SOFTWARE REQUIREMENTS SPECIFICATION

for

Abashon (A House Rental Website in Bashundhara R/A)

Version 1.0

Prepared by :
1. K. M. Lutfor Rahman (1411162042) 2. Most. Airin Akter (1812770042)
3. Salman Sad Sakil (1711064642)
4. Sanjida Akter (1831522042)

Submitted to :
Dr. Md. Sazzad Hossain
Professor

March 28, 2023

Contents

1. Introduction

1.1. Purpose

Our city life is becoming increasingly crowded, and finding To-Lets roaming the streets of Bashundhara R/A can be daunting. Home to three universities and numerous office buildings, Bashundhara Residential Area, is one of the most densely populated areas in Dhaka. As a result, students living outside Dhaka find it more difficult to find rental housing near their universities. Seeing the same recommendation system for a rental home is very challenging, and typically the owner does not advertise their home online but promotes it to the locals. The student will generally go outside and inquire about the resident. For the duration of the semester, it will take a long time for students to secure housing. To solve this problem, we need a platform to help students find a house. For people in this category, the online house rental system is a relief. Here people can easily find information about rooms, flats, or houses just by sitting in front of their laptop or smartphone. The system we will develop has various features to make life easier for urban people. Here tenants can choose from a large variety of houses that meets their requirements. Also, landlords can rent or sell their homes or flats via this online system just by a click of a button. So it will be helpful not only for tenants but also for the house owners. Moreover, The application will be developed keeping all the essential features in mind for both the renters and owners.

1.2. Document Conventions

This document was created based on the IEEE System Requirement Specification Doc uments template.

1.3. Intended Audience and Reading Suggestions

The intended audience for this SRS is:

1. Users
2. Developers
3. Testers
4. Marketing Members
5. Project Manager

The document consists of all the information required by the team of software engineers who will be working on the project.

1.4. Project Scope

Our House Rental Management System aims to simplify the process of renting properties for its users. To make it accessible on all electronic devices the application will be a web based application and will be mobile responsive as much as possible. The application will provide all necessary functionalities for searching properties and adding properties and their images. The scopes for this project are identified to make the application development process easier. The scope will be explained from the user aspect of view and system function.

1. Admin
    - Manage and monitor the whole system
2. Owner
    - Register and log in to the system
    - Edit and update house description
    - Edit profile, delete profile
    - Receive details from tenant/student
3. Tenant/Student
    - Register and login to the system
    - Search for house
    - Booking the house and pay rent

2. Overall Description

2.1. Product Perspective

Currently, there are no similar services as ours in the market. Our software mainly focuses on students who study in nearby universities and want to reside in Bashundhara R/A. The system aims to create a user-friendly online house rental management system. The following is a list of requirements that shall be met to achieve the overall goal: To make it easier to track who wishes to come home and for an administrative management system. For the house finders, create an internet home renting system. Allow the administrator to see home finders. To create a completely functional and automated online system for house rental management. To offer a well-organized and dependable system with the fewest possible faults. Customers may easily register to rent a property anytime from their homes.

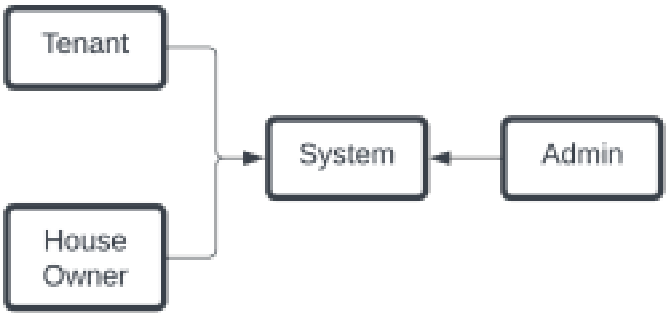

Figure 2.1.: User Types

2.2. User Classes and Characteristics

We have designed our software for three types of users. One is Tenants, they can look for houses and rent them. The second one is house Owners; they will register their houses on our site. And the final user class is the site administrator. He will have full access to the entire system. House owners and Tenants must have different login credentials.

2.3. Product Functions

Our product will allow users to sign up as tenants or house owners. House owners will be able to put their houses up for rent, and the tenants will be able to book the houses for them. To perform any significant action, the user must be logged in via email or google log-in. During the sign-up process, the user must verify their email. When renting, the tenant will pay through a payment gateway. The tenant will receive an email and a digital receipt upon successful payment. The tenant can search for houses considering rent cost, block number, or lift/generator facility. Tenants can check house location on google maps.

2.4. Operating Environment

Our software is a cross-platform web app It was built using HTML, JS, CSS, PHP, and

SQL. Any device that has a modern internet browser can access and use it. For a better experience, we recommend a PC with a full HD display and Chrome as the browser.

2.5. Design and Implementation Constraints

This product uses several APIs from different vendors, so their influence may affect our services. Also, when those 3rd party services get updated, we may have to update our codes manually to stay functional. Also, server up-time and host stability may affect the usability of our product significantly.

2.6. User Documentation

The software usage is very simple and intuitive. Users will be able to use it naturally.
But for any help, contact me via email.

2.7. Assumption and Dependencies

As a cross-platform software, it should run on all the major operating systems perfectly, as the primary data processing and decision-making happens on the server. But the front end might get affected by your browser version, brand, extensions, display size, etc.

3. External Interface Requirements

3.1. User Interfaces

The user interface will be simple and consistent, using terminology commonly understood by intended users of the system. The Abashon website will have multiple user-facing views.

UI-1:

The first set of views is for logging in to the website. These views consist of

- Sign up
- Log in
- Verify account
- Reset password

UI-2:
There will be a navigation bar at the top of the web page which will help users to navigate to different web pages.

UI-3:
Instructions will be provided to the users on top of the forms to be filled.

UI-4:
There will be alerts and pop-ups which appear in case the user makes a mistake while using the application.

UI-5:
The interface will be responsive for all screen sizes as much as possible to provide the users with a seamless experience

3.2. Hardware Interfaces

- No extra hardware interfaces are needed.
- The system will use the standard hardware and data communication resources. This includes but is not limited to, general network connection at the server/hosting site, network server, and network management tools

3.3. Software Interfaces

- Windows 7, 8, 10, Mac OS, Linux Operating system used to house all the applications and tools.
- Google Chrome Web browser used to display the data to end user.

3.4. Communications Interface

Communication from users and electronic forms on the site will be handled through the SQL database, and payments through credit card information will be shall be securely held by its back-end tools. Payment data will be encrypted as well. When signing up, users will be asked to fill out electronic forms requesting their information as well as the information of their property management.

4. System Features

4.1. Description and Priority

First, the user will sign up and log in to the account through google log-in or other email services. He will get a mail for verification. For safety, the password that the user or owner will give would be hashed, so others can’t see it. The user can reset the password. And the payment can see in PDF for printing out a copy of it. The houses will be marked in google maps. And they can pay through local online payment services.

4.2. Stimulus/Response Sequences

1. Google login
2. Email verification
3. Password hashing
4. Password reset
5. Payment info storing as PDF
6. SQL injection prevention
7. Google maps integration with house markers
8. 3rd party payment API

4.3. Functional Requirements

It entailed user involvement and statements of facts and assumptions that define the expectations of the system in terms of mission objectives, environment, constraints, measures of effectiveness, and suitability. The users:

- A system that improves on the efficiency of information storage and retrieval.
- A system that is easy to learn and use
- A system that is fast in processing transactions
- A system that is flexible, safe and convenient

This is a necessary task, action, or activity that was accomplished. The proposed system is able to:

• Allow the administrator to add houses, tenants, and defaulters details •

Allow the administrator to delete houses, tenants and defaulters' details •

Allow the administrator to search data in the database

• Allow the administrator to edit data in the database

5. Other Nonfunctional Requirements

5.1. Performance Requirements

The application should be scalable and should perform without any interruption for all the users. The system should have a high-performance rate when executing the user’s input and should be able to respond within a short period.

5.2. Safety Requirements

Login and visit the website should be safe for the users. That malware can't get into the device by browsing it.

5.3. Security Requirements

All personal data should be protected, and no one can access any further information without the administrator's permission. The system provides a username and password to prevent the system from unauthorized access. The user password must be greater than eight characters. Only users with valid passwords and usernames can login to view the users' page.

5.4. Software Quality Attributes

The system provides a help and support menu in all interfaces for users to interact with the system. The user can use the system by reading help and support. The system should always be available for access 24 hours, seven days a week. Also, in the occurrence of any significant system malfunctioning, the system should be available in 1 to 2 working days so that the business process is not severely affected. Considering the level of knowledge possessed by the users of this system, a simple but quality user interface should be developed to make it easy to understand and require less training.

5.5. Business Rules

The administrator of the application has full permission to control the system.

A. Analysis Models

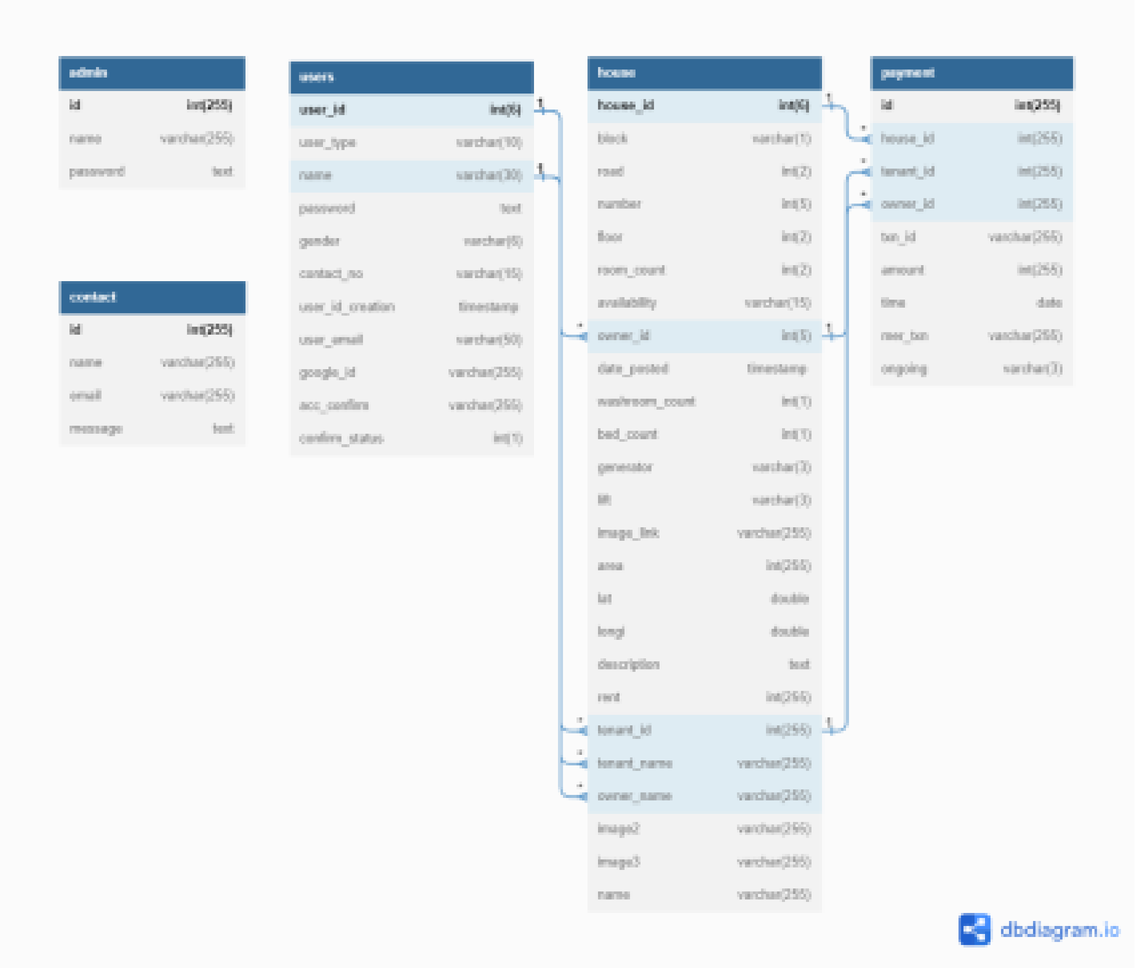

Figure A.1.: ER Diagram

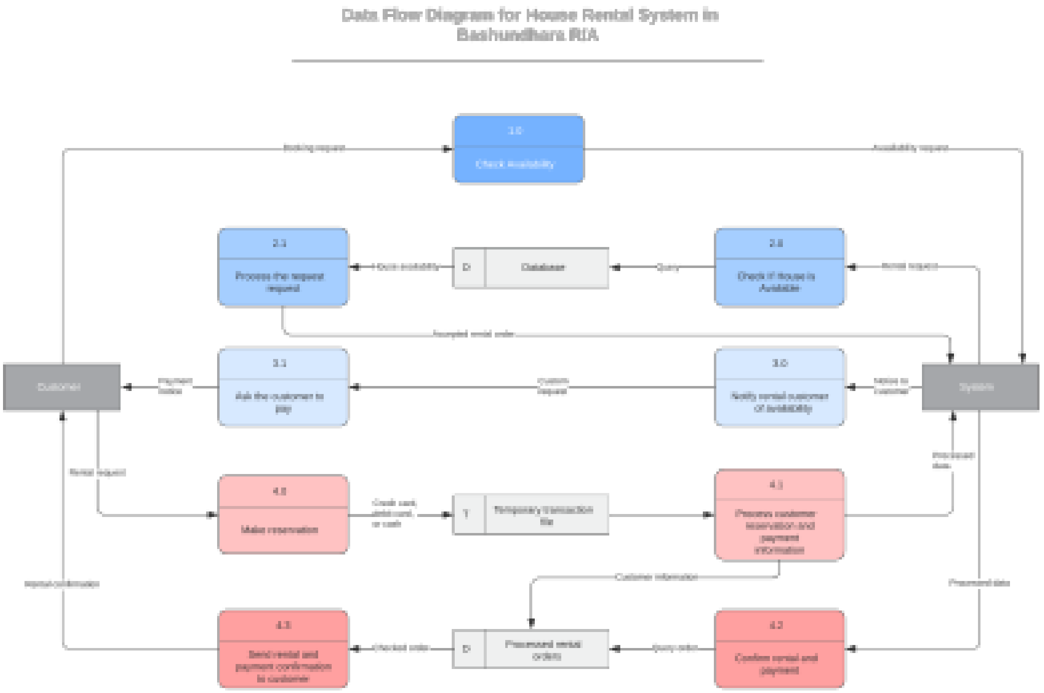

Figure A.2.: Data-Flow Diagram

Glossary

API Application Programming Interface An application programming interface (API) is a way for two or more computer programs to communicate with each other. It is a type of software interface offering a service to other pieces of software. A document or standard that describes how to build or use such a connection or interface is called an API specification.. 9

CSS Cascading Style Sheets Cascading style sheets work in conjunction with HTML and are directly responsible for dictating color, background, text color, font, po sitioning, and additional features to the web browser. CSS is a key component of web development.. 6

HTML HyperText Markup Language HTML is a form of programming code used to instruct a browser to create the structure for individual web pages on a website. HTML is responsible for things like text representation, defining visual format, providing search engines with page context, and designating page sections.. 6

JS JavaScript (JS) is the most popular lightweight, interpreted compiled programming language. It can be used for both Client-side as well as Server-side developments. JavaScript is also known as a scripting language for web pages.. 6

PHP PHP: Hypertext Preprocessor) An extremely popular scripting language that is used to create dynamic Web pages.. 6

SQL Structured Query Language (SQL) is a standardized programming language that is used to manage relational databases and perform various operations on the data in them.. 6