Software Requirements Specification

for

Dreamland

Version 1.0 approved

Prepared by Jordan Litsas

19.09.21

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

The product at hand is designed to streamline the connection and interactions between property owners and renters. This document will outline the demand for the product and how that is addressed by its features. The use cases and requirements are outlined. The SRS encompasses a single system, although that is intended to be grown and refined as this project unfolds.

Version 1.0

1.2 Intended Audience and Reading Suggestions

The document is intended for the project team and for Deakin University to provide advice. The current version contains a description of the core business and professional transactions that will be supported, proposed user documentation, and the core requirements for the system to provide critical business functions

.

1.3 Project Scope

The key purpose for this software is to streamline the connection between people seeking rentals and those renting their properties. The envisioned system scope includes:

- Users to facilitate the requirements listed below
- A renter profile which can be used to quickly apply for properties
- An application management interface
- A property management interface
- Automatic screening of rental applications
- Instant messaging between stakeholders
- Booking system for house visits such as viewings, inspections and maintenance
- Display, search and filter of property listings according to property criteria
- Connection to services concerning the provision of credit scores and police checks

2. Overall Description

2.1 Product Perspective

Dreamland is designed to be a novel self-contained product. In context, the system should operate to provide typical business functions provided by real estate agents, such as connecting stakeholders, property advertisement, application management, house inspection bookings, provision of legal documentation such as bond lodgement, disputes and lease agreements. The MO, therefore, is to automated manual and low complexity tasks performed by real estate agents. Although the product is new, it draws on features from established systems, such as domain.com.au, realestate.com.au, flatmates.com.au, Airbnb and facebook (marketplace). Core business features will be utilised to both meet and improve upon competition through holistically providing these critical business functions.

2.2 Product Features

- Account creation and management
- Renter profile creation and management
- Rental application made with renter profile
- Automatic renter profile screening for applicant suitability
- Instant messaging between stakeholders
- Property listing and management
- Wide browserable collection of properties
- Inspection and booking management

2.3 User Classes and Characteristics

- Individuals seeking rental properties for housing
- Individuals seeking to privately lease their properties, particularly those that do not wish to engage real estate agents.

3. System Features

3.1 User signup and login

3.1.1 Description and Priority

High priority

Users will be able to create an account thereby storing their signup information in the user collection. Following which, users will be able to return to their accounts with an email and password combination.

3.1.2 Stimulus/Response Sequences

Login authorisation is not considered in this section.

1. The user is prompted to provide a first name, last name, email address and postcode.
2. A new document is created from the data in step 1 and stored in the user collection.
3. The user is prompted to create a renter profile (as outlined in section 3.2).
4. If logging in:
    a. The user is prompted to enter their email and password.
    b. Upon a successful request, the user's id will be stored in their live instance.
5. The user will enter the site.

3.1.3 Functional Requirements

- **REQ 1.1**: View includes an input field for the user's first and last names, email address, postcode and two entries for password.
- **REQ 1.2:** A HTTP request is made to store that information, except password, within the user document collection.
- **REQ 1.3:** Login view includes an input field for email and password.
- **REQ 1.4:** A HTTP request is made to retrieve the user's id upon authorisation.

3.2 Renter Profile

3.2.1 Description and Priority

High priority

Users will be able to input key rental application information, a full list of which can be found in appendix 5.1. This profile can be used to assess the suitability of applicant's for a given property. A property owner may also create a renter profile to represent the minimum values a renter application must possess, and to autonomously filter unsuitable applicants.

3.2.2 Stimulus/Response Sequences

1. User enters requested information, and supporting documentation, into the UI.
2. That information is packaged into a HTTP request once structured.
3. That request body is inserted into a new renterApplication collection document and referenced with the user's id.

3.2.3 Functional Requirements

REQ 2.1: View includes input fields for relevant renter profile information (see appendix 5.1).

REQ 2.2: A HTTP request is made to store that information within the renterProfile document collection.

3.3 Property Listing

3.2.1 Description and Priority

High priority
Users are able to list properties on the site. This stores all relevant information pertaining to that property within the property document collection. The property is referenced with the listing user's id.

3.2.2 Stimulus/Response Sequences

1. User clicks the list property button.
2. The user is presented with input fields to enter the relevant data for their property (see appendix 5.1).
3. Once submitted, a HTTP request is made with the property information.
4. A new property document is inserted into the property collection.

3.2.3 Functional Requirements

REQ 3.1: View includes input fields for relevant property information (see appendix 5.1).

REQ 3.2: A HTTP request is made to store that information within the property document collection.

3.4 Submitting rental application for a property

3.2.1 Description and Priority

High priority
Users are able to view properties, click apply, and have their application referenced to that property. Doing so will store their userId in an array of ids referenced as a value to a propertyId key.

3.2.2 Stimulus/Response Sequences

1. User clicks the apply button linked to a property.
2. If it is the first application, a new document is created in the activeApplication collection structured as seen in appendix 5.1. If it is not the first, then their userId is added to the value field seen in appendix 5.1 under the activeApplication document.

3.2.3 Functional Requirements

REQ 4.1: View includes properties that can be applied for.

REQ 4.2: A HTTP request is made with the propertyId and userId to make such a reference in the activeApplication collection.

Appendix 5.1:

MongoDB Document Schema

USER

Category	Example Data
firstname	string
surname	string
email	string
postcode	int

ACTIVE_APPLICATIONS

Category	Example Data
propertyId	[<userId>]

RENTER_PROFILE

Category	Example Data
userId	ObjectId
employment	{employer, lengthOfEmployment, position, income}
personalReferences	[{name, contactNumber, email, relationship }]
professionalReferences	[{name: contactNumber, email, relationship} }]
pets	[{species, breed, size, age}]
children	[<age>]
rentalHistory	[{address, landlord/propertyManagerName,

	landlord/propertyManageEmail, landlord/propertyManagerContactNumber, lengthOfTenancy, reasonLeaving, evicted?, rentalAgreementBroken?}]
smoker	boolean
preferredMoveInDate	string
commitedOfCrime	boolean

PROPERTIES

Category	Example Data
propertyID	ObjectId
owner _id	string
applicantCriteria	{ propertyApplicantCriteria }
rent-type	Room/house
Marketvalue	int
demographics	string
Commuteprofile	string
bedrooms	int
pet	boolean
nbn	string
parking	int
Bathrooms	int
Housing_type	string
price	int
Energylevels	string

bidding	boolean
Availabledate	string
location	string
postcode	int
Outdoorfeat	string
Indoorfeat	string
keywords	string
heating&cooling	string

System Design