Software Development Document

Chalmers On The Go – the Complete Chalmers Experience

This document describes the product vision, examples of user stories, requirements, design decisions, examples of tests carried out, change log, release history and end result of the development of the Android application ChalmersOnTheGo. Note that only examples of user stories, associated acceptance tests and unit tests are described in this document. For details, see document Test Report.

For ChalmersOnTheGo 1.0, Jelly Bean 4.1.

Fredrik Einarsson - Niklas Johansson - René Niendorf
Anders Nordin - Sofie Peters

Software Development Document

Table of Contents

1 Project description

The Android application ChalmersOnTheGo was designed, developed and released as the main task of the course Software Engineering Project, DAT255, on the Chalmers University of Technology, April-May, 2013. Android was the operating system of choice on the course. The development team consisted of Fredrik Einarsson, Niklas Johansson, René Niendorf, Anders Nordin and Sofie Peters. The tools used during the development process were the web-based hosting service Github using the Git revision control system, the agile development planning tool Pivotal Tracker and the IDE Eclipse.

2 Product vision

The main vision for the product was to create a map application over the Chalmers area, giving the user not only a regular map, but instead "the complete Chalmers experience". This meant that the user should not only be able to find locations, such as those found with addresses on Google Maps and similar applications. The user should instead be able to search inside Chalmers buildings for Chalmers specific names on locations, such as group rooms, lecture halls or pubs. Moreover, the application should contain some other features than the core map-features, features with connection to the "Chalmers experience". The application should be useful for the target group of the product: Chalmers students and people new to the Chalmers area. It should furthermore be intuitive and fun to use. It would have the edge over other map applications, since ChalmersOnTheGo would provide ways to find locations inside buildings, as well as specific Chalmers buildings with Chalmers specific names, not needing the exact address. An essential constraint to make the vision realistic, considering the development team's limited time resources, was that the actual location data would be limited to show the application's potential, not the complete Chalmers area.

2.1 Core features

- Searching and getting marked Chalmers specific locations
- Searching on Chalmers specific names, not addresses
- Getting the user's current location
- Get shortest path from user's current position to arbitrary location
- Expanded map to include shortcuts between and through buildings

2.2 Non-core features

- Section specific colouring of buildings when searching for pubs
- Logos for pubs and section buildings
- Night and day mode

3 Developing

In this chapter user stories, product backlog and requirements presented, as well as design decisions, test examples and release history, which includes both change log and the final result.

3.1 User stories

In the text below, a cross-section of the user stories used when developing the application are presented. For complete list, see document Test Report. The subscript numbers after most user stories are identical to the subscript number after the associated requirements in section 3.2 Product backlog.

General

- As a user, I want to be able to reverse my actions with a back-button[2]

Map

- As a user, I should only be able to see and navigate inside the Chalmers area[3]

Navigation

- As a user having gotten a location marked on the map, I want to be able to get the shortest path from my current location to the wanted location, by clicking the location's information window[6]

Searching

- As a user typing in a search for some item, I want a dropdown menu to appear with word-completed suggestions[8]
- As a user searching and getting suggestions, I want to be able to click any suggestion and get the location I clicked marked on the map[9]

Layer function

- As a user, I want to be able to have a checkbox-regulated layer function where I can choose between location types[13]
- As a user, I want to have layers with computer rooms, lecture halls, group rooms and pubs[15]

Different application modes

- As a user, I should be able to switch between night and day mode at any point or time in the application

Non-navigational features

- As a user, I want useful and fun features not only concerned with the map and navigation

Design

- As a user, I want to be able to reach all the application's functions from a menu system similar to Google maps[17]

Performance

- As a user, I want the application to perform wanted actions reasonably fast[22]

3.2 Product backlog

In the sections below, you will find the functional and non-functional requirements used when developing the application. Most of them are extracted from different user stories and have evolved during the development process. There is no internal prioritising in the listing. The subscript number after most requirements are identical to the subscript number after the associated user story in section 3.1 User stories and in the document Test Report. Requirements without a subscript number have no associated user story. At the end of the section, you will also find a burn-down chart of the development process, extracted from Pivotal Tracker.

3.2.1 Functional requirements

General

- The application will have an ”exit-button” which exits the application[1]
- The application will have a “back”-button, reversing performed steps, in accordance with typical Android design[2]

Map

- When clicking a marked location, a popup window will show up, containing the location’s name, floor and a button for navigating to the location[5]
- The application will have an “erase”-button, erasing all marked locations on the map, including layers[24]

Navigation

- The application will have a ”target”-button, targeting the current location[7]
- The popup window on a marked location will contain a button which draws out the shortest path (according to Google) from the current location to the wanted location[6]
- When the path to a certain location is drawn, the time and distance getting there from the user’s current location will be printed on the screen[29 & 30]
- The path between two separate locations will be drawn when searched for[31]
- Pressing and holding on the map will result in a location being marked[32]

Search

- The application will have a ”search”-button which triggers a search field[8]
- The search field will provide word-completed suggestions when writing letters in it[8]
- The suggestions will concern all the possible alternatives with the chosen letters[8]
- Specific rooms, buildings or pubs can be searched[8]
- When clicking a search item, it/they will be marked on the map in the visible view[9 & 11]
- More than one item can be searched for and marked on the map[9]
- When a room is clicked, the closest entry to the room will be marked on the map[10]
- Generic room types can be searched with “smart search” functionality[12]

Layers

- The application will have a ”layers”-button, showing checkbox options for the different possible layers[13]
- When checking one or more layers, those types of locations will show on the map[14]
- Layers can be unchecked[14]
- Possible layers will be group rooms, lecture halls, computer rooms, floors and pubs[15]

Database

- A database containing information about locations will be set up[8]
- The database will provide the application with information when needed[8]

- Complete data for buildings: EDIT, Vasa, Architecture and Student Union will be added
- Arbitrary data for food and spare time locations will be added; restaurant, billiard room, cinema, rooms with microwave, sauna, gym, ATM and student association room

Menu (external button)

- The phone's "menu"-button will trigger a menu showing other options than the core features
- The menu will contain the "exit"-button[1]
- The menu will contain an "activate StepCounter"- and "deactivate StepCounter"-button[25]
- The menu will contain a "Calorie drinking progress"-button[26]

Non-navigational extra features

- A step counter can be activated and deactivated on the same button[25]
- The step counter will count and show the user's steps when activated, until the user deactivates the step counter[25]
- The step counter will continue to count step even when the phone is in sleep mode or the application is minimised[25]
- The step counter will not save steps taken if the application is exited[25]
- A calorie drinking progress window, connected to the step counter, can be opened[26]
- The calorie drinking progress dialog will show progress bars for wine, beer, shots, cider and water[26]
- The calorie drinking progress dialog will show the user how many calories and steps worth of each drink she has burned[26]
- The calorie drinking progress dialog will contain "Drink it"-buttons, one per drink type, notifying the application that the user has drunk some drink, starting the progress bar for new drinks[27]
- The calorie drinking progress dialog will show the user how many drinks she has taken[26 & 27]
- A user pressing "Drink it" without having taken enough steps according to the progress bar, will trigger a message window, informing the user that she might become fat if continuing this way[27]
- The calorie progress dialog will save information for a minimised application or sleep mode[26 & 27]
- The calorie progress dialog will not save information if the application is exited[26 & 27]

3.2.2 Non-functional requirements

Appearance

- The application icon will have an appealing design[16]
- The overall appearance shall be appealing[21]
- When searching, the suggestions will show icons related to the suggested location type[18]
- Change colour scheme when changing between night and day mode

Design

- The menu system will be in accordance with typical Google maps design[17]
- The GUI shall be intuitive

Support

- If the GPS is not enabled, a window prompting this will show[20]
- The user's current position shall be targeted when opening the application inside the map boundaries[23]
- Switched orientation on the phone will not result in losing chosen data[28]

Reliability

- The application will not crash when used in the intended way

Performance

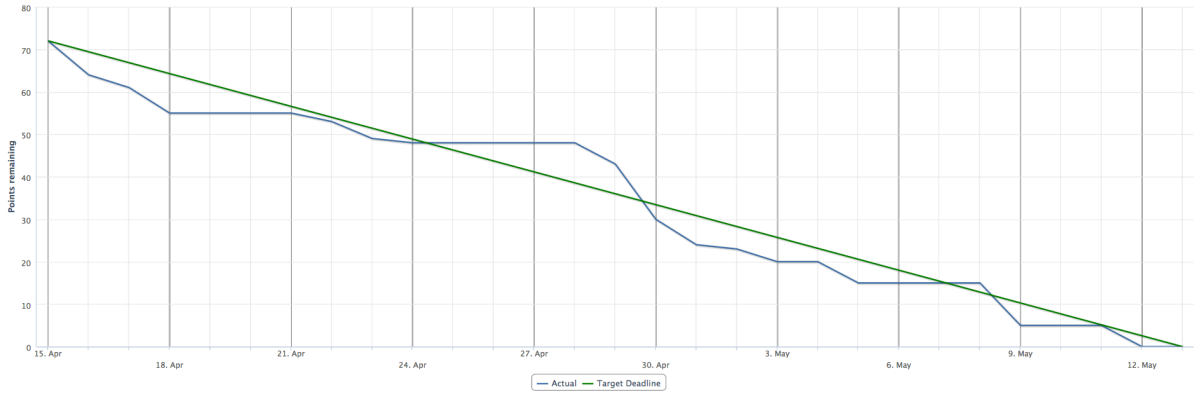
- Any chosen function should take 1 second or less to compute on a phone model not older than 1 year[22]

Constraints

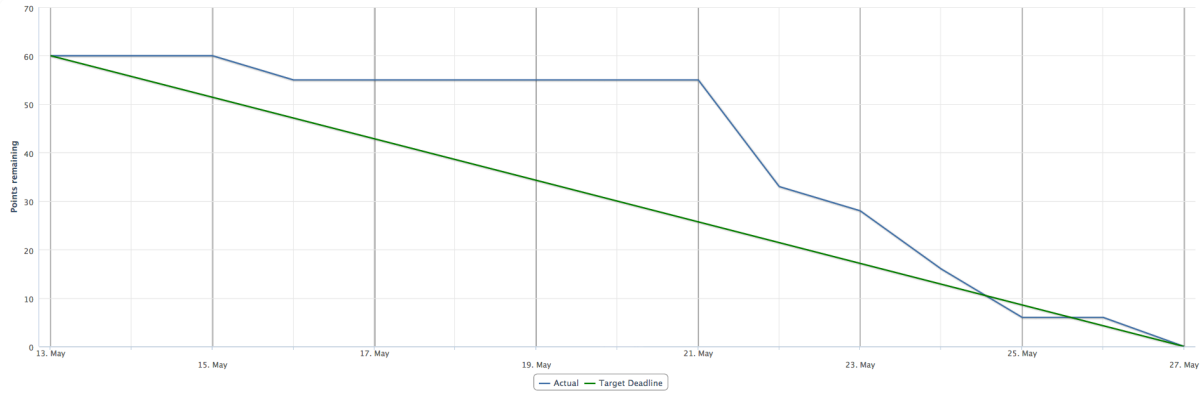
- The map will contain location data only for some parts of the Chalmers area
- The map will have boundaries, making it only possible to navigate inside the Chalmers area[3]
- The map can not be zoomed out of the Chalmers area[3]
- The map will have fixed coordinates in the starting view, while showing the user's current position[4]

3.2.3 Burn down chart

Version 0.1 (Beta)



Version 1.0 (Final)



3.3 Design decisions

The overall code structure of the application is created with the fundamentals of object-oriented programming patterns in mind, but also with respect to limited time resources and the Android framework. Below you will find design choices made by the team. Note that most choices correlate with the initial vision and that most removed functionalities were features which were discussed and researched, but not necessarily coded to any great extent.

External dependencies

The application ChalmersOnTheGo depends on the Google Maps Android API from the Google Play Services library. Since the application's core features concerns navigation on a map,

the Google Maps dependency is major. The alternative to let the development team write a map was not found to be a realistic alternative, not with respect to time resources, nor to the team's competence. An alternative to Google Maps discussed initially was the OpenStreetMap (http://www.openstreetmap.org/). However, the extensive support for Google Maps in Android software developing, along with it's well known GUI as well as some previous experience with the API in some team members settled the decision.

GUI
To make the application intuitive and easy to orientate in for the user, the GUI was based on typical Android and Google Maps designs. These designs were deemed as both well known and keeping a high intuitive standard. All core features were to be reached from the main view, whereas the non-core features were to be reached via the "menu"-button.

Double functionality
A user who, for an example, searches and marks the suggestion for "computer rooms" can receive the same result when marking "computer rooms" as a layer option. However, both features are intuitive: a user should be able to search for anything in the search bar *as well as* checking any preferred layers. The layer function just provides a simplified way to do a specific generic search, a shortcut if you will. The team want the user to choose how to use the application, not the other way around. In addition, the layer function is a common map feature which might be missed if not implemented.

Considered or removed functionality
Due to limited time resources, the team's final release only features data for some buildings, resulting in limited navigational possibilities. The product however provides a complete framework for "the complete Chalmers experience", with the possibility for future developers to easily add location data, consequently expanding the navigational possibilities. This limitation was planned from an early stage.

An expansion of the existing map with shortcuts between and through buildings was initially considered and researched. This was however found to be impossible to implement because of certain limitations in the API; the tool used for this type of development was not yet available in Sweden.

Roles were briefly considered initially, such as "nolla", "IT-student", "student with 180 p" etcetera, but also initially deemed both difficult to implement and to motivate with user value. The greatest user value would have been the ability for the application to plot different shortcuts through buildings where the current user had access. This was however not deemed as valuable enough for such a heavy implementation.

A night mode/day mode functionality with mode associated activities and colour schemes was a planned feature. It was found to be superfluous however since the layer functionality could just as easily entail the intended night mode functions as the mode itself. In this way, the user would experience a more compact and intuitive GUI. Furthermore, the feature of buildings being coloured in the section's colour was implemented and later removed since it was not found to add much value, especially with the closest entry already being marked on the map.

The possibility for a user to synchronise her schedule from Time Edit with the application was explored. The user should be able to see where her respective lectures or meetings were, and also be prompted to run if she risked being late to the appointed time. The functionality was however found to be both difficult to implement, as well as superfluous, since there already existed

worthwhile ways to look up scheduled places and times. The lookup could therefore be done elsewhere, and the ChalmersOnTheGo could be used purely for navigation, which is it's core feature.

3.4 Testing

In this section you will find samples of performed acceptance and unit tests, along with their associated user stories. For a complete test report, see document Test Report.docx.

3.4.1 Acceptance tests, sample

What is tested	Map boundaries
How it is tested	Repeatedly scrolling outside of the map.
Expected result	The map should get stuck on the boundaries.
Actual result	Expected result
Associated user story	As a user, I should only be able to see and navigate inside the Chalmers area[3]

What is tested	Activate/Deactivate message
How it is tested	The StepCounter will be activated, arbitrary functions carried out, then the StepCounter deactivation window will be visually controlled
Expected result	The StepCounter option should show "Deactivate StepCounter"
Actual result	The StepCounter option shows "Activate StepCounter" even though it is activated if phone orientation is changed
Fix	Activate/Deactivate message saved as variable and fetched when orientation changed
New result	Expected result
Associated user story	As a user, I want to se how many steps I have taken[25]

What is tested	Counting steps in sleep mode
How it is tested	The StepCounter will be activated and sleep mode will be engaged. The user will walk and manually count her steps, then waking up the application, comparing the counted steps to those in the StepCounter.
Expected result	The number of steps manually counted and those counted by the StepCounter should correlate.
Actual result	Expected result
Associated user story	As a user, I want to se how many steps I have taken[25]

3.4.2 Unit tests, sample

- **Insertion and getting in table 1 (coordinates and buildings table), test suite**
  - insertIntoTable1 and getClosestEntry were tested together. Connected to user story "As a user, I want to be able to search for a room, mark it and get the closest entry to the room marked on the map":
    - A pair of coordinates (Double) and a building name (String) were inserted into table 1 via insertIntoTable1.
    - The coordinates (Double) were used to create an object (LatLng) containing latitude and longitude.
    - The object (LatLng) and the building name (String) served as input in getClosestEntry.

        - The result of getClosestEntry (LatLng) and the object (LatLng) containing the coordinates were compared and found to be equal.
        - Calculating the closest entry
    - insertIntoTable2, insertIntoTable3, insertIntoTable4 and getAllRoomsInBuilding were tested together. Connected to user story "As a user typing in a search for some item, I want a dropdown menu to appear with word-completed suggestions":
        - A room type (String) was inserted into table 2 via insertIntoTable2.
        - A false room type (String) was inserted the same way.
        - A building name (String) was inserted into table 4 via insertIntoTable4
        - Room name1 (String), a coordinate pair1 (Double), the true room type (String), the building name (String) and floor1 (String) were inserted into table 3 via insertIntoTable3.
        - Room name2 (String), coordinate pair1 (Double), the true room type (String), the building name (String) and floor2 (String) were inserted into table 3 via insertIntoTable3.
        - A false room name (String), coordinate pair1 (Double), the false room type (String), the building name (String) and floor1 (String) were inserted into table 3 via insertIntoTable3.
        - The building name (String) served as input in getAllRoomsInBuilding.
        - The result of getAllRoomsInBuilding (ArrayList<String>) was tested using methods size and contains, and found to be satisfactory.
        - Getting suggestions
    - insertIntoTable3 and getRoomCoordinates were tested together. Connected to user story "As a user, I want to be able to search for a building and get all the rooms in the building marked on the map":
        - A pair of coordinates (Double) were used to create an object (LatLng) containing latitude and longitude.
        - A room type (String) was inserted into table 2 via insertIntoTable2.
        - A building name (String) was inserted into table 4 via insertIntoTable4.
        - The room name (String), the coordinates (Double, the room type (String), the building name (String) and a floor (String) were inserted into table 3 via insertIntoTable3.
        - The room name (String) served as input in getRoomCoordinates.
        - The result of getRoomCoordinates (LatLng) and the object (LatLng) containing the coordinates were compared and found to be equal.
        - Getting all rooms in a specific building

3.5 Release history (Change log)

Since the application has had only two releases, this section serves both as release history and change log.

3.5.1 Pre release, version 0.1

Features

- Search with Chalmers specific names on locations
- Have Chalmers specific locations marked on the map
- Clicking a marked location results in a popup window, informing the user about name and potential floor of the marked location
- When searching, get word-completed suggestions over all the data in the database
- When searching, get Smart Search suggestions on location types, such as pubs, group rooms, lecture halls etc

- Layer function for computer rooms, lecture halls and group rooms
- Basic Google Maps features such as zoom in and out on map, dragging etc
- Tageting user's current position inside map
- On click switch between Night mode and Day mode, changing the colour scheme of the map

Non-core features
- Back-button
- Search suggestions shows logos or icons for pubs and some rooms

3.5.2 Final release, version 1.0

Features
- Search with Chalmers specific names on locations
- Have Chalmers specific locations marked on the map
- Press and hold on the map for a new position to be marked, making it possible to navigate to the marked position
- Show shortest path from some location to a wanted location or closest entry to the wanted location
- Clicking a marked location results in a popup window, informing the user about name and potential floor of the marked location
- Clicking the popup window results in a route navigation window appearing, where the user can navigate to the marked location, from her current position, or some other location
- When searching on the Search button, get word-completed suggestions over all the data in the database
- When searching on the Search button, get Smart Search suggestions on location types, such as pubs, group rooms, lecture halls etc
- Layer function for computer rooms, lecture halls, group rooms and pubs
- Floor options to show the room layers on all or on specific floors
- Basic Google Maps features such as zoom in and out on map, dragging etc
- Tageting user's current position inside map
- Intuitive GUI

Non-core features
- Menu option for exiting the application
- Menu option for emptying the map
- Menu option for activating or deactivating step counter
- Menu option for showing calorie drinking progress, connected to the step counter
- The application notifying the user of how many calories counted as steps the user has burnt by walking, in relation to drinks: beer, shot, wine, cider and water
- The user notifying the application of how many drinks the user has had, and what kind of drinks these were
- The application notifying the user about risks of becoming sick by drinking to much and similar
- Back-button
- Search suggestions shows logos or icons for pubs and some rooms

Bug fixes

- All content on map as well as open menus completely recreated when switching orientation
- Map lagging when starting up
- Project properties showing error
- Distance and duration still showing after emptying of map
- "My location" working, but inaccurate
- Exiting application not working

3.5.3 Resulting product, comment

The product contains both core features and non-core features, all of which contributes to "the complete Chalmers experience" as the development team interpreted the slogan. The data in the application's database is believed to show the possibilities and potential of the product.