Requirements and Analysis Document for Studentbostäder Göteborg

Version: Final version
Date: 28/5-17
Author: Jonathan Gildevall, Peter Gärdenäs, Erik Magnusson, John Segerstedt

This version overrides all previous versions

1 Introduction

There exists no single application which gathers the student homes from the two major student accommodations sites SGS and Chalmers. This results in users having to juggle the output of two different accommodation sources.

The solution to this issue is simply to gather both sites in one app, optimized for use on the go. Students will no longer need to struggle with poorly implemented mobile versions of websites and the lacking multitasking capabilities of a phone.

The application is built to suit your mobile needs and aim to be as little in your way as possible. With features like notifications when accommodations that suit your preferences are published and powerful search tools to let you choose what's important.

1.2 Definitions, acronyms and abbreviations

Chalmers - Chalmers Studentbostäder, a student housing host.

Enum: Enumerated type, a data type consisting of a set of named values

Post/Get Request: Network requests. Used to receive data from web locations.

Search watcher - A self autonomous service that observes an object feed until certain requirements are met.

SGS - Stiftelsen Göteborgs Studentbostäder, a student housing host.

Studentbostäder Göteborg: The name of the application this document concerns.

UI: User interface, the interface the user is presented with when running an application.

2 Requirements

2.1 User interface

When the user starts the app it should be presented by the search view, see Appendix 1. Here the user will be able to casually browse accommodations by sliding up and down, favoriting by either tapping the star icon or sliding the accommodation object sideways, sort the list, and search by using the search field in the middle top. More advanced filters and search options should be available in an advanced search view, see Appendix 2.

The second main view is the favorite view where the user is presented with their list of already favorited accommodations, see Appendix 3. It must be possible to scroll through the list, remove favorites and open the object view of an item in the list.

The third main view is the search watcher view where the user is presented with a list of their already created search watchers in a list, see Appendix 4. There must be an edit button and a search button on each search watcher. There has to be a button at the top where the user can create new search watchers.

The last main view is the settings view, it should display simple settings for the user such as clearing all search watchers.

It must be easy to navigate between the four main views for a mobile user and this navigation must be found in the same place in all main views. The UI must provide multiple ways of doing the most common tasks such as favoriting accommodations, creating search watchers and searching.

2.2 Functional requirements

Users, or the application, must be able to do the following:

The application must be able to get accommodation data from at least two different housing providers.

The user must be able to search for student accommodations in two ways using a main search field in the search view, and in the advanced search view with at least the following additional search attributes: max and min price, max and min living area, days since the accommodation got uploaded, days until the accommodation gets taken down, housing type, region, and landlord.

The user must have the ability to create autonomous search watchers that periodically checks if a new accommodation fits the defined filters and then alerts the user. This must work even if the user does not have the application running. The user must be able to edit these search watchers, changing their search terms.

Tapping on an accommodation will open a detailed page displaying all additional details about the accommodations such as a larger image. Here the user can swipe left or right to see other accommodations from the list they came from.

The user must be able to apply to an accommodation by tapping the apply button which takes the user to the object's webpage.

The user must be able to save accommodations for later, meaning that they can find them in the favorite view, either by tapping the star or swiping it in the search view.

In the search watcher view the user must be able to browse their search watchers, edit them, use them to conduct searches, and create new ones.

2.3 Non-functional requirements

2.3.1 Testability

The application has to be testable and 100% of the model methods has to be covered by adequate tests.

2.3.2 Responsive

Performance is extra important since it's a mobile app and users should be able to use the application during very short sessions. To achieve a quick start up time the local dataset will not sync with the database on start-up. Instead, the data is synced automatically every six hours. The workload must be split on as many threads as possibly to alleviate the main thread and to increase performance.

2.3.3 Packaging and installation

To run the application all that should be necessary to do is acquire the .apk file and install it on an Android device with a sufficient Android version.

2.3.4 Usability

The application must feel like other android applications in order to not confuse the user. This means using standard gestures and navigation placements. Usability requirements are not too high though as the targeted audience is university students who tend to be experienced mobile users.

3 Use cases

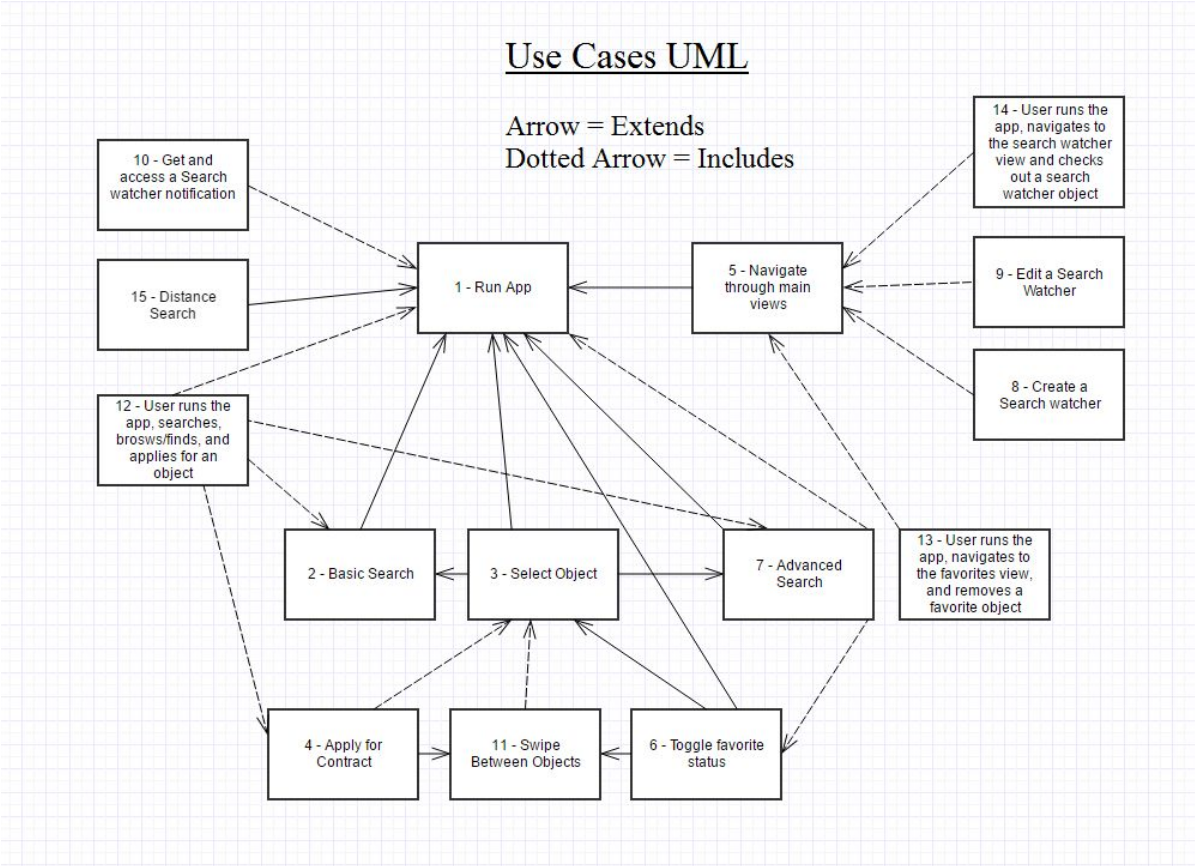


Figure 1: Use cases UML for the application

3.1 Use case listing

List of implemented use cases in order of priority, highest to lowest.

1-Run App
2 -Basic Search
3-Select Object
4-Apply for Contract
5-Navigate through main views
6-Toggle favorite status
7-Advanced Search
8-Create a Search watcher
9-Edit a Search watcher
10-Get and access a Search Watcher notification
11-Swipe Between Objects

12 User runs the app, searches, browses/finds, and applies for an object
13 User runs the app, navigates to the search watcher view and checks out a search watcher object
14 User navigates to the favorites view, and removes a favorite object
15 Distance Search

(See Appendix 7)

4 Domain model

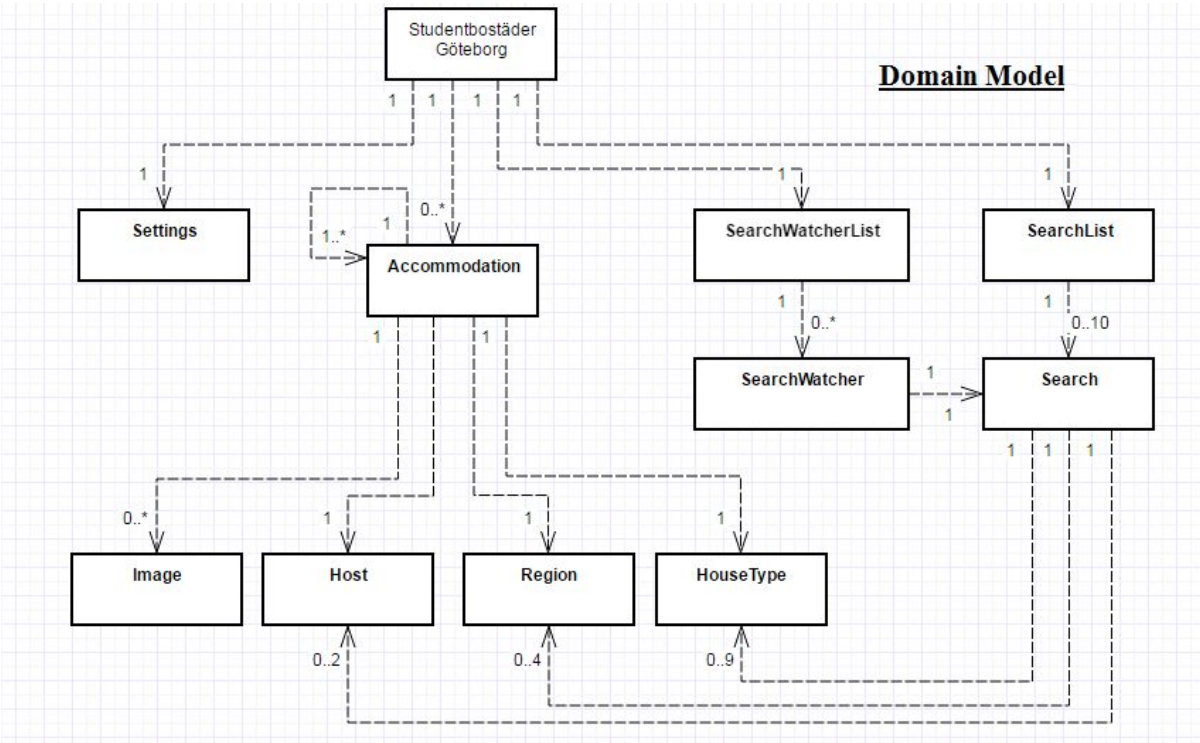


Figure 2: A UML class diagram of the domain model.

4.1 Class responsibilities

Accommodation: Holds information about an Accommodation either from SGS or Chalmers, also holds all Accommodations.

Host: A custom enum class that represents the two different housing hosts; Chalmers and SGS.

HouseType: A custom enum class that represents the different housing types: one room and kitchen, one room with shared kitchen, etc.

Image: Holds images so they only need to be loaded once per session

Region: A custom enum class that represents different parts of Gothenburg. Ex: North, West, etc.

Search: Responsible for performing searches filtering Accommodations.

SearchList: Responsible for creating Search objects and storing them in a “previous searches” list.

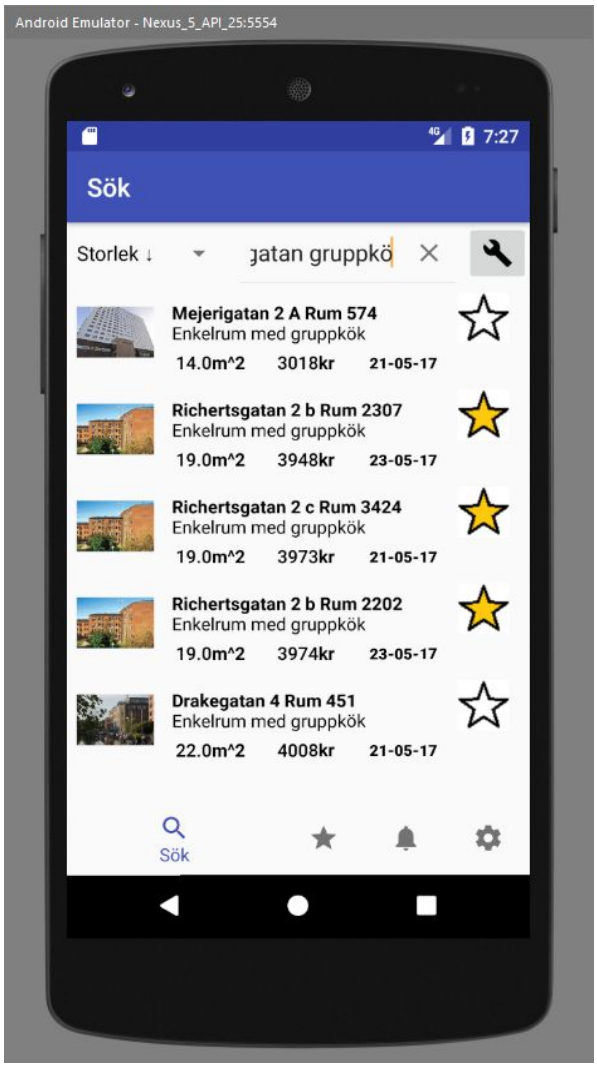
SearchWatcher: Holds a Search and performs stores new matches to that search.

SearchWatcherList: Holds all SearchWatcherItems that are shown in SearchWatcherActivity

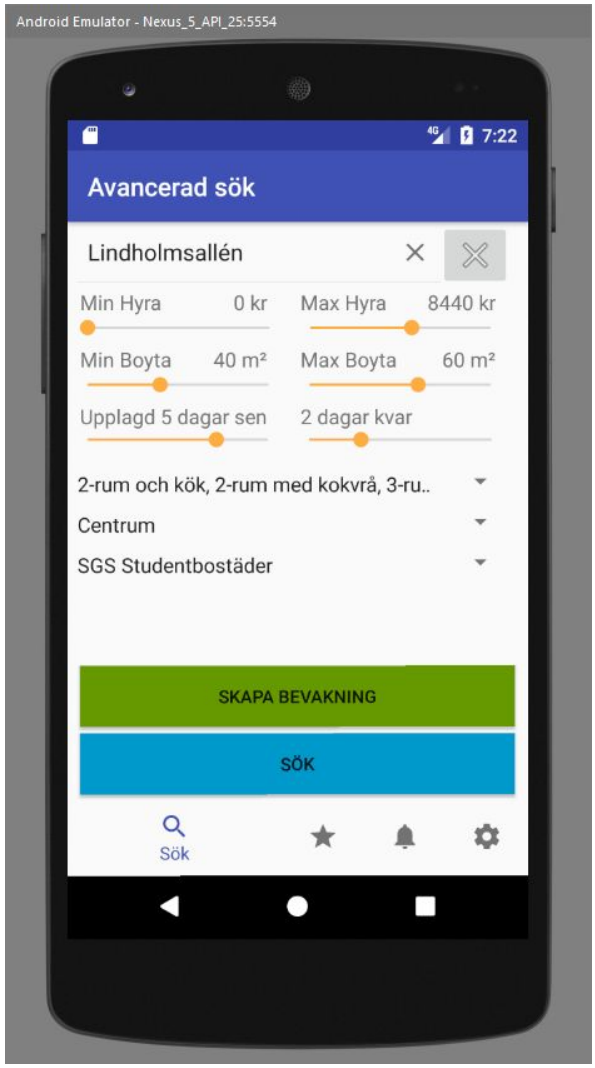
Settings: Stores settings

5 Appendix

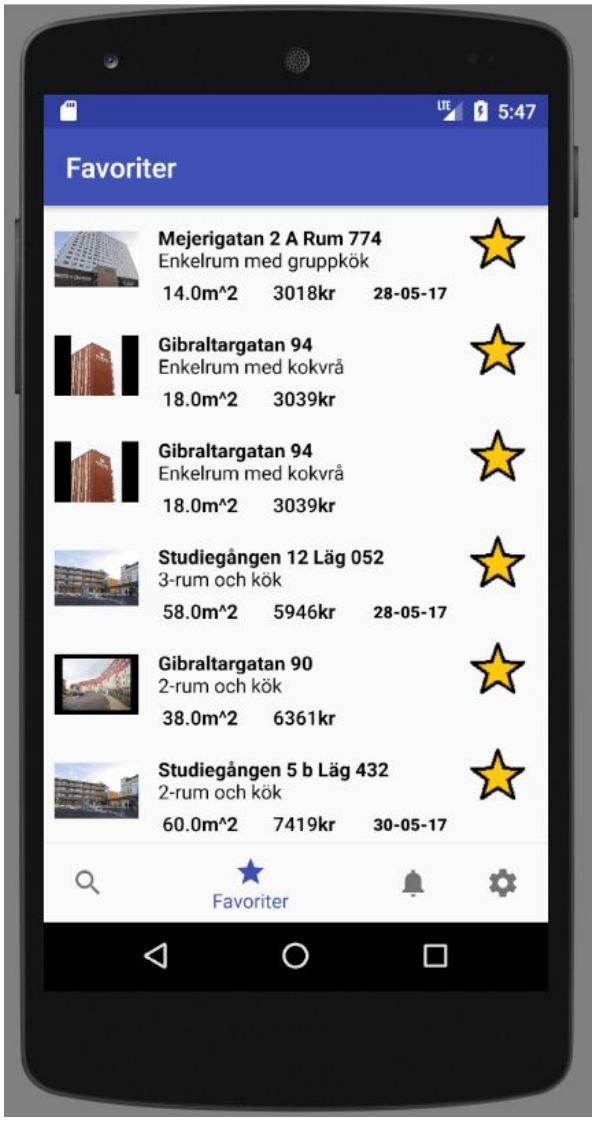
Appendix 1: The search view.



Appendix 2: The advanced search view

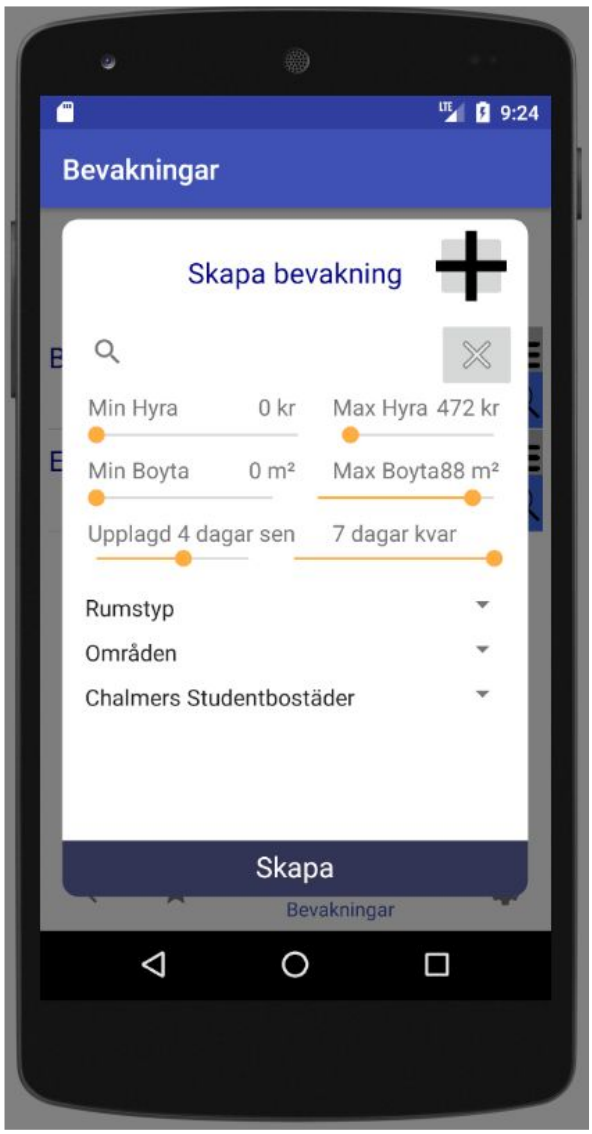


Appendix 3: Favorites view



Appendix 4: Search watcher view

Appendix 5: Create new search watcher modal



Appendix 6: New search watcher name dialog

Appendix 7:

UC: 1-Run app

Summary: The user wants to start the app and therefor taps the app icon or push notification
Priority: high
Participants: User
Normal flow of events: The user taps the icon of the app on his device and the app launches

	Actor	System
1a	The user taps the icon of the app on his device	
1b	The user taps the push notification.	
2a		The system launches the application and the user ends up in the search view.
2b		The system launches the application and the user ends up in the search watcher view.

UC: 2-Basic Search

Summary: A user conducts a search for housing using the normal search bar
Priority: high
Extends: 1-Run app
Participants: User
Normal flow of events: The user uses the main search bar to conduct a normal search, the result is display.

	Actor	System
1	The user selects the search bar in the search view, writes something and then presses the search button.	
2		The system filters the database by using the user input and then display the matched accommodations in the result list.

UC: 3-Select Object

Summary: The user taps on a search object.
Priority: high
Extends: 1-Run app, 2-basic search, 7-Advanced Search
Participants: User
Normal flow of events: The user taps on a search object and the system takes the user to the objects page.

	Actor	System
1	The user taps on a search object displayed in either: • The search result panel. • The list of favorites • The list of search watcher matches	
2		The system takes the user to another panel where more information about the object is displayed.

UC: 4-Apply for contract

Summary: The user chooses an application and presses the apply button in a objects detailed information view and then the system opens a browser window with the correct objects page.
Priority: high
Extends: 11-Swipe between objects
Includes: 3-Select Object
Participants: User
Normal flow of events: The user select an accommodation and presses the Apply button in an object's detailed information view and then the system opens a window with the correct object's web page.

	Actor	System
1	3-Select Object	
2	The user presses the apply button in an object's detailed information view.	
3.1		The system opens a browser window of the correct page from the object's origin page.
3.2		The system opens a browser window of the login page of the object's provider.

Alternative flow

The user browser the accommodation until an interesting accommodation is found.

	Actor	System
1	11-Swipe between objects	
2.1	User is satisfied, presses the apply button in an object's detailed information view.	
2.2	User is not interested, repeat from step 1.	
3.1		The system opens a browser window of the correct page from the object's origin page.
3.2		The system opens a browser window of the login page of the object’s provider.

UC: 5-Navigate through main views

Summary: A user navigates through the main views by tapping the main navigation buttons.
Priority: high
Extends: 1-Run app
Participants: User
Normal flow of events: A user taps the navigation button corresponding to where he wants to go.

	Actor	System
1a	The user taps the search button in the navigation menu.	
1b	The user taps the favorites button in the navigation menu.	
1c	The user taps the search watcher button in the navigation menu.	
1d	The user taps the settings button in the navigation menu.	
2a		The system takes the user to the search view.
2b		The system takes the user to the favorites view.
2c		The system takes the user to the search watcher view.
2d		The system takes the user to the settings view.

UC: 6-Toggle favorite status

Summary: A user wants to change the favorite status of an accommodation, the user presses the favorite button or swipes in a direction on an accommodation in any of the list views.
Priority: high
Extends: 1-Run app, 3-Select Object, 11-Swipe between objects
Participants: User
Normal flow of events: The user taps the star icon and the system saves the object in the list of favorites in the favorites page.

	Actor	System
1	The user taps the star icon.	
2		The system change the color of star icon, showing the user that the favorite status has been changed.

Alternative flows:

The user swipes in the correct direction

	Actor	System
1	The user swipes the correct direction on an accommodation object in any of the list views	
2		The accommodation follows the swipe motion and background behind the swiped accommodation changes color.
3		The system change the color of star icon, showing the user that the favorite status has been changed.

The user swipes in the wrong direction

	Actor	System
1	The user swipes the wrong direction on an accommodation object in any of the list views	
2		The accommodation follows the swipe motion

UC: 7-Advanced Search

Summary: A user conducts an advanced search for housing using the advanced search menu
Priority: high
Extends: 1-Run app
Participants: User
Normal flow of events: A user conducts an advanced search for housing using the advanced search menu and the system displays the result.

	Actor	System
1	User presses the wrench button in the search view.	
2		System shows the advanced search view.
3	User fills in the advanced search form.	
4	User presses the search button.	
5		The system filters the database by using the user input and then displays the result in the search view.

Alternative flows

The user presses create search watcher button

	Actor	System
1	User presses the wrench button in the search view.	
2		System shows the advanced search view.
3	User fills in the advanced search form.	
4	User presses the create search watcher button.	
5		The name search watcher dialog is displayed
5.1.1	The user names the search watcher and presses the create button	
5.1.2		The search watcher is created and saved.
5.2	The user closes the dialog by pressing outside of it.	
6		The dialog is closed and the user returns to the advanced search.

UC: 8-Create a search watcher

Summary: A user creates a search watcher that gives the user notifications when a certain type of housing gets posted.
Priority: high
Includes: 5-Navigate through main views (c)
Participants: User
Normal flow of events: A user creates a search watcher by filling out the advanced search form and the pressing the save button. Which results in the system giving the user notifications when a certain type of housing gets posted.

	Actor	System
1a	User presses the wrench button in the search view.	
1b	5-Navigate through main views (c).	
2a		System shows the advanced search view.
2b	The user taps the plus-icon in the search watcher view.	
3a	User fills in the advanced search form and presses the save as search watcher button.	
3b	User fills in the form and presses the save button.	
4		The system displays a naming modal.
5	The user fills in the name of the search watcher and presses the done button.	
6		The system creates a search watcher which regularly checks the database for matches with the defined criteria and when a match is find the user gets a notification. (Even if the app is not opened.)

UC: 9-Edit a search watcher

Summary: The user wants to change a detail of a “bevakning” and does so under the “bevakning” view.
Priority: high
Inclouds: 5- Navigate through main views (c)
Participants: User
Normal flow of events: The user navigates to the “bevakning” view and taps the edit-”bevakning” button.

	Actor	System
1	5- Navigate through main views (c)	
2	The user taps the edit search watcher button on the search watcher object he wants to edit.	
3		The system opens up a modal where the user can edit all the information in the search watcher.
4	The user changes the information in the form to match what he wants and then presses the save button.	
5		The system correctly changes the information in the search watcher object to the new credentials. The modal is closed.

UC: 10-Get and access a search watcher notification

Summary: After a user has created a search watcher and a matching household contract gets put on sale the user receives a out-of-app push notification.
Priority: high
Includes: 1-Run app
Participants: User
Normal flow of events: A matching household contract to one of the user's active "bevakning":es gets put on sale and the system shows it to the user with a out-of-app push notification.

	Actor	System
1		The system periodically fetches data from the different databases for matches with all of the user's search watcherers. When a match is done the user gets an out-of-app push notification.
2		The system remembers the match and lists it in the search watcher view as a NEW match.
3	1-Run app (b)	
4		The system takes the user to the search watcher view with the correct search watcher list showing NEW matches.

UC: 11-Swipe between objects

Summary: The user selects an object and then swipes on the edges of the app when in a object information page and the system takes the user to the next searched object.
Priority: high
Includes: 3-Select Object
Participants: User
Normal flow of events: The user swipes on the edges of the app when in a object information page and the system takes the user to the next searched object.

	Actor	System
1	3-Select Object	
2	The user swipes on the edges of the app when in an object's detailed view.	
3.1		Swiping to the left takes the user to the detailed view of the next object in the search result. Swiping to the right takes the user to the detailed view of the previous object in the search result.
3.2		There are no more accommodations in the swiped direction, the user stays on the same view.

UC: 12-User runs the app, searches, browses/finds, and applies for an object

Summary: The user starts up the app because he wants to apply for a householding contract. He uses the search function to find a suitable contract and applies.
Priority: high
Includes: 1-Run app, 2-Basic Search, 4-Apply for contract, 7-Advanced Search
Participants: User
Normal flow of events: The user starts the app, conducts a search for householding contracts and then applies to one of them.

	Actor	System
1	1-Run app	
2a	2-Basic Search	
2b	7-Advanced Search	
3	4-Apply for contract	

UC: 13-User runs the app, navigates to the favorites view, and removes a favorite object

Summary: The user wants to remove an object from his lists of favorites and does so under the lists of favorites after he has opened the app and navigated to the correct page.
Priority: high
Includes: 1-Run app, 5-Navigate through main views (b), 6-Toggle favorite status
Participants: User
Normal flow of events: The user runs the app, navigates to the correct page and then removes the object from his list of favorites.

	Actor	System
1	1-Run app	
2	5-Navigate through main views (b)	
3	6-Toggle favorite status	

UC: 14-User navigates to the search watcher view and checks out a search watcher object

Summary: The user wants to check on a new search watcher match and does so.
Priority: Medium
Includes: 5-Navigate through main views (c)
Participants: User
Normal flow of events: The application is started, and the search watcher view is selected. The user then taps a search watcher.

	Actor	System
1	5-Navigate through main views (c)	
2		The system displays a number on the search watcher object where the numbers is how many new householding contracts that match the “bevakning” there are.
3	The user taps on the search watcher object of which he wants to check.	
4		The system slides open a panel under the search watcher object that shows the list of new objects that match the “bevakning” criteria.

UC: 15-Distance Search

Summary: The user wants to use a maximum distance from a certain point as a search criteria.
Priority: Low
Extends: 1-Run App
Participants: User
Normal flow of events: The user goes to the advanced search view, fills in a maximum distance, and then taps on a certain location on the map.

	Actor	System
1	User presses the wrench button in the search view.	
2		System shows the advanced search view.
3	The user enters a maximum distance in the maximum distance box.	
4	The user taps on the map where he wants the system to calculate the distance to.	
5	The user taps the search button.	
6		The system calculates the distance from every accommodation and the given point and compares it to the given maximum distance.
7		The system takes the user back to the search view.
8		The system displays the resulting list of accommodations in the search view.