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 Sign up 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 Premium and Normal membership 5

4.2.1 Description and Priority 5

4.2.2 Stimulus/Response Sequences 5

4.2.3 Functional Requirements 5

4.2.4 Target Requirements 5

4.3 Log in (different types of account) 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 Log out 6

4.4.1 Description and priority 6

4.4.2 Stimulus/Response sequences 6

4.4.3 Functional requirements 6

4.5 Display profiles 6

4.5.1 Description and priority 6

4.5.2 Stimulus/Response sequences 6

4.5.3 Functional requirements 7

4.6 Edit personal data 7

4.6.1 Description and priority 7

4.6.2 Stimulus/Response sequences 7

4.6.3 Functional requirements 7

4.6.4 Non-functional requirements 7

4.7 Create an ad 7

4.7.1 Description and Priority 7

4.7.2 Stimulus/Response Sequences 7

4.7.3 Functional Requirements 8

4.7.4 Non-Functional Requirements 8

4.8 See recent ads 8

4.8.1 Description and Priority 8

4.8.2 Stimulus/Response Sequences 8

4.8.3 Functional Requirements 8

4.8.4 Non-Functional Requirements 8

4.9 Inspect advertisements 8

4.9.1 Description and Priority 8

4.9.2 Stimulus/Response Sequences 8

4.9.3 Functional Requirements 8

4.10 Edit an ad 9

4.10.1 Description and Priority 9

4.10.2 Stimulus/Response Sequences 9

4.10.3 Functional Requirements 9

4.10.4 Non-Functional Requirements 9

4.11 Delete an ad 9

4.11.1 Description and Priority 9

4.11.2 Stimulus/Response Sequences 9

4.11.3 Functional Requirements 9

4.11.4 Non-Functional Requirements 9

4.12 Search for an ad 9

4.12.1 Description and Priority 9

4.12.2 Stimulus/Response Sequences 10

4.12.3 Functional Requirements 10

4.13 Extended search options 10

4.13.1 Description and Priority 10

4.13.2 Stimulus/Response Sequences 10

4.13.3 Functional Requirements 10

4.14 Send an enquiry 10

4.14.1 Description and Priority 10

4.14.2 Stimulus/Response Sequences 10

4.14.3 Functional Requirements 10

4.15 Manage Enquiries 11

4.15.1 Description and Priority 11

4.15.2 Stimulus/Response Sequences 11

4.15.3 Functional Requirements 11

4.16 Manage visits (schedule) 11

4.16.1 Description and Priority 11

4.16.2 Stimulus/Response Sequences 11

4.16.3 Functional Requirements 11

4.17 Bookmark ads 11

4.17.1 Description and Priority 11

4.17.2 Stimulus/Response Sequences 11

4.17.3 Functional Requirements 12

4.18 Manage alerts 12

4.18.1 Description and Priority 12

4.18.2 Stimulus/Response Sequences 12

4.18.3 Functional Requirements 12

4.18.4 Non-Functional Requirements 12

4.19 Manage messages 12

4.19.1 Description and Priority 12

4.19.2 Stimulus/Response Sequences 12

4.19.3 Functional Requirements 12

4.20 Buy a property through auction 13

4.20.1 Description and Priority 13

4.20.2 Stimulus/Response Sequences 13

4.20.3 Functional Requirements 13

4.20.4 Non-Functional Requirements 13

4.20.5 Target requirements 13

4.21 See bids that have been made 13

4.21.1 Description and Priority 13

4.21.2 Stimulus/Response Sequences 13

4.21.3 Functional Requirements 13

5. Other Nonfunctional Requirements 14

5.1 Performance Requirements 14

5.2 Safety Requirements 14

5.3 Security Requirements 14

5.4 Software Quality Attributes 14

5.5 Business Rules 14

5.6 Legal requirement 14

# Introduction

## Purpose

This is the Software Requirements Specification document for *HomeLender* 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 log in by creating an account on 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(s)* stands for *Advertisement(s).*

Priority scale: from 1 to 10.

## 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.

## 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);
* Buy properties directly or through auction (the transaction is not made on the site).

## User Classes and Characteristics

There are two types of users:

* Users that are not logged in. They can see ads on HomeLender and search ads, but do not have access to other features;
* Users that are logged into their HomeLender account or Facebook account or Google account. They have access to all the features described in **4..**

## 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. From the homepage, the log in is made in a pop-up window, where there also is 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 pop-up window from the homepage. Otherwise, it is a normal page (the user has the possibility to log in from many pages). 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 ads page** (list of the ads the user owns), the **my bookmarks page** (list of the ads the user has bookmarked), the **my auction** page (list of all bid made and all user’s auctions), 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

## Sign up

### Description and priority

Ability of a user to create an account quickly has a priority of **10**.

### Stimulus/Response sequences

The user can fill a short form with minimal data (names, password, e-mail address, gender, type of membership). When he clicks on “sign up”, he becomes a member of *HomeLender* and has access to extended features.

### Functional requirements

* The software must be able to recognize if the names and/or the e-mail address are(is) already in the data base (error messages are displayed).
* The software must be able to save the entries in the data base.
* If the user has chosen the Premium membership, the software must be able to send him/her a message with the instructions for the payment.

### Non-functional requirements

* The user should not be able to have several accounts.

## 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, the user is able to create a Premium account. If the user chooses this membership, he/she should receive the payment instructions by message.

### Functional Requirements

* The software must be able to display both options (Normal and Premium).

### Target Requirements

* The software must be able to ensure a secure online money transaction.
* The software must keep track of the amount of time left for the premium membership of the user and revoke premium privileges once that time has passed.

## Log in (different types of account)

### Description and priority

Ability of a user to log in has a priority of **10**.

### Stimulus/Response sequences

The user is able to log in with a *HomeLender* account previously created, or a Google account, or a Facebook account. When the user clicks on the button *login*, he/she can choose between these 3 options. If the information required are correct, the user should be able to access his/her account.

### Functional requirements

* The software must be able to recognize the input data to log in the user.
* The software must be able to display the correct account after logging in.

### Non-functional requirements

* The user should not be able to log in with wrong credentials.

## Log out

### Description and priority

Ability of a user to log out has a priority of **10**.

### Stimulus/Response sequences

The user is able to log out quickly from *HomeLender*, whatever the type of account used (*HomeLender*, Google, Facebook).

### Functional requirements

* The software must be able to display the homepage after logging out.

*It is accepted that the user is logged in to use the following features, except for points 4.12. and 4.13..*

## Display profiles

### Description and priority

Ability of a user to display his/her own profile and profiles of other users has a priority of **8.**

### Stimulus/Response sequences

By clicking on his/her name (top right), the user is able to see his/her personal information (username, name, kind of membership, short personal description). By clicking on an ad that it not his/hers, the user is able to display the profile of the owner (by clicking on a button).

### Functional requirements

* The software must be able to display the right profile (personal profile or profile of another user).

## Edit personal data

### Description and priority

Ability of a user to edit his/her personal data and public profile has a priority of **9**.

### Stimulus/Response sequences

When the user is on his/her profile, he/she is able to edit this latter (username, names, password, kind of membership, short description). The data have been modified once clicked on “Update”.

### Functional requirements

* The software must be able to save the entries and to display the new data in the profile.
* The software must be able to check the entries according to the same process it does when creating an account.

### Non-functional requirements

* The user should not be able to input entries that are not correct regarding **4.1.3.**.
* The user should not be able to edit personal data of other users.

## Create an ad

### Description and Priority

Ability of a user to create an ad for a property he/she owns of **10**.

### Stimulus/Response Sequences

The user must be able to display the place ad page, where he/her can fill a form (with relevant data) to place his/her ad on HomeLender. Particularly, the user can choose between 3 options, regarding the “status” of his/her property: for rent, for sale or auction (with a quick buy option).

When the user is done, a validation button can be pressed, which creates an entry in the data base for his/her ad. The user can also cancel his/her request.

### Functional Requirements

* The software must be capable to distinguish logged in users and users who are not logged in.
* The *place ad* page must be accessible.
* The *place ad* page must have textboxes, choice boxes or other input interfaces that can collect relevant data.

### Non-Functional Requirements

* 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.

## See recent ads

### Description and Priority

Ability of users to see recent ads that have been created by him/her-self or by other users has a priority of **7.**

### 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). Then, the user has access to these ads in each ad description page.

### Functional Requirements

* The software must be able to send the information regarding ads entries to the user.
* *T*he software must be able to display to the user the data of the advertisements sent to user.

### Non-Functional Requirements

* The software must be able to send the data within 10 seconds of the request in order to not keep the user waiting. This requirement is a general requirement for the whole software. From now, it is accepted that it is an implicit requirement.

## Inspect advertisements

### Description and Priority

Ability of users to get more detailed information about an ad has a priority of **10**.

### 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

* The software must be able to get and display all the data available about an ad.

## Edit an ad

### Description and Priority

Ability of user to edit an ad he/she created has a priority of **8**.

### Stimulus/Response Sequences

A button is available on each user’s ad to edit this latter. The user is able to see the form he/she filled previously, to validate the changes or to cancel the changes.

### Functional Requirements

* The software must be able to display all the previous inputs of the user. There should not be any mistakes in the data.
* The software must be able to save the changes and to display these changes in the *ad description* page.

### Non-Functional Requirements

* The user should not be able to create invalid entries, with respect to the same rules as when he/she creates an ad.

## Delete an ad

### Description and Priority

Ability of the user to delete an ad has a priority of **7**.

### Stimulus/Response Sequences

A button is available on each user’s ad to delete this latter. The user is able to have a success message when the ad is deleted.

### Functional Requirements

* The software must be able to remove the entry in the data base and in the *My ads* page.
* The software must be able to display a success message when the ad is deleted.

### Non-Functional Requirements

* The user should not not be able to see that his/her ad has been successfully deleted.

## Search for an ad

### Description and Priority

Ability of the user to search for ads created by him/her-self or other users has a priority of **10**.

### Stimulus/Response Sequences

When the user selects the search option, the search page should open so that the user can specify what to search for. Once the user confirms the search, he/she should see the results of his search on the result page.

### Functional Requirements

* The software must be able to display a form to input search criteria (price, city, surface area, number of rooms, status)
* The software must be able to get the data the user has given to filter all hits on the data base.
* The software must be able to display correct results of search.

## Extended search options

### Description and Priority

Ability of user to choose additional search criteria has a priority of **7.**

### Stimulus/Response Sequences

When the user makes a first search and has plenty of results, he/she is able to add new search criteria to find the

ad that fits the best to what the user is looking for.

### Functional Requirements

* There must be all criteria that are available when creating an ad.
* The software is able to display all the results corresponding to the right criteria.

## Send an 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/she should be able to send an enquiry to the owner of the property.

### Functional Requirements

* The software must be able to save the enquiry the user has sent.
* 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. The user has the possibility to accept or decline enquiries, and sees if an enquiry has been accepted or not.

### Functional Requirements

* The software must be able to get, format and display all the enquiries a user has pending.

## Manage visits (schedule)

### Description and Priority

Ability of users to manage his/her visits has a priority of **5**.

### Stimulus/Response Sequences

The user is able to see, on the Schedule page, all the visits scheduled for his/her properties, as well as all the visits scheduled to which the user participates. For the user’s presentations of his/her property, he/she can see the list of all subscribed visitors (for each visit) and rate them.

### Functional Requirements

* The software must be able to format all the visits in two clear arrays.

## Bookmark ads

### Description and Priority

Ability of users to add a place 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 (*My bookmarks* page).

### Functional Requirements

* The software must be able to save all bookmarks of an individual user.
* The software must be able to get and display all bookmarked ads of a user.

## Manage alerts

### Description and Priority

Ability of the user 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, and receive a message when an ad that fits to the alert has been created.

### 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.

## Manage messages

### Description and Priority

Ability of the user to send a message, read a new message, reply to a message and to see old messages has a priority of **7**.

### Stimulus/Response Sequences

The user is able to send a message to another user by 3 ways: on the other user’s public profile, on the other user’s ad or directly from the internal messaging. In the *Messages* page, the user is able to use all the mentioned functionalities.

### Functional Requirements

* The software is able to transfer a message from a user to another, to display it and to save the messages.

## Buy a property through auction

### Description and Priority

Ability of the user to make a bid for an auction has a priority of **8**.

### Stimulus/Response Sequences

When the user is on an ad that is for sale through auction, he/she is able to make a bid and/or to buy directly the property. The direct sell is materialized with a button that ends the auction, but no money transaction is made on the site.

### Functional Requirements

* The software must be able to display the current bid and the name of the user that has made this bid.
* The software must be able to display a timer (expiration of the offer).
* When the user bought the property directly, the software should be able to stop the auction.
* The software must be able to send a message to the seller and to the buyer when there is an instant buy done or a bid made.

### Non-Functional Requirements

* The user should not be able to make a bid smaller than the current bid.
* The user should not be able to make a bid smaller than the current bid plus the increment (given by the owner).

### Target requirements

* The seller should be informed by message if nobody made a bid.
* The seller and users who made bids should be informed by message when the auction is over.

## See bids that have been made

### Description and Priority

Ability of the user to see all the bids he/she made has a priority of **6**.

### Stimulus/Response Sequences

The user is able to find the list of all ads he/she bid for on the *My auction* page. On this page, he/she is also able to see all his/her own auctions.

### Functional Requirements

* The software must be able to display the correct information in *My auction* page.

# 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.

## Legal requirement

It is important that confidentiality of the private user data is guaranteed. The data base must be as secured as possible.