# Chitkara University

Himachal Pradesh, Baddi

# Software Requirement Specification

For

## Apartment Rental System

# Submitted To

## Ms. Tazeen Khan

Submitted By –

Name: Vatsal Uppal

ID: 1611981409

Year: 3rd

Section: 13L

Table of Contents

[Introduction 3](#_Toc528697306)

[Purpose 3](#_Toc528697307)

[Scope 3](#_Toc528697308)

[Definition, Acronyms or Abbreviations 3](#_Toc528697309)

[References 4](#_Toc528697310)

[Overview 4](#_Toc528697311)

[General Description 5](#_Toc528697312)

[Product perspective 5](#_Toc528697313)

[Product Functions 5](#_Toc528697314)

[User Characteristics 5](#_Toc528697315)

[General Constraints 5](#_Toc528697316)

[Assumptions 6](#_Toc528697317)

[Functional Requirements 7](#_Toc528697318)

[Non-Functional Requirements 9](#_Toc528697319)

[System Architecture 10](#_Toc528697320)

[System Model 11](#_Toc528697321)

[Appendix 12](#_Toc528697322)

[Appendix A. Java Beans and DAO’s 12](#_Toc528697323)

[*Appendix B.* *No-SQL Databases* 12](#_Toc528697324)

# Introduction

## Purpose

The purpose of this project is to build an apartment rental web application. The web application must be able to assist people looking for an apartment to rent to be connected with people who want their property to be rented.

The document is intended for the customer and the developer (designers, testers, maintainers).

The reader is assumed to have basic knowledge of Online Marketing.

## Scope

The online rental web application would be able to provide a safe and convenient online environment to connect potential customers with people putting up their property for rent.

The application can reek monetary benefits by –

1. Attracting significant traffic of people wanting to use the application.
2. Providing users with premium services like endorsed listings.

The application could also be made into a mobile application to target a wider demographic.

## Definition, Acronyms or Abbreviations

|  |  |
| --- | --- |
| SRS | Software Requirement Specification |
| App | Application |
| PC | Personal Computer |
| OS | Operating System |
| IEEE | The Institute of Electrical and Electronics |
| DAO | Data Access Object |
| CRUD | Create, Read, Update and Delete |

## References

|  |  |
| --- | --- |
| Briggs 2005 | Briggs, J. (2005). SUMS documentation. Retrieved 3rd  December 2005, from  http://www.tech.port.ac.uk/staffweb/briggsj/jimapp/SUMS/ |
| IEEE 1998 | IEEE Std 830-1998, IEEE Recommended Practice for  Software Requirements Specifications. ISBN 0-7381-0332-2. |

## Overview

This document has been prepared in accordance with the IEEE Std 830‐1998, IEEE Recommended Practice for Software Requirements Specifications [IEEE 830‐1998 (1998)]. It provides the information of Product perspective, Product functions, User characteristics, Constraints, Assumptions and Dependencies and specific requirement

# General Description

## Product perspective

The online rental web application is an online platform that provides the user to let out or rent an apartment without the need for a property dealers or brokers. The user is identified by the information he provides during sign-up, that contains a unique username and some personal information, such as email, locality and mobile number.

The online rental web application is a generic web app that can be classified as one of many web apps that target the real estate market.

## Product Functions

The online rental system web application will be able to do the following –

* Register new users and authenticate old ones.
* Provide a view for listings in authenticated user’s locality unless locality is explicitly specified.
* Provide the ability for users to create new listings.
* Provide the ability to display available listings with corresponding listing’s information.
* Provide contact details for listing’s owner upon request.

## User Characteristics

The online rental web application targets anyone who want to put a property for rent or is looking for one. Anyone who logs in into the web app can look for properties up for rent or even upload his own listing. The customer is not required to have any technical expertise but the ability to browse the website. Anyone maintaining the website must be proficient in Java.

## General Constraints

The application would be accessible to all the customers around the clock. The user interface would be simple enough so that no training is required to use the application. User’s contact details and account information is persistently stored with passwords encrypted using a hashing algorithm.

## Assumptions

It is assumed that the customer will browse the web application using well known web browsers like Chrome, Opera and Microsoft Edge. Older versions of said browsers and those the web app is not tested on are not guaranteed to work.

# Functional Requirements

* User Login and Registration –

User login and Registration requires implementation of two basic classes –

* The User Bean Class –
  + The user bean class encapsulates an entity “user”.
  + It consists of fields like assigned user id, and other attributes submitted by the user via the online web form namely, username, email, mobile and password.
  + A user bean object would serve as an intermediator for communication between the distinct backend and the front end.
* The User DAO class –
  + The user DAO class abstracts the database CRUD operations.
  + Every time a “user bean” must be stored in the database, a corresponding instance of the “user DAO” class must be created and the required method invoked for performing some database operation.
  + The user DAO class would also be responsible for encrypting passwords using one-way hash like sha-256.
* Listing Management –

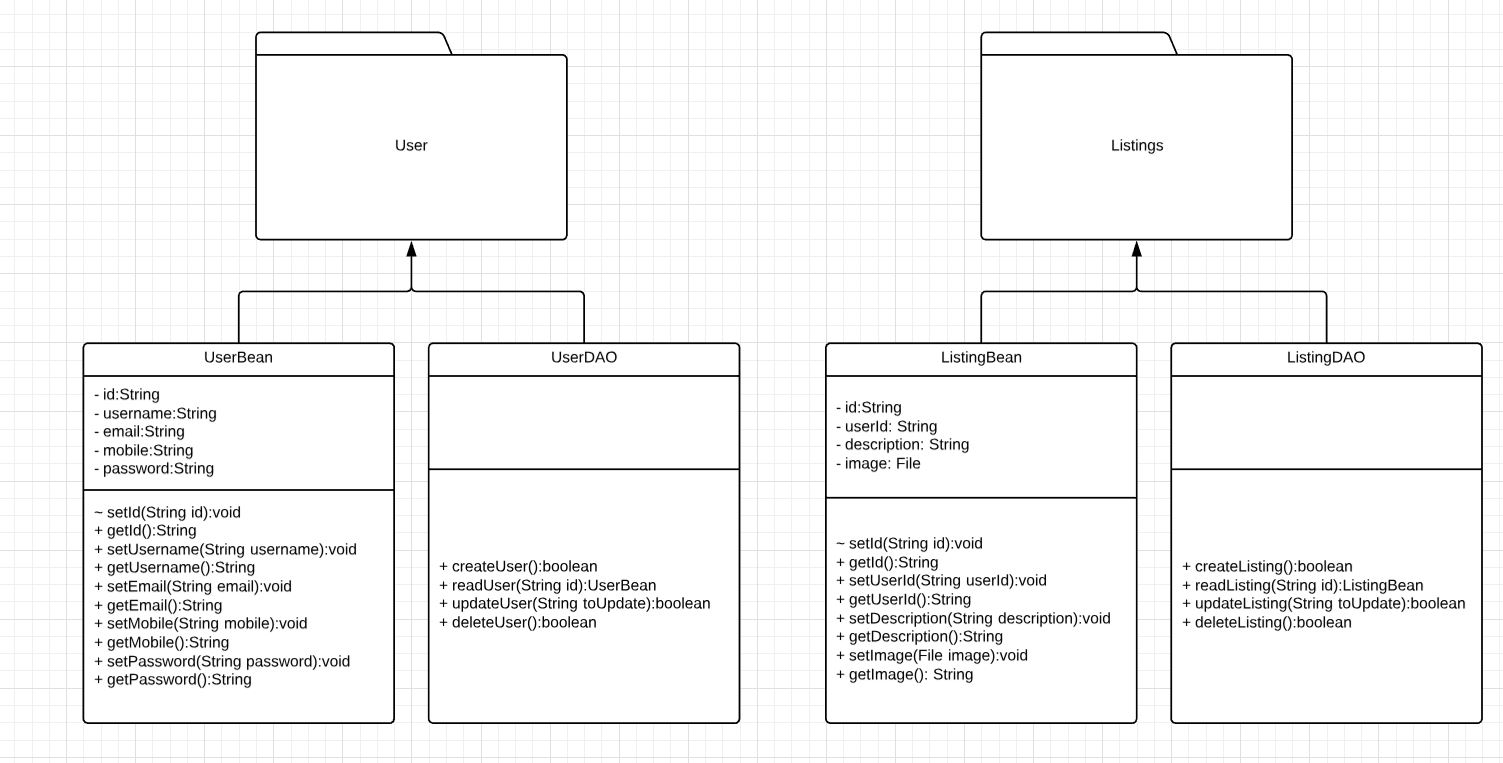
Listing management requires implementation of two basic classes –

* The Listing Bean Class –
  + The listing bean class encapsulated an entity “listing”.
  + It consists of fields representing properties of a listing namely, listing id, corresponding user id, listing description and optionally an image for the said listing.
  + A user can have multiple listings hence many listing bean objects can have the same user id.
* The Listing DAO class –
  + The listing DAO class abstracts the database CRUD operations.
  + Every time a “listing bean” must be stored in the database, a corresponding instance of the “listing DAO” class must be created and the required method invoked for performing some database operation.
  + The “listing DAO” class would also be able to infer a listings owner (via the user id stored in a listing bean) and retrieve the corresponding “user bean” object from the database.

# Non-Functional Requirements

* Performance Requirements –
  + The web platform must be accessible from all devices (PC’s, Android, IOS, etc)
  + The web platform must be able to provide the user with smooth user experience.
* Security Requirements –
  + The website must enforce proper form validations.
  + The password set by the user must be at least 8 digits long.
  + Un-Authorised users should not be able to access any of the databases.
  + Authorised users should only be able to view information relevant to their scope.
  + Users are advised to set a strong password on signup.

# System Architecture



# System Model

The web application to be developed adopts the “Waterfall Model”.

Waterfall model is deemed suitable for the current project due to the following reasons –

* The project has static requirements.
* The Waterfall is easy to manage due to its rigid structure, it is defined what must be done in each stage.
* The Waterfall model works well for small projects where the requirements are very well understood.

# Appendix

## Appendix A. Java Beans and DAO’s

Java Bean –

According to Java white paper, it is a reusable software component. A bean encapsulates many objects into one object, so we can access this object from multiple places. Moreover, it provides the easy maintenance.

DAO –

The Data Access Object (DAO) pattern is a structural pattern that allows us to isolate the application/business layer from the persistence layer (usually a relational database, but it could be any other persistence mechanism) using an abstract API.

The functionality of this API is to hide from the application all the complexities involved in performing CRUD operations in the underlying storage mechanism. This permit both layers to evolve separately without knowing anything about each other.

## Appendix B. No-SQL Databases

No-SQL Databases –

NoSQL is an approach to database design that can accommodate a wide variety of data models, including key-value, document, columnar and graph formats. NoSQL, which stand for "not only SQL," is an alternative to traditional relational databases in which data is placed in tables and data schema is carefully designed before the database is built. NoSQL databases are especially useful for working with large sets of distributed data.