Software Requirements Specification

for

<FlatFindr>

Version 1.0 approved

Prepared by <Patrick Indermühle>

<Team 6>

<09.10.2016>

Table of Contents

1. Introduction 1

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Intended Audience and Reading Suggestions 1

1.4 Product Scope 1

1.5 References 1

2. Overall Description 2

2.1 Product Perspective 2

2.2 Product Functions 2

2.3 User Classes and Characteristics 2

2.4 Operating Environment 2

2.5 Design and Implementation Constraints 2

2.6 User Documentation 2

2.7 Assumptions and Dependencies 2

3. External Interface Requirements 3

3.1 User Interfaces 3

3.2 Hardware Interfaces 3

3.3 Software Interfaces 3

3.4 Communications Interfaces 4

4. Current System Features 5

4.1 Creating advertisements 5

4.1.1 Description and Priority 5

4.1.2 Stimulus/Response Sequences 5

4.1.3 Functional Requirements 5

4.1.4 Non-Functional Requirements 5

4.2 Seeing advertisements 5

4.2.1 Description and Priority 5

4.2.2 Stimulus/Response Sequences 5

4.2.3 Functional Requirements 5

4.2.4 Non-Functional Requirements 6

4.3 Searching advertisements 6

4.3.1 Description and Priority 6

4.3.2 Stimulus/Response Sequences 6

4.3.3 Functional Requirements 6

4.3.4 Non-Functional Requirements 6

4.4 Inspect advertisements 6

4.4.1 Description and Priority 6

4.4.2 Stimulus/Response Sequences 6

4.4.3 Functional Requirements 6

4.4.4 Non-Functional Requirements 7

4.5 Send enquiry 7

4.5.1 Description and Priority 7

4.5.2 Stimulus/Response Sequences 7

4.5.3 Functional Requirements 7

4.5.4 Non-Functional Requirements 7

4.6 Manage enquiries 7

4.6.1 Description and Priority 7

4.6.2 Stimulus/Response Sequences 7

4.6.3 Functional Requirements 8

4.6.4 Non-Functional Requirements 8

4.7 Bookmarking adds 8

4.7.1 Description and Priority 8

4.7.2 Stimulus/Response Sequences 8

4.7.3 Functional Requirements 8

4.7.4 Non-Functional Requirements 8

4.8 Subscribe to adds 8

4.8.1 Description and Priority 8

4.8.2 Stimulus/Response Sequences 8

4.8.3 Functional Requirements 9

4.8.4 Non-Functional Requirements 9

5. Future System Features 10

5.1 Create advertisements for property sales 10

5.2 Search for advertisements for property sales 10

5.2.1 Description and Priority 10

5.2.2 Stimulus/Response Sequences 10

5.2.3 Functional Requirements 10

5.2.4 Non-Functional Requirements 10

5.3 create alerts for advertisements for property sales 11

5.3.1 Description and Priority 11

5.3.2 Stimulus/Response Sequences 11

5.3.3 Functional Requirements 11

5.3.4 Non-Functional Requirements 11

5.4 Extended search options 11

5.4.1 Description and Priority 11

5.4.2 Stimulus/Response Sequences 11

5.4.3 Functional Requirements 11

5.4.4 Non-Functional Requirements 11

5.5 Premium and Normal users 12

5.5.1 Description and Priority 12

5.5.2 Stimulus/Response Sequences 12

5.5.3 Functional Requirements 12

5.5.4 Non-Functional Requirements 12

6. Other Nonfunctional Requirements 13

6.1 Performance Requirements 13

6.2 Safety Requirements 13

6.3 Security Requirements 13

6.4 Software Quality Attributes 13

6.5 Business Rules 13

7. Other Requirements 14

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Maëlle Boughattas | 11/10/2016 | Additional information + correction + layout | 0.3 |
|  |  |  |  |

# Introduction

## Purpose

This is the Software Requirements Specification (SRS) document for FlatFindr revision 0 version number 1.0. This document covers only the user interaction with the system and the features of the system but not the infrastructure or server side communication or implementations of the system.

## Document Conventions

## Intended Audience and Reading Suggestions

This document is intended for the developers and document writers (Patrick Indermühle, Bernhard Zahnd, Maëlle Boughattas, Michael Baur and Valerie Haftka), for the customer (FlatFoundrs), the project supervisors (Haidar Osman, Claudio Corrodi, Manuel Leuenberger) and developers of other projects that are interested in the requirements of FlatFindr.

## Product Scope

The product in question is FlatFindr, an online website for finding rooms and studios to rent. You can also advertise your own free rooms you want to rent to others. FlatFindr’s major benefit is its user-interface.

## References

Initial project description from the customer: https://github.com/scg-unibe-ch/ese2016/wiki/Project-Description  
  
Documentation of the system as it is : https://github.com/scg-unibe-ch/ese2016-team6/tree/master/documentation

# Overall Description

## Product Perspective

The product is a standalone software that was created by FlatFounders that requires maintenance in order to expand its features. This is an absolute condition for FlatFindr to remain competitive on the market.

## Product Functions

The product allows users to create advertisements for their own rooms, inspect rooms whose advertisements have been created by other owners, schedule meetings with other room owners for room visits and contact other people through the website.

## User Classes and Characteristics

There are two types of users:

* Users that are not logged in. They can see rooms that are advertised on FlatFindr, but cannot contact logged in users or schedule room visits;
* Users that are logged into their FlatFindr account. They can schedule meetings with other room owners and create advertisements for their own rooms.

## Operating Environment

The software is written in Java (IDK 8.0 or +) and runs on a server. It creates a webpage that users can visit with their browsers. Operation system and platform can be chosen freely within reason.

## Design and Implementation Constraints

It is important that the software functions with all major browsers (Chrome, Internet Explorer, Opera, Firefox, Safari) and that response times do not keep users waiting.

## User Documentation

The Documentation of the system can be acquired at https://github.com/scg-unibe-ch/ese2016-team6/tree/master/documentation

## Assumptions and Dependencies

In this document we assume that the software is mostly written in Java and in HTML, Javascript and CSS to design the website.

# External Interface Requirements

## User Interfaces

On the **homepage**, the user can see a list of the newest advertisements. Clicking on the advertisements moves the user to the advertisement screen

On the **advertisement screen**, the user can see all relevant data about the room in question. If the user is logged in, he can send enquiries for visits to the owner of the room, or contact him by message. If the user is not logged in, the buttons to contact the owner or sending enquiries are replaced by buttons that send the user to the login screen.

On the **search screen**, the user can select the search requirements. The user can search rooms and studios separately, by city or a zip code, define a radius (km) of search from the mentioned place, and set an upper price limit for rooms.

On the **login screen**, the user can type in his email address and password, and click on the login button to login. There is also a button to sign up, which carries the user to the sign up screen.

On the **sign up screen**, the user can enter his first name, last name, password, email address and gender and click on the sign up button in order to create an account.

On the **header tile**, which is visible from all screens, there is a button with the name of the website (FlatFindr), which moves the user to the homepage. There is also a search button which moves the user to the search screen. Two cases are possible:

* If the user is not logged in, there is a login button on the top right which moves the user to the login screen;
* If the user is logged in, there is instead the username and the user’s profile picture. A drop done list appears when the user hovers the mouse over the profile picture. This drop down list contains buttons that can take the user to the **add-placement screen**, the **my-rooms screen**, the **message screen**, the **enquiries screen,** the **schedule screen**, the **alerts screen**, the **profile scree**n and the **logout button**.

## Hardware Interfaces

The software interacts with the hardware through the Java library. The hardware must be compatible with Java.

## Software Interfaces

The software has a dependency on Java for its main server side logic and uses Javascript, CSS and HTML for the website interface. Software compatibility with Spring, the database language, has to be assured.

## Communications Interfaces

The software must be able to recognize email addresses and to create website that can be opened by a browser. Users must be able to send their passwords and other personal data in encrypted format. The software must be able to access the internet to send its data to users.

# Current System Features

## Creating advertisements

### Description and Priority

Ability of users to create advertisements for the room they want to rent has a priority of 9.

### Stimulus/Response Sequences

Logged in users must be able to enter the create-advertisement screen, where the system displays a screen that allows the user to enter the relevant data for the room. When the user is done, a confirmation button can be pressed which creates an entry in the data base for this room.

### Functional Requirements

* *REQ-LoginCheck*: the software must be capable of distinguishing logged in users and users that are not logged in;
* *REQ-CAS-Entry*: there must be a way for users to enter the create-advertisement-screen;
* *REQ-CAS-Input*: the create-advertisement screen must have textboxes, choice boxes or other input interfaces that can collect data for all the relevant data that the advertisement requires.

### Non-Functional Requirements

*REQ-DataBase-Consistency*: The user must not be able to create an invalid entry. In this context this means that the user must not be able to create an advertisement that does not have the minimum necessary data.

## Seeing advertisements

### Description and Priority

Ability of users to see advertisements created by theirselves or other users has a priority of 9.

### Stimulus/Response Sequences

When the user enter the homepage, he should see a list of advertisements for rooms.

### Functional Requirements

* *REQ-FetchAdd*: the software must be able to send the information of the advertisements entries to the user.
* *REQ-FetchAdd-Display*: The software must be able to display to the user the data of the advertisements sent to user.

### Non-Functional Requirements

*REQ-FetchAdd-Latency*: the software must be able to send the data within 10 seconds of the request in order to not keep the user waiting.

## Searching advertisements

### Description and Priority

Ability of users to search for advertisements created by theirselves or other users has a priority of 9.

### Stimulus/Response Sequences

When the user selects the search option, a search option screen should open so that the user can specify what to search for. Once the user confirms the search, he should see the results of his search.

### Functional Requirements

* *REQ-searchInput*: the software must be able to display an input screen for the search criteria (such as area, price limit, etc).
* *REQ-search filter-Display*: the software must be able to get the data the user has given to filter all hits on the data base.
* *REQ-searchdisplay filter-Display*: the software must be able to display the results of a search.

### Non-Functional Requirements

*REQ-FetchAdd-Latency*: The software must be able to send the data within 10 seconds of the request in order to not keep the user waiting.

## Inspect advertisements

### Description and Priority

Ability of users to get more detailed information about a room has a priority of 7.

### Stimulus/Response Sequences

When the user selects an ad, a screen should be displayed that shows more detailed information about the room (such as pictures of the room, location, roommates etc).

### Functional Requirements

*REQ-detailGet*: the software must be able to get and display all the data available about a room.

### Non-Functional Requirements

* *REQ-FetchAdd-Latency*: the software must be able to send the data within 10 seconds of the request in order to not keep the user waiting.
* *REQ-OptionalData*: The data base must allow for optional room data to be entered and saved. (Not all rooms have roommates for example and there is no clear upper limit to roommates)

## Send enquiry

### Description and Priority

The ability of users to send an enquiry to the owner of a room.  
Priority : 7

### Stimulus/Response Sequences

When the user selects an add, a screen should be displayed that shows more detailed information about the room (like pictures of the room, location, roommates etc). On this room there must be a button to send an enquiry to the owner of that room

### Functional Requirements

REQ-saveEnquiry: The software must be able to save the enquiry the user has sent.

REQ-displayEnquiry: The software must be able to display to the receiver that an enquiry has been sent to him/her.

### Non-Functional Requirements

## Manage enquiries

### Description and Priority

Ability of users to have the enquiries displayed to them in an organized fashion has a priority of 5.

### Stimulus/Response Sequences

When the user enters the enquiry screen, a list of pending enquiries should be displayed to the user.

### Functional Requirements

*REQ-formatEnquiry*: the software must be able to get, format and display all the enquiries a user has pending.

### Non-Functional Requirements

*REQ-FetchAdd-Latency*: the software must be able to send the data within 10 seconds of the request in order to not keep the user waiting.

## Bookmarking adds

### Description and Priority

Ability of users to add a room to the bookmark list has a priority of 7.

### Stimulus/Response Sequences

When the user selects an ad, a screen should be displayed that shows more detailed information about the room (such as pictures of the room, location, roommates etc). On this screen there must be a way for a room to be book marked.

### Functional Requirements

* *REQ-BookMarkSave*: The software must be able to save all book marks of an individual user.
* *REQ-BookMarkDisplay*: The software must be able to get and display all bookmarks a user has.

### Non-Functional Requirements

*REQ-FetchAdd-Latency*: The software must be able to send the data within 10 seconds of the request in order to not keep the user waiting.

## Subscribe to adds

### Description and Priority

Ability of users to add a room to the subscription list when a room on the subscription list changes in a meaningful way the user should be informed has a priority of 7.

### Stimulus/Response Sequences

When the user selects an ad, a screen should be displayed that shows more detailed information about the room (such as pictures of the room, location, roommates etc). On this screen there must be a way for a room to be added to the subscription list.

### Functional Requirements

* *REQ-subscriptionSave*: the software must be able to save all subscriptions of an individual user.
* *REQ-subscriptionDisplay*: the software must be able to get and display all subscriptions a user has.
* *REQ-subscriptionAlert*: the software must be able to send a message to the user if a subscribed room changes in a meaningful way.

### Non-Functional Requirements

* *REQ-FetchAdd-Latency*: The software must be able to send the data within 10 seconds of the request in order to not keep the user waiting.
* *REQ-AlertOptimization*: The software must be able to handle possibly up to hundreds of alerts being sent every time a room changes. This must not add to the response time to user requests.

# Future System Features

## Create advertisements for property sales

### Description and Priority

Users must be able to create advertisements for property they want to sell.  
Priority : 7

### Stimulus/Response Sequences

When the user enters the Homepage there should be a button to create an advertisement for properties for sale that is only available to logged in users. When the user presses that button, a window opens where the user can enter the data corresponding to that advertisement. When the user presses confirm the advertisement is created or an error message is displayed depending on the validness of the entered data.

### Functional Requirements

REQ-SaleAddInput: The software must be able to create a window in which the user can enter the data for the advertisement.

REQ-saveSaleAdd: The software must be able to save all subscriptions of an individual user.

### Non-Functional Requirements

REQ-SaleDataConsistency: The software must not allow invalid data to be entered into the database.

## Search for advertisements for property sales

### Description and Priority

Users must be able to search for advertisements for property others want to sell.  
Priority : 7

### Stimulus/Response Sequences

On the search window there should be an option to search for rentable rooms or rooms for sale. When the users selects the sale option the system must search for rooms for sale and display the results.

### Functional Requirements

REQ-saleDisplay: The software must be able to get and display all the sale advertisements that fit the search criteria.

### Non-Functional Requirements

REQ-FetchAdd-Latency: The software must be able to send the data within 10 seconds of the request in order to not keep the user waiting.

## Create alerts for advertisements for property sales

### Description and Priority

Users must be able to create alerts that inform the user when an advertisement for sales is created that fit a certain criteria.  
Priority : 7

### Stimulus/Response Sequences

When the user enters the Homepage there should be a button that allows a logged in user to go to a screen where they can specify what the alert must look for and then confirm the criteria. Once an alert has been created it will send messages to the user every time an advertisement is created that fits the criteria

### Functional Requirements

REQ-alertSave: The software must be able to save all alerts of an individual user.

REQ-alertCheck: The software must be able to check if an alert has been triggered every time a room is created.

REQ-alertMessage: The software must be able to send a message to the user if an alert has been triggered.

### Non-Functional Requirements

REQ-AlertOptimization: The software must be able to handle possibly up to hundreds of alerts being sent every time a room is created. This must not add to the response time to user requests.

## Extended search options

### Description and Priority

Users must be able to specify more criteria for the search. Priority : 7

### Stimulus/Response Sequences

When the user enters the search screen there should be more options for filtering the results of a search.

### Functional Requirements

*REQ-moreSearchFilters*: There must be more search filter options in the search-options-screen

### Non-Functional Requirements

## Premium and Normal users

### Description and Priority

Users must be split into premium and normal users. Premium users get alerts sooner than normal users and the advertisements of premium users are higher up in the list of ads. Priority : 8

### Stimulus/Response Sequences

When the user enters the Homepage there should be a button that allows users to pay money to become a premium user. A window should be opened that allows the user to specify how much money they want to spend and how they want to pay (with pay pal, credit card, bill to home address, etc.). Once the user has confirmed the action the price will be charged from the user and the account will be upgraded to premium level. Premium accounts may be limited in time and expire after that time has passed.

### Functional Requirements

* *REQ-displayPremiumOptions*: The software must be able to display all the options for differently priced premium accounts.
* *REQ-chargeMoney*: The software must be able to charge the users money for the premium account through whatever medium the user selects.
* *REQ-countDownPremium*: The software must keep track of the amount of time left on the premium account of a user and revoke premium privileges once that time has passed.

### Non-Functional Requirements

*REQ-LegalitiesOfOnlinePayment*: The software must be able to consider the laws of the country the user resides in and properly charge taxes or even deny the user the right to order a premium account if this action breaks the law.

# Other Nonfunctional Requirements

## Performance Requirements

All client to server requests must be completed within 10 seconds to make sure users are not kept waiting too long when moving around the website.

## Safety Requirements

It is important that regular backups of private user data is made so that in case of a system failure the data can be restored. On top of that, the system must be designed in a way so that no invalid data entries can be made and inserted into the data base.

## Security Requirements

It is important that all private data is secured against hacking and theft attempts at all times.

## Software Quality Attributes

The software provides a user interface in order to create advertisements (create data entries) and see advertisements of others (search the data base for certain data entries). It is important that the user interface part of the software remains uncoupled from the data base handling part of the software so that either can be changed without the other. On top of that, since the product will receive maintenance for quite some time from now, it is important that readability of the code remains a strong point of the software.

## Business Rules

Only logged in users can create advertisements, schedule meetings with room owners or contact owners of rooms.

# Other Requirements

For legal reasons it is important that confidentiality of the private user data is guaranteed. The data base where this data is saved must be secured.

Appendix B: Analysis Models

TBD

Source: http://www.frontiernet.net/~kwiegers/process\_assets/srs\_template.doc