

---

# **Software Requirements Specification**

**for**

## **FlatFindr**

**Version 1.02 approved**

**Julian Schuhmacher, Joel Fimian,  
Jérôme Imfeld, Alexander Campbell**

**ISF Software Engineering**

**01.11.2016**

## Table of Contents

<b>1. Introduction.....</b>	<b>1</b>
1.1 Purpose .....	1
1.2 Stakeholders .....	1
1.3 Definitions .....	1
1.4 Scope .....	1
1.5 References .....	1
<b>2. Overall Description.....</b>	<b>2</b>
2.1 System overview .....	2
2.2 Use cases .....	3
2.3 Actor Characteristics .....	7
<b>3. Specific Requirements .....</b>	<b>8</b>
3.1 Functional Requirements .....	8
3.2 Nonfunctional Requirements .....	11

## Revision History

Name	Date	Reason For Changes	Version
Fimian Joel	16.10.16	Change and addition of use cases and requirements.	1.01
Fimian Joel	01.11.16	Update Use Cases with a numbered work flow and set a complexity and priority to each functional requirement.	1.02

# 1. Introduction

## 1.1 Purpose

This document gives an overview over a website of a real estate system and specifies its software requirements. The website should be extended with new features for buying, selling, and renting properties of all types, not only apartments or rooms. This document defines the associated functional and non-functional requirements of this new features.

## 1.2 Stakeholders

This document differentiates between two types of stakeholders, internal and external stakeholders. External stakeholders are customers, who want to buy, sell or rent different properties, and interested real estate portals, which want to include advertisements in their website. Internal stakeholders are vendors, who have to manage the different products on the website, process managers, who have to control these activities and administrators, who have to maintain the website and the associated systems, like databases, server infrastructure, etc.

## 1.3 Definitions

Ad:	Buyable or sellable product.
Advertisement:	Same as ad.
Alert:	An alert informs the user about new estates which match given criteria.
Auction:	Other way of selling product.
Bookmark:	Marking of an interesting product.
Enquiry:	Direct request to another user for visiting an estate.
Presentation:	Open presentation of an estate, where interested parties can attend
Schedule:	List of the advertiser for planned presentations of the estate.
User role:	User can have different user roles. A premium user has the advantage, that he gets alerts and results earlier than a normal user.

## 1.4 Scope

The second chapter of this document provides an overview over the project in term of use cases and actor characteristics. The third chapter describes the functional and non-functional requirements.

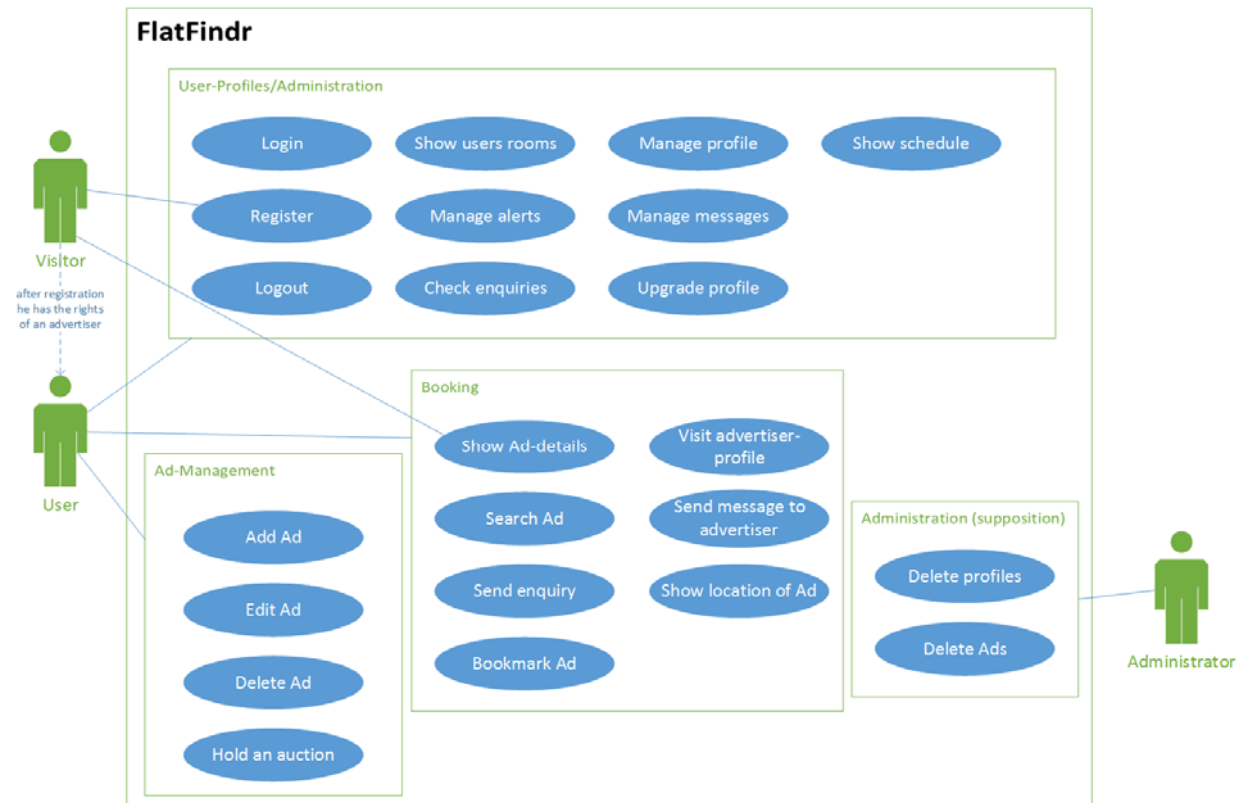
## 1.5 References

This document refers to the following documents and web-addresses:

- System Description and Use Cases:  
<https://github.com/scg-unibe-ch/ese2016/wiki/Project-Description>
- Use Case Diagram Team 2:  
[https://github.com/scg-unibe-ch/ese2016-team2/blob/master/documentation/System\\_as\\_is.pdf](https://github.com/scg-unibe-ch/ese2016-team2/blob/master/documentation/System_as_is.pdf)

## 2. Overall Description

### 2.1 System overview



16.10.2016, Version 1.01, ISF Software Engineering

The system is a website of a real estate company, which should be expanded and redesigned. A customer can buy, sell or rent apartments and rooms. The users can place advertisements, search for specific ads in a specific area and send an enquiry to an advertiser. These can manage the enquiries and organize the planned presentations and the most promising candidates. The company want to expand their markets and this has an impact on the existing website which should be expanded by the opportunity to buy, sell or rent other types of objects.

The actions on the website can be summarized in three main action groups: booking, managing the profile and managing the advertisements. The figure above shows a use case where these action groups are formed.

## 2.2 Use cases

---

<b>Number:</b>	U1
<b>Name:</b>	User-Registration in the system
<b>Description:</b>	A user visits the website of the real estate company and want to register to place advertisements or/and buy and rent different objects.
<b>Event Flow:</b>	<ol style="list-style-type: none"><li>1. User types URL into address bar</li><li>2. User clicks login button</li><li>3. User is presented with the login dialog</li><li>4. User clicks sign-up button</li><li>5. User is presented with sign-up dialog</li><li>6. User fills in form and clicks the sign up button and the Administrator verifies password strength. If password is verified, user enters login dialog again, else a message prompts for another pw with instruction.</li></ol>

---

<b>Number:</b>	U2
<b>Name:</b>	Log in to the system
<b>Description:</b>	An already registered user logs himself on to the system with his own credentials.
<b>Event Flow:</b>	<ol style="list-style-type: none"><li>1. User types in login credentials</li><li>2. User hits login button and the Administrator verifies credentials. If credentials are correct, user enters home page in logged in state, else a message prompts for reentering pw.</li></ol>

---

<b>Number:</b>	U3
<b>Name:</b>	Log out from the system
<b>Description:</b>	An already registered user logs himself out from the system.
<b>Event Flow:</b>	<ol style="list-style-type: none"><li>1. User hovers over name in top right corner</li><li>2. User chooses/clicks logout from dropdown</li></ol>

---

<b>Number:</b>	U4
<b>Name:</b>	Advertiser manages alerts
<b>Description:</b>	A registered user creates alerts with the city, object type, search radius and maximal price as criteria. He also can delete this alerts, but he can't edit already existing alerts.
<b>Event Flow:</b>	<ol style="list-style-type: none"><li>1. User hovers over name in top right corner</li><li>2. User chooses/clicks Alerts</li><li>3. User fills in criteria and the User hits Subscribe button</li></ol>

---

**Number:** U5  
**Name:** Manage profile  
**Description:** A registered user edits his existing profile by change his username, first and last name, his personal password and his personal information for the public.  
**Event Flow:**

1. User hovers over name in top right corner
2. User chooses/clicks Puplic profile
3. User clicks Edit Profile
4. User changes Profile and saves new Profile. If pw was changed, administrator verifies pw strength (It does NOT in fact !!!!), else should prompt for another pw with instructive message.

---

**Number:** U6  
**Name:** Manage messages  
**Description:** A registered user checks the received messages in the inbox. He also can watch on the messages he sent and write new messages to other users.  
**Event Flow:**

1. User hovers over name in top right corner
2. User chooses/clicks Puplic profile
3. User is presented with Inbox
4. User clicks on unread message or User clicks on New or User clicks on Sent, Unread: new message loads, New: Pop up form is shown, Sent: Sent messages list is shown, New: User fills in form (to, subject, message fields),
5. User clicks send button and the Administrator (acting as mail server) sends message to recipient.

---

**Number:** U7  
**Name:** Check enquiries  
**Description:** A registered user accepts or declines an enquiry of another user.  
**Event Flow:**

1. User hovers over name in top right corner
2. User chooses/clicks Enquiries
3. User is presented with list of enquiries
4. User clicks on Accept/Decline of each open enquiry entry or User Undos Declined or Accepted enquiries and the Administrator sends state of enquiry to enquirers.

---

**Number:** U8  
**Name:** Check visiting schedule  
**Description:** A registered user checks the schedule of presentations. He also can look at a list of all visitors of each presentation and visit the advertisement site.  
**Event Flow:**

1. User hovers over name in top right corner
2. User chooses/clicks Schedule
3. User is presented with presentations list
4. User clicks Visit to visit ad or see list to see visitors, Visit: User is presented with the real estate detail view, Visitors: User is presented with list of the visitors, Visitor: User clicks on Visit, Visitor: User is presented with the public profile of the visitor, Visitor: User clicks

message, Visitor: User fills in pop up form, Visitor: User clicks send and the Administrator deposits message in visitors user account.

**Number:** U9  
**Name:** Publication of an advertisement  
**Description:** A registered user places a new advertisement with general information, a room description, roommates, preferences, pictures and visiting times.  
**Event Flow:**

1. User logs in
2. User chooses Place an ad
3. User clicks on Place Ad
4. User types in an ad description and characteristics
5. User clicks Button Submit to publish an ad

---

**Number:** U10  
**Name:** Modification of an advertisement  
**Description:** A registered user edits an existing advertisement.  
**Event Flow:**

1. User clicks My rooms
2. User chooses one of the ads
3. User clicks on Edit Ad
4. User changes ad description and characteristics
5. User saves the changes by clicking on Save

---

**Number:** U11  
**Name:** Search advertisement  
**Description:** A user searches for an advertisement.  
**Event Flow:**

1. User clicks on Search
2. User enters object type, search radius, maximal price as criteria and different optional characteristics of the ads he is looking for
3. User clicks the button Search to search the ads
4. User now sees the results of the search
5. User can change the search properties with a filter

---

**Number:** U12  
**Name:** Visit advertiser-profile  
**Description:** A user visits over an advertisement the profile of the advertiser.  
**Event Flow:**

1. User chooses an ad
2. User clicks on Visit Profile to see information about the advertiser

---

**Number:** U13  
**Name:** Opening of an advertisement  
**Description:** A user opens an advertisement by clicking on it on the main surface.  
**Event Flow:**

1. User clicks on the ad and sees all information about it

---

**Number:** U14  
**Name:** Send enquiry  
**Description:** If a user likes an advertisement and want to buy or rent it, he sends an enquiry to the advertiser.  
**Event Flow:**

1. Users chooses an ad

2. User clicks on the button Send enquiry to advertiser

---

**Number:** U15  
**Name:** Send message to advertiser  
**Description:** A user contacts the advertiser with a message to get more information and ask additional questions.  
**Event Flow:**

1. User chooses an ad
2. User clicks on the button Contact Advertiser
3. User types in a message
4. User sends the message by clicking on the button Send

---

**Number:** U16  
**Name:** Show location of advertisement  
**Description:** A user opens Google Maps to see the location of the object by clicking on the address.  
**Event Flow:**

1. User selects an advertisement
2. User clicks on the address to get the location of the property displayed in a new window of Google Maps

---

**Number:** U17  
**Name:** Closing of a profile  
**Description:** An administrator deletes a user profile because the deletion was requested or for other reasons.  
**Event Flow:** It is possible that a user want that his profile will be deleted and therefore, the administrator has to delete the profile from the system. This use case is invented, but should be realized as a feature of the system.

---

**Number:** U18  
**Name:** Closing of an advertisement  
**Description:** An administrator deletes an advertisement because it is too old or does not fit to an internal policy.  
**Event Flow:** It is possible that a user put incorrect information in an advertisement or want to deceive a possible customer and therefore, the administrator has to delete such an advertisement from the system. This use case is invented, but should be realized as a feature of the system.

---

**Number:** U19  
**Name:** Hold an auction  
**Description:** An advertiser holds an auction for a property for sale. He defines a minimum prize and the duration of the auction.  
**Event Flow:**

1. User logs in
2. User chooses Place an ad
3. User clicks on Place Auction
4. User enters the information about the ad inclusive the information about the duration and minimum price of the property
5. User clicks the button Submit to publish the auction



---

**Number:** U20  
**Name:** Create newsletter by alert  
**Description:** A user searches for properties by different criteria and saves this filter for an automatic alert as a newsletter.  
**Event Flow:**  

1. User selects Alerts
2. User enters the criteria for the alert
3. User saves the alert with the button Subscribe

---

**Number:** U21  
**Name:** Upgrade normal profile to premium  
**Description:** A normal registered user upgrades his profile to get Prioritized position of his advertisements.  
**Event Flow:**  

1. User logs in
2. User clicks on Premium Account
3. User enters his address
4. User clicks on the button Register as Premium User to store the address and upgrade the account

---

## 2.3 Actor Characteristics

---

**Number:** A1  
**Name:** Visitor  
**Description:** This is a user, who has no profile and can not place advertisements on the website. He can search for advertisements and register in the system.

---

**Number:** A2  
**Name:** User  
**Description:** A user is registered on the system and can be a person, who wants to buy, sell or rent an object. He can do all actions described in the use cases above, except deleting a profile and an advertisement. Therefore, an advertiser is a special type of a user.

---

**Number:** A3  
**Name:** Administrator  
**Description:** The administrator can be a system administrator or the responsible manager of the system, who can, in addition to the actions of a user, delete profiles and advertisements.

---

### 3. Specific Requirements

#### 3.1 Functional Requirements

---

<b>Number:</b>	R1
<b>Description:</b>	Users can place an ad for advertising a room in a shared apartment.
<b>Complexity:</b>	- (already implemented)
<b>Priority:</b>	- (already implemented)

---

<b>Number:</b>	R2
<b>Description:</b>	Users can place an ad for advertising a studio.
<b>Complexity:</b>	- (already implemented)
<b>Priority:</b>	- (already implemented)

---

<b>Number:</b>	R3
<b>Description:</b>	Users can place an ad for advertising an house.
<b>Complexity:</b>	Medium
<b>Priority:</b>	1

---

<b>Number:</b>	R4
<b>Description:</b>	Users can sell, rent or buy properties.
<b>Complexity:</b>	Medium
<b>Priority:</b>	1

---

<b>Number:</b>	R5
<b>Description:</b>	Users can search for relevant ads in a specific area.
<b>Complexity:</b>	- (already implemented)
<b>Priority:</b>	- (already implemented)

---

<b>Number:</b>	R6
<b>Description:</b>	Users can get exhaustive information about a room (e.g. pictures, location, roommate profiles, ..).
<b>Complexity:</b>	- (already implemented)
<b>Priority:</b>	- (already implemented)

---

<b>Number:</b>	R7
<b>Description:</b>	Users can quickly send an enquiry to the ad placer.
<b>Complexity:</b>	- (already implemented)
<b>Priority:</b>	- (already implemented)

---

<b>Number:</b>	R8
<b>Description:</b>	Users can conveniently manage enquiries and organize on-site visits.
<b>Complexity:</b>	- (already implemented)

**Priority:** - (already implemented)  
**Number:** R9  
**Description:** Users can manage scheduled visits.  
**Complexity:** - (already implemented)  
**Priority:** - (already implemented)

---

**Number:** R10  
**Description:** Users can compile a list of the most promising candidates.  
**Complexity:** - (already implemented)  
**Priority:** - (already implemented)

---

**Number:** R11  
**Description:** Users can bookmark interesting ads.  
**Complexity:** - (already implemented)  
**Priority:** - (already implemented)

---

**Number:** R12  
**Description:** Users can subscribe to alerts to be informed about relevant ads.  
**Complexity:** - (already implemented)  
**Priority:** - (already implemented)

---

**Number:** R13  
**Description:** Users can register in the system.  
**Complexity:** - (already implemented)  
**Priority:** - (already implemented)

---

**Number:** R14  
**Description:** Users can log in to the system.  
**Complexity:** - (already implemented)  
**Priority:** - (already implemented)

---

**Number:** R15  
**Description:** Users can log out from the system.  
**Complexity:** - (already implemented)  
**Priority:** - (already implemented)

---

**Number:** R16  
**Description:** Users can manage their profile information.  
**Complexity:** - (already implemented)  
**Priority:** - (already implemented)

---

**Number:** R17  
**Description:** Users can contact an ad-placer by sending a message.  
**Complexity:** - (already implemented)  
**Priority:** - (already implemented)

<b>Number:</b>	R18
<b>Description:</b>	Users can manage their messages.
<b>Complexity:</b>	- (already implemented)
<b>Priority:</b>	- (already implemented)
<hr/>	
<b>Number:</b>	R19
<b>Description:</b>	Users can put an ad to sell properties directly or through an auction.
<b>Complexity:</b>	High
<b>Priority:</b>	2
<hr/>	
<b>Number:</b>	R20
<b>Description:</b>	Users can search for properties for sale similarly to the search form for properties for rent.
<b>Complexity:</b>	Medium
<b>Priority:</b>	3
<hr/>	
<b>Number:</b>	R21
<b>Description:</b>	Users can create search alerts which notify him about new ads corresponding the search criteria.
<b>Complexity:</b>	Medium
<b>Priority:</b>	3
<hr/>	
<b>Number:</b>	R22
<b>Description:</b>	Users can search for properties with more filter criteria.
<b>Complexity:</b>	Medium
<b>Priority:</b>	2
<hr/>	
<b>Number:</b>	R23
<b>Description:</b>	Users can upgrade their normal to premium profiles to get alerts and results earlier. The adds of premium user are a bit higher in the result list.
<b>Complexity:</b>	High
<b>Priority:</b>	2
<hr/>	
<b>Number:</b>	R24
<b>Description:</b>	Users can see in a private area that he has a premium account.
<b>Complexity:</b>	Low
<b>Priority:</b>	3
<hr/>	
<b>Number:</b>	R25
<b>Description:</b>	Administrators can delete profiles and ads.
<b>Complexity:</b>	Low
<b>Priority:</b>	4

<b>Number:</b>	R26
<b>Description:</b>	The UI of the website looks and feels old style and needs to be refurbished into a more elegant and modern shape.
<b>Complexity:</b>	High
<b>Priority:</b>	1

---

## **3.2 Nonfunctional Requirements**

### **3.2.1 Security**

- Integrity of user data
- Safe keeping of user data in terms of Swiss data protection law

### **3.2.2 Speed/Performance**

- Visitor can register within three minutes
- Search process completed with a reply within three minutes