Software requirements specification

Sport@Unibe - Team 2

Project: Sport@Unibe

Customer: Andrea

Group: 2

Team 2: Michael Scheurer, Thomas Steinmann, Karan Sethi, Lea Wollensack

Table Of Content

1. Introduction

Purpose

The purpose of this application is that the users can get information about sport activities at the University of Berne easily through an Android gadget. This application should be very easy to use so the users have no problem to get what they are looking for. There will be many features for them in this application.

This indicates that the users can see the list of different sports with schedules in one go. If necessary they can make a separate list with their own favourite activities and schedules. This application should also be able to offer them the location and direction of the selected sport. They can have an option to search certain activities with a specific criteria. Users should be able to make a reminder or even subscribe to a chosen sport at a specific time and automate the sign in process. Every sport can be rated by them and they should also be able to see the ratings of other people. The users should also know and be notified, if their friends are participating in the same sport.

Stakeholders

In this project our internal stakeholder is the customer, Andrea Caracciolo, who has given us an assignment to program an Android application of the Univerisity Sports Berne website.

Also we, the developers are internal stakeholders, as we want to create a successful product which is used by as many users as possible.
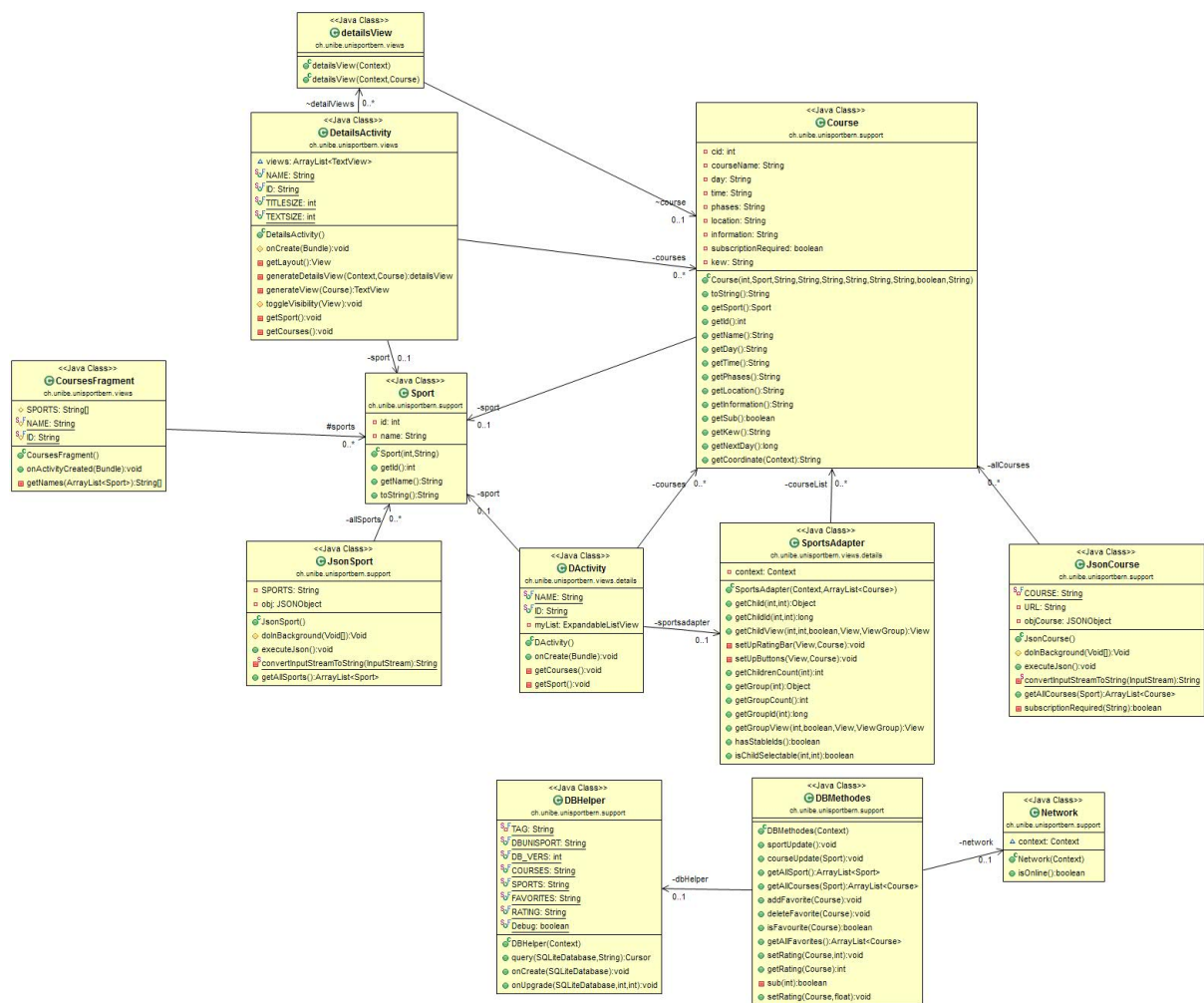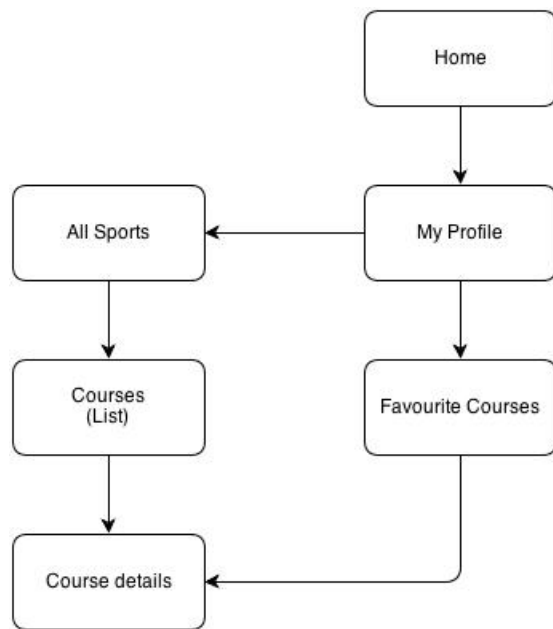
We also have external stakeholders like the students, who are the users of this application.They do not have anything to do with the project directly, but can affect its actions or can be affected by it.

Definitions

User: Students

Customer: Andrea Caracciolo

System overview

References

- Ian Sommerville, Software Engineering 2011
- Slides of the lecture "Introduction Software Engineering" Universität Bern 2013   5

2. Overall description

Use Cases

6

The following use cases are defined:
- see all sports
- show all courses of a sport
- show details of a certain course
- add a course to Favourites
- see all favourite courses
- show place where a course is hold
- rate courses
- see rating of courses

See all sports

Actors	User
Description	As a user I want to display a list of all available sports
Trigger	User clicks „Sports" tab
Pre-conditions	All sports are stored in the database
Post-conditions	A list of all sports appears
Main scenario	User opens app User clicks „Sports" tab
Alternative scenarios	There is no data in the database List in "Sports" tab is empty

Show all courses of a sport

Actors	User
Description	As a user I want to display all the courses of one sport.
Trigger	User clicks on a sport
Pre-conditions	All sports and all courses are stored in the database
Post-conditions	A list of all courses of this sport appears.
Main scenario	User navigates to "Sports". User chooses a sport by clicking on it Courses of that sport appear
Alternative scenarios	There are no courses stored in the database When the user clicks on a sport, the list of courses of this sport is empty.

Show details of a certain course

Actors	User
Description	As a user I want to display the details of a course by clicking on it. These include Time, Place, Rating and some other information depending on the type of course.
Trigger	User clicks on a course (in the „Sports" tab or in the „Favourite" section)
Pre-conditions	At least one course is displayed in the current view
Post-conditions	A standardized view appears, showing all relevant information on the selected course
Main scenario	User navigates to either „Sports" or the „Favourite" section in his profile. User chooses a course by clicking on it Details appear
Alternative scenarios	There are no courses stored in the database When the user clicks on a sport, the list of courses of this sport is empty.

Add a course to Favourites

Actors	User
Description	As a user I want to be able to add a course which I like to my Favourites.
Trigger	User presses the "Favourite" icon (star)
Pre-conditions	User is displaying the details of a course at the moment
Post-conditions	The selected course is added to the user's favourite courses, meaning it appears in the list seen under the "Favourites" section in the Profile tab.
Main scenario	User navigates to the courses of a sport User chooses a course by clicking on it Details appear expanded. User clicks on the „Favourite" icon Course is now stored in "Favourites"
Alternative scenarios	Course is already stored in "Favourites" The "Favourite" icon (star) is yellow When the user clicks on the "Favourite" icon, he unfavours the course. The course is no longer stored in "Favourites".

See all favourite courses

Actors	User
Description	As a user I want to display a list of all my favourite courses
Trigger	User clicks "Profile" tab
Pre-conditions	User must have at least one favourite course
Post-conditions	A list of all favourite courses is found under the "Favourite" section in the "Profile" tab
Main scenario	User opens app „Profile" is the starting tab
Alternative scenarios	User has no favourite courses: under the "Favourite" section there is no information shown.

Show place where the course is hold

Actors	User
Description	As a user I want to display the place where a course is hold
Trigger	User clicks the map in the details view
Pre-conditions	User is connected to the internet
Post-conditions	Google map is opened and shows the address where the course takes place
Main scenario	User opens app User navigates to the details view of the course he wants to know the place of User clicks on the map in the details view The address is shown in google maps
Alternative scenarios	User is not connected to the internet Error message appears: no internet connection

Rate courses

Actors	User
Description	As a user I want to rate a course
Trigger	User clicks on a rating scale in the details view of a course
Pre-conditions	User is displaying the details of a course at the moment
Post-conditions	The selected course got a rating.
Main scenario	User opens app User navigates to the desired course's details view User clicks on a rating bar, representing his overall impression on the course A rating view opens, where the user can fill in additional informatio and comments on the course User fills in the form and clicks the „Confirm" button to send the feedback The detail view of the course returns
Alternative scenarios	The user already rated the course. When the user rates the course again, the initial rating will be replaced by the new rating.

See rating of courses

Team 2: Michael Scheurer, Thomas Steinmann, Karan Sethi, Lea Wollensack

Actors	User
Description	As a user I want to see the rating of a certain course
Trigger	User clicks on a certain course
Pre-conditions	There is at least one rating for the selected course stored in the database.
Post-conditions	The details of the selected course are expanded
Main scenario	User opens app User clicks „Sports" tab User chooses a sport User clicks on a certain course of that sport The details view where the rating is incuded expands
Alternative scenarios	There are no ratings for this specific course. The rating of this shown course is indicated by 0.

Actor characteristics

Users of the Sport@Unibe App are typically students at the University of Berne. Since there are eight different faculties and many different programs of study the characteristics of someone using the App turn out to be quite varied. As an example, let us point out three possible users:

Andreas, 20, studies Psychology in the first semester. He's not a very sporty person, but he used to enjoy the sports classes when he was still in the Gymnasium. Hearing about the wide sport offered by the University he decides to check out some courses. He downloads the App and searches for team sports and their timetables. There are 23 results. Andreas sets a reminder for the football course on Wednesday noon.

Tania, 24, is a Sports student in her masters course. Sport is not only her field of studies but also her passion and she exercises regularly. She is looking for a possibility to coordinate her timetable. Thus she downloads the App, marks her favourite sports and checks on their timetables. Now she can compare her course schedule with the sports schedules.

Cathrine, 22, studies Physics. She likes doing sports but usually she is too lazy to go to the sports courses offered by the University. That is why she makes an agreement with her fellow student Patrick to do at least one sports course a week. They both download the App and add each other as friends. They do not decide on a specific course but mark the courses they are planning to visit. Through the App they see the information who is planning to go to which course and are able to decide on a course they visit together.

3. Specific requirements

A functional requirement describes what a software system should do, while non-functional requirements place constraints on how the system will do so.

Functional requirements

- A list of all sports in the campus should be shown.
- The app should enable the user to see course schedules for a selected sport.
- The user should be able to define his favorite courses and see their schedules.
- It must be possible to see the location of a selected course and get directions to get to this place.
- The user should be able to see a list of courses that match with a certain criteria e.g start at a given day/time i.e. it should be possible to search the list of all courses with a certain filter.
- The user should get notifications for upcoming courses e.g. 2h before the course starts.
- There should be a feature in which the user can rate attended courses.
- The user should see how other people have rated a certain course
- The user should be able to let his friends know to which course he has planned to go. If they want to join him, the user should be notified.

Non-functional requirements (external, performance, etc.)

Product requirements:

The app should support all versions of Android between Android 4.0 (Ice Cream Sandwich) and Android 4.2 (Jelly Bean).

Usability: The app should be designed in a way that it is easy understandable and fast to learn for the user on how to operate the app. An attractive design should be sought.

Emotional factors: It should be fun to use the app and the app should encourage the user to do more sports at the University of Berne.

Accessibility: The app should be accessible for all members of the University of Berne.

Backup: There won't be any backup provided. The user will download the app and use it locally on his Android device. If he deletes the app there will not be a possibility to restore the collected data.

Documentation: The SRS (Software requirements specification) will be kept up-to-date throughout the whole design process.

Extensibility: This app is made during the "Introduction to Software Engineering" class in the fall semester 2013 at the University of Berne. After that, future growth is not intended.

Organisational requirements:

Development environment: How proposed in the lecture, we'll work with Eclipse and SDK (Android)

Used programming language therefore is java.

Operational system: Android devices

External requirements:

Safety: In case the user has to provide personal data, i.e. a log in or something similar is implemented, the user should be able to inspect his personal data and correct errors in the data / change the data.