product

There are no memory requirements.

The computers must be equipped with web browsers such as Internet explorer.

The product must be stored in such a way that allows the client easy access to it.

Response time for loading the system should take no longer than five minutes.

A general knowledge of basic computer skills is required to use the product

## On-line User Documentation and Help System Requirements

As the product is Apartment Finder, online help system becomes a critical component of the system which shall provide –

It shall provide specific guidelines in the form of pdf to a user for using the system.

## Purchased Components

Not Applicable

## Interfaces

There are many types of interfaces as such supported by the E-Store software system namely; User Interface, Software Interface and Hardware Interface.

The protocol used shall be HTTPS.

The Port number used will be 80.

There shall be logical address of the system in IPv4 format.

### User Interfaces

The user interface for the software shall be compatible to any browser such as Internet Explorer, Mozilla or Netscape Navigator by which user can access to the system.

### Hardware Interfaces

Since the application must run over the internet, all the hardware shall require to connect internet will be hardware interface for the system. As for e.g. Modem, WAN – LAN, Ethernet Cross-Cable.

### Communications Interfaces

The e-store system shall use the HTTPS protocol for communication over the internet and for the intranet communication will be through TCP/IP protocol suite.

## Licensing Requirements

Not Applicable

## Legal, Copyright, and Other Notices

Apartment Finder should display the disclaimers, copyright, word mark, trademark and product warranties of the product.

## Applicable Standards

It shall conform to the industry standards.

# Supporting Information

Please refer the following document:

1. Vision document for E-store.
2. Use case analysis.
3. Structural models.
4. Behavioral models.
5. Non functional requirements model.
6. Traceability Matrix.
7. Project Plan