Software Requirement Specification

ESE Team 9

Sven Kellenberger
Rafael Ottersberg
Levi Ryffel
Marcel Schmutz
Kevin Studer

06.10.2016

Contents

1 Introduction 3

1.1 Purpose . . . 3

1.2 Scope of the Project . . . 3

1.3 Glossary . . . 3

1.4 Stakeholders . . . 3

1.5 System Overview . . . 4

1.6 References . . . 4

2 Overall Description 4

2.1 Product Perspective . . . 4

2.2 Product Functions . . . 4

2.3 User Classes and Characteristics . . . 4

2.3.1 User . . . 4

2.3.2 Advertiser . . . 4

2.3.3 Roommate . . . 4

2.3.4 Premium User . . . 4

2.4 Operating Environment . . . 5

2.5 Design and Implementation Constraints . . . 5

3 External Interfaces 5

3.1 User Interfaces . . . 5

3.2 Hardware Interfaces . . . 5

4 System Features 7

4.1 Use-Cases . . . 7

5 Other Requirements 10

1 Introduction

1.1 Purpose

This document presents a detailed description of the web application Flatfindr. Foremost, it provides a legaly binding contract between the stakeholders and the developer team 9. This document is a RUP conform SRS document.

1.2 Scope of the Project

The web application Flatfindr will help the participating user to promote free places in their rooms or flats. Also it should help user to search for free places in rooms or flats.

More specifically, the application will provide methods to search for free flats, help to ensure flats with specific parameters are presented for users and that the application will provide enough methods, to get in contact with the advertiser. On the other side, users will be able to insert an ad for a free place in their flats, will help define the visitation time (if an user wants to see the flat which is advertised) and provide some other instruments to describe the flat as best as possible.

1.3 Glossary

Term	Description
Advertiser	User who advertise his free place in his room or flat.
RUP	Stands for Rational Unified Process. Process developed by IBM. Has the SRS document as a requirement for all development activities.
SRS	Stands for Software Requirements Specification. A document that completely describes all of the functions of a proposed system and the constraints under which it must operate.
Stakeholder	Any person with an interest in the project who is not a developer.
User	Participant in the application. Can either be a normal user who searches for a flat or an advertiser.

Table 1: Glossary

1.4 Stakeholders

The table 2 will give an overview of the known Stakeholders.

Name	Contact
ESE Assistant	?

Table 2: Stakeholders

1.5 System Overview

The image 1 will introduce a general system overview of the product in development.

1.6 References

2 Overall Description

2.1 Product Perspective

Flatfindr is a self-contained product that has no connections to other software.

2.2 Product Functions

- Creating / viewing ads for flats and studios
- Linking roommates to ads
- General search functionality with specific criteria
- Scheduling visits

2.3 User Classes and Characteristics

2.3.1 User

The general user that presumably is looking for a new place to live will be the largest user class. Their satisfaction depends on the ads and hereby indirectly from the Advertiser user class, as well as on crucial functionalities in the system, such as a customisable search and a smooth way of interacting with advertisers.

2.3.2 Advertiser

The user class that creates ads is called advertiser. Advertisers are the most important user class to satisfy. This depends strongly on the ad-creating interface, which should be as easy as possible to use.

2.3.3 Roommate

A roommate is someone who was linked to an ad by an advertiser. Their goals differ from the general user in the way that they generally are not looking for a new place, but are instead supporting the advertisers. Their satisfaction depends on their possibility to schedule events easily.

2.3.4 Premium User

By paying a fee for additional features, the premium user strongly increases the resources of the programming team. If the premium features are beneficial to the general user, the quantity of premium users will be optimal.

2.4 Operating Environment

Flatfindr runs on any frequently used browser, i.e. Firefox, Safari, Chrome, Microsoft Edge, Internet Explorer.

2.5 Design and Implementation Constraints

The template of Flatfindr was delivered using Java, MySQL, Hibernation and JavaScript. Changing this would present unnecessary additional effort, which is why, in this way, we are constrained by these frameworks and languages.
The software should be maintainable, so object-oriented design will be best-practice. Other than that, there seem to be no requirements, especially security.

3 External Interfaces

3.1 User Interfaces

As is shown by image 2, the user interface is straight forward due to it being a homepage.

3.2 Hardware Interfaces

The homepage will be able to be displayed on a computer, as well as mobile phones and tablets.

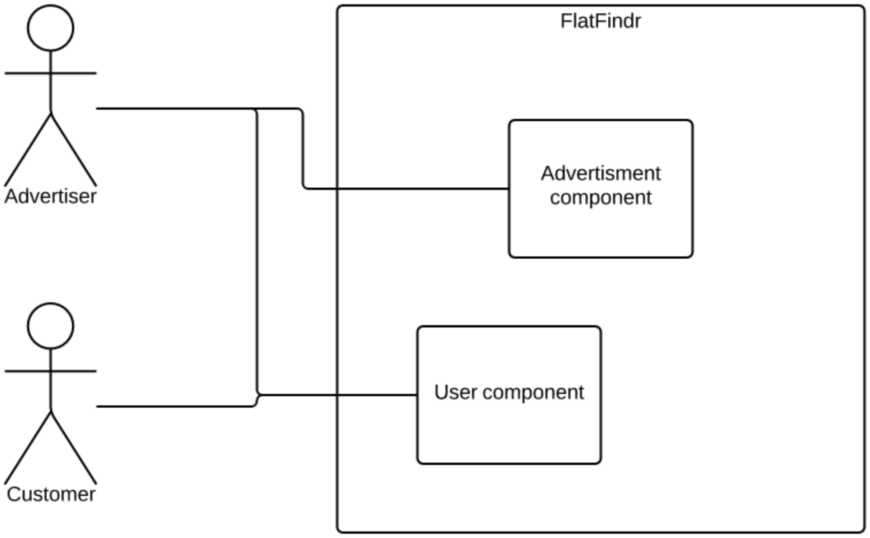
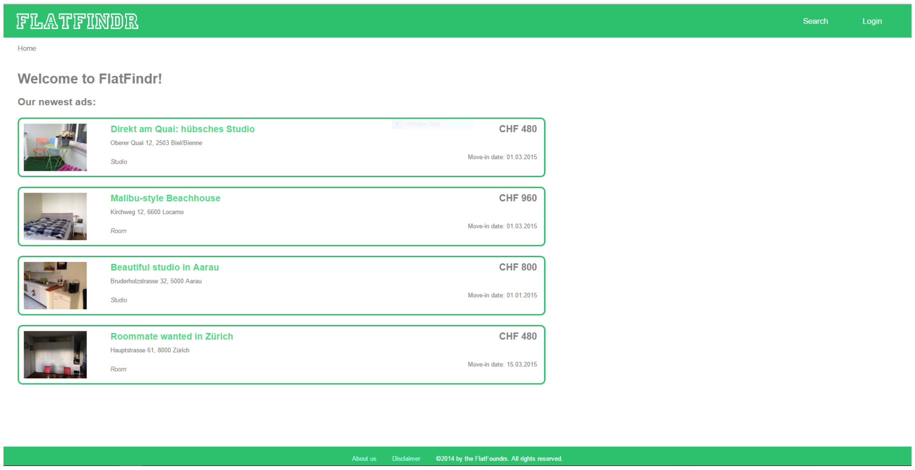


Figure 1: System Overview



Welcome to FlatFindr!

Our newest ads:

Direkt am Quai: hübsches Studio
Oberer Quai 12, 2503 Biel/Bienne
Studio
CHF 480
Move-in date: 01.03.2015

Malibu-style Beachhouse
Kirchweg 12, 6600 Locarno
Room
CHF 960
Move-in date: 01.03.2015

Beautiful studio in Aarau
Bruderholzstrasse 32, 5000 Aarau
Studio
CHF 800
Move-in date: 01.01.2015

Roommate wanted in Zürich
Hauptstrasse 61, 8000 Zürich
Room
CHF 480
Move-in date: 15.03.2015

©2014 by the FlatFoundrs. All rights reserved.

Figure 2: GUI

4 System Features

4.1 Use-Cases

Use Case ID	1
Use Case Name	Login
Trigger	A user wants to login to the webapplication
Precondition	The user has already an account on the webapplication
Basic Path	1. The user access the webapplication and selects the link "Login" 2. The user enters his credentials for his account 3. The user is redirected to the homepage
Alternative Paths	None
Postconditions	The user is logged-in
Exception Paths	In step 2, if the user enters not valid credentials into the system, the system should exit with the error message, that the credentials were wrong
Other	n/a

Table 3: Use Case 1

Use Case ID	2
Use Case Name	Post ad to sell property
Trigger	A user wants to sell a property
Precondition	The user is logged in
Basic Path	1. The user accesses the webapplication and selects the link ’Sell’ 2. The user is redirected to a form, where he can enter the information about the property he wants to sell. He can also add pictures and among other options he can select wheter he wants to sell directly or through an auction. 3. After submitting the form the user is redirected to the finished page of his ad
Alternative Paths	None
Postconditions	The user has the new ad attached to his account.
Exception Paths	None
Other	n/a

Table 4: Use Case 2

Use Case ID	3
Use Case Name	Buy property directly
Trigger	A user wants buy the property described in an ad he saw directly
Precondition	The user is logged in and on the ad page of the property he wants to buy
Basic Path	1. The user selects the link ’Buy’ 2. The user is redirected to a page with a form through which he can contact the seller 3. After submitting the message and the contact data of the buyer is forwarded to the seller and the user is redirected to the ad page
Alternative Paths	None
Postconditions	None
Exception Paths	None
Other	n/a

Table 5: Use Case 3

Use Case ID	4
Use Case Name	Bid on a property
Trigger	A user wants to bid on the auction of a property
Precondition	The user is logged in and on the auction page
Basic Path	1. The user selects the specific auction he wants to participate in 2. The user is redirected to the auction page, where information about the property and the auction itself (current bid, amount of bidders, time until the deadline etc.) is displayed 3. The user can enter a bid of his own which has to be higher than the present highest bid.
Alternative Paths	None
Postconditions	The highest bid now displays the bid made by the user
Exception Paths	In step 3 if the bid is not higher than the present one an error flashes to point the fact out.
Other	n/a

Table 6: Use Case 4

Use Case ID	5
Use Case Name	Search for porperties on sale
Trigger	A user wants to look for properties on sale
Precondition	The user is logged in and on the search page
Basic Path	1. The user selects in the search criterias that he wants to look for properties on sale 2. All properties on sale (directly or through an auction) matching his other search criterias are displayed to the user
Alternative Paths	None
Postconditions	None
Exception Paths	None
Other	n/a

Table 7: Use Case 5

Use Case ID	6
Use Case Name	Create a search alert
Trigger	A user wants to create a search alert
Precondition	The user is logged in and on the search page
Basic Path	1. The user selects the link ’Create search alert’ 2. The user is redirected to a form where he can select the search criterias he want to cover with his alert 3. After submitting the form the user is redirected to the search page
Alternative Paths	None
Postconditions	The search alert is attached to the profile of the user and he gets an email notification every time a new ad for a property matching his alert criterias is created
Exception Paths	None
Other	n/a

Table 8: Use Case 6

Use Case ID	7
Use Case Name	Delete or disable a search alert
Trigger	A user wants to delete or disable a previously created search alert
Precondition	The user is logged in and on his profile page
Basic Path	1. The user selects the link ’Search Alerts’ 2. The user is redirected to the search alerts page, where all his search alerts are displayed. He can now delete or disable each search alert via the respective buttons.
Alternative Paths	None
Postconditions	The user gets no longer notifications for disabled or deleted search alerts
Exception Paths	None
Other	n/a

Table 9: Use Case 7

5 Other Requirements

5.1 Performance Requirements

??
The system should, given a reasonable connection to the internet, be able to search and find results in a fluent manner, e.g. in at most a second. This presumably presents the greatest danger of not fulfilling user requirements, due to the non-dynamic way of loading pages out of a database.

5.2 Safety Requirements

??

5.3 Security Requirements

??
Identity authorization is nice and all, but passwords should be stored in a way that anyone who has access to the database can steal everything.