Software Requirements Specification

for

HomeLender

ESE Team 6

Michael BAUR

Maëlle BOUGHATTAS

Valerie HAFTKA

Patrick INDERMÜHLE

Bernhard ZAHND

Final version – 12.12.2016

Table of Contents

1. Introduction 1

1.1 Purpose 1

1.2 Intended Audience and Reading Suggestions 1

1.3 Product Scope 1

1.4 Abbreviations and acronyms 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 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 6

4.2.1 Description and Priority 6

4.2.2 Stimulus/Response Sequences 6

4.2.3 Functional Requirements 6

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 7

4.4.1 Description and Priority 7

4.4.2 Stimulus/Response Sequences 7

4.4.3 Functional Requirements 7

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

4.6 Manage Enquiries 8

4.6.1 Description and Priority 8

4.6.2 Stimulus/Response Sequences 8

4.6.3 Functional Requirements 8

4.6.4 Non-Functional Requirements 8

4.7 Bookmarking ads 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 Manage alerts 9

4.8.1 Description and Priority 9

4.8.2 Stimulus/Response Sequences 9

4.8.3 Functional Requirements 9

4.8.4 Non-Functional Requirements 9

4.9 Extended search options TO MERGE WITH SEARCHING CRITERIA ? 9

4.9.1 Description and Priority 9

4.9.2 Stimulus/Response Sequences 9

4.9.3 Functional Requirements 9

4.10 Premium and Normal membership 10

4.10.1 Description and Priority 10

4.10.2 Stimulus/Response Sequences 10

4.10.3 Functional Requirements 10

4.10.4 Non-Functional Requirements 10

5. Other Nonfunctional Requirements 11

5.1 Performance Requirements 11

5.2 Safety Requirements 11

5.3 Security Requirements 11

5.4 Software Quality Attributes 11

5.5 Business Rules 11

6. Other Requirements 12

# Introduction

## Purpose

This is the Software Requirements Specification document for version number 3.0. 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.

## Intended Audience and Reading Suggestions

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

## Product Scope

The product in question is HomeLender, an online website to find places for rent and for sale. The sale can be direct or through auction. The software offers the possibility to manage advertisements (display, create, edit, delete and bookmark) as well as manage enquiries to visit the properties (enquiries, schedule), manage alerts when the type of property looked for does not exist, and dispose of an internal messaging.

The user of HomeLender is able to login by creating an account in the site, as well as using Facebook or Google account. When the user is logged in, he is able to provide a short description of him(her)self and a picture.

HomeLender’s major benefit is its user-interface.

## Abbreviations and acronyms

*SRS* stands for *Software Requirements Specification.*

*Ad* stands for *Advertisement.*

## References

Initial project description from the customer:

https://github.com/scg-unibe-ch/ese2016/wiki/Project-Description  
  
Documentation of the system:

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 FlatFoundrs and that requires maintenance in order to expand its features. This is an absolute condition for HomeLender to remain competitive on the market.

MUST AD ALL THE INTERFACES OF THE SYSTEM

## Product Functions

The main product functions have been already quoted in 1. 1. 3. :

* Manage ads : display, create, edit, bookmark, delete
* Search a place for rent or for sale (with some other features)
* Manage enquiries to schedule visits (visits for the user’s properties or properties the user is interested in)
* Manage alerts (create, delete, receive a message) for properties the user looks for and that are not available on HomeLender
* Internal messaging
* Manage an account (create, add some personal information) with 2 types of membership (premium and normal)

## User Classes and Characteristics

There are two types of users:

* Users that are not logged in. They can see ads on HomeLender, but cannot neither contact other users nor schedule room visits;
* Users that are logged into their HomeLender account or Facebook account or Google account. They can schedule meetings with other owners and create ads for their own property.

## Operating Environment

The software is written in Java (IDK 8.0 or +) and runs on a virtual server. It creates a local 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.

## Assumptions and Dependencies

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

# External Interface Requirements

## User Interfaces

On the **index page (homepage)**, the user can see a list of the newest ads, sorted by “status” (for rent or for sale). Clicking on the advertisements moves the user to the ad description page. The index page also provides a button to log in and a button to sign up.

On the **ad description page**, the user can see all relevant data about the property in question. If the user is logged in, he can send enquiries for visits to the owner of the property, 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 page**.

On the **search page**, the user can select the search requirements. The user can search either a place for rent or a place for sale, select a zip code and a city with a radius of search, a maximum price for each status (for rent or for sale).

On the **login page**, the user can type in his/her email address and password, and click on the login button to log in. The login page is a little window in the middle of the screen except when the user has arrived on the login page via the ad description page. The user is also able to log in with his/her Google account or Facebook account.

On the **signup page**, the user can enter his first name, last name, password, email address, gender and kind of membership, and click on the sign up to create an account.

The **header**, visible from all pages, is composed of the name of the website (HomeLender), which moves the user to the homepage, the search button which moves the user to the search page and:

* If the user is not logged in, a login button on the top right which moves the user to the login page;
* If the user is logged in, the username and the user’s profile picture that brings the user to his/her **personal profile page** (that can be edited in the **edit profile page**). A drop-down 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 **ad placement page** (form to place an ad), the **my rooms page** (list of the ads the user owns), the **my bookmarks page** (list of the ads the user has bookmarked), the **schedule page** (list of visits scheduled), the **message page** (internal messaging between users), the **enquiries page** (list of enquiries received for the user’s properties), the **alerts page** (create and manage alerts) 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

he 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 placeAd 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, it means that the user must not be able to create an advertisement that does not have the minimum necessary data.

ADD RENT/SALE

## Seeing advertisements

### Description and Priority

Ability of users to see advertisements created by them self or other users has a priority of **10** (scale : 1 to 10).

### Stimulus/Response Sequences

When the user is on the index page, he should see a list of ads, sorted by “status” (for rent or for sale).

### Functional Requirements

* *REQ-FetchAdd*: the software must be able to send the information regarding 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 them self or other users has a priority of **9**.

### Stimulus/Response Sequences

When the user selects the search option, the search 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 on the result page.

### 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-searchdisplayfilter-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. ADD RENT/SALE

## Inspect advertisements

### Description and Priority

Ability of users to get more detailed information about an ad 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 property (price, location, number of rooms, surface area, short description, additional features, 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

Ability of users to send an enquiry has a priority of **7**.

### Stimulus/Response Sequences

When the user selects an ad, he should be able to send an enquiry to the owner of the property and to find a list of all enquiries received.

### 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.

## Manage Enquiries

### Description and Priority

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

### Stimulus/Response Sequences

When the user enters the enquiry page, 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 ads

### 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, he should be able to bookmark it and to find a list of all his/her bookmarks.

### 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.

## Manage alerts

### Description and Priority

Ability of users to create and manage alerts for properties he/her looks for has a priority of **7**.

### Stimulus/Response Sequences

The user should be able to create and delete alerts on a specific page. AD IF MESSAGE

### Functional Requirements

* *REQ-subscriptionSave*: the software must be able to save all the alerts of a user.
* *REQ-subscriptionDisplay*: the software must be able to get and display all alerts of a user.
* *REQ-subscriptionAlert*: the software must be able to send a message to the user if an ad has been created that fits to the user’s 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.
* *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.

## Extended search options TO MERGE WITH SEARCHING CRITERIA ?

### Description and Priority

Ability of users to use additional search criteria (such as …)

### Stimulus/Response Sequences

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

based on the type of deal and the amount of rooms of the flat.

### Functional Requirements

* *REQ-searchForType*: There must be a search option to filter based on the type of deal
* *REQ-searchForRoomNr.*: There must be a search option to filter based on the amount of rooms.

## Premium and Normal membership

### Description and Priority

Ability of users to choose between two types of membership (Premium and Normal), whose Premium membership enables to get alerts sooner than Normal and to put the user’s ads higher up in the list of ads, has a priority of **8**.

### Stimulus/Response Sequences

When the user signs up there should be the option to create a Premium account. When the user selects this kind of membership and completes the form, the user will receive the payment instructions for the Premium membership by message.

### 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 clients 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 are 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 owners or contact other owners.

# 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