Software Requirements Specifications

Version	Date	Revision Description
0.01	29.09.2013	Collected information after first contact with the customer.
0.02	02.10.2013	Feedback information about SRS (Use Cases, Diagrams, etc)
0.03	09.10.2013	Prototype review and planning (Trello addition, iteration1)
0.03	16.10.2013	User stories review and implementation (iteration2)
1.0	06.11.2013	Version 1
2.0	11.12.2013	Version 2

Table 1: Revision History

1. Introduction

The University of Bern offers a variety of sports for their students. Students registered regularly at Unibern can benefit from free entry at the University's numerous sporting facilities or from joining the sport courses. Sport places are distributed throughout the campus and have a schedule time. Courses have different levels and meet at a scheduled time and location. To attend a course, students sometimes need to register by a certain date.

1.1. Purpose

As university students, we want to serve our peers by spreading important sports-related information and the latest news regarding the Unibe sports program. Students should be able to find all the necessary information about sports so they can plan their training period without losing any opportunity.

The idea of this project is to build an application that allows students to be better organized for their sport hobbies. We want students to be informed of our sports and we want to have a large number attending.

1.2. Stakeholders

Stakeholders are anyone who has an interest or that participates in the project. Until now, the people that are actively involved in the project, or that its completion or execution may affect their interest are:

- Customer: Andrea Caracciolo

- Technical tutor: Bledar Aga
- Team members:
  - Noemi Wälle
  - Aliaksei Syrel
  - Stefan Borer
  - Antonio Terán Espinoza

1.3. System overview

1.4. References

Information regarding sports can be retrieved from the following data source:

- http://www.sport.unibe.ch
- http://scg.unibe.ch/ese/unisport/sports.php
- http://scg.unibe.ch/ese/unisport/sport.php?id=[ID]
- http://scg.unibe.ch/ese/unisport/location.php?loc=[STRING]
-

2. Overall description

This section is used to provide a more thorough description of the available functions of the system. Each function is represented by its own use case, accounting for an action inside of the Sport@Unibe application system. The Use Case diagram available in Figure 1 allows us to see the involved parties in the different use cases there are, graphically summarizing the structure of the application. By following the different paths on the diagram, one may create different scenarios for each possible variation of flow through the use case.

Aside from the use case diagram, a navigation prototype of the application is provided in Figure 2. It is useful to have a sense of the application's

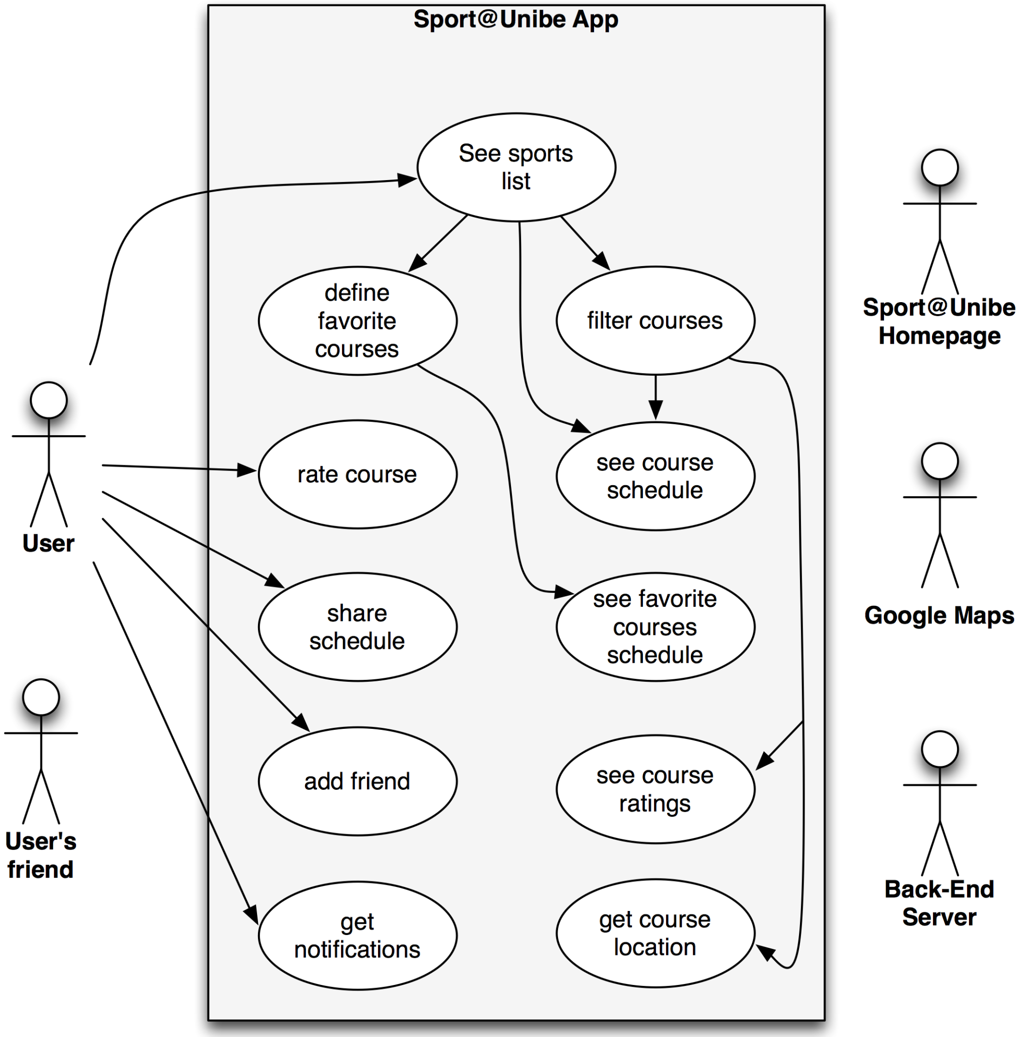

Figure 1: Use Cases Diagram.

architecture, since it allows the user to know exactly the different paths that lead you to different parts of the application, making it clear how to access or leave them utilizing the directional arrows that point in the sense of the application's flow.

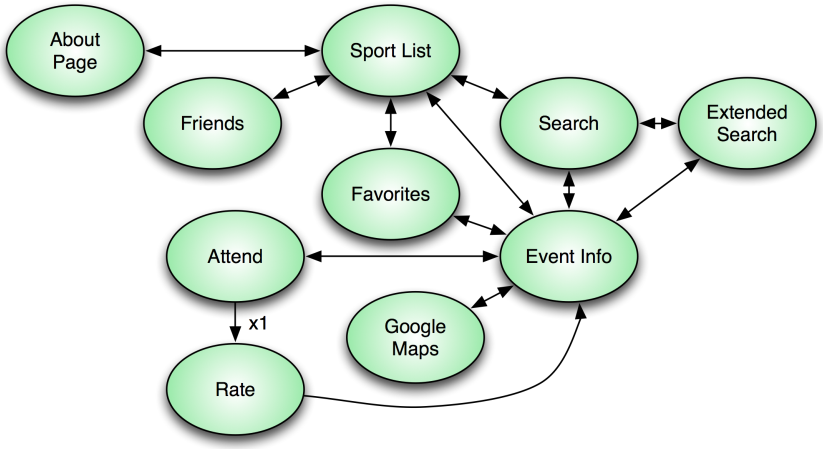

Figure 2: Navigation of the applicacion.

2.1. Use cases

0. **Searching for sport events**

0.1 **Actors**
Sport@Unibe application user.

0.2 **Description**
As a user I want to see a list of courses that match a certain criteria.

0.3 **Trigger**
User launches Sport@Unibe app or clicks on the search option.

0.4 **Pre-conditions**
1.- Customer has an Android device with at least version 2.1 on it he is using any Android emulation software.
2.- Customer has downloaded Sport@Unibe application and can run it

0.5 **Post-conditions**
1.- Customer can see and browse all information he was looking for.
2.- Customer wants to be able to access other use cases based on located information.

0.6 **Main Scenario**

1.- Customer opens application.
2.- Customer enters the searching area.
3.- Application prompts customer to enter some optional search criteria.
4.- Application validates if there exists data which matches the criteria.
5.- Application lists all found data on display.
6.- Customer can browse all located data.
7.- Customer returns to main applications or exits as he has received all necessary information.

0.7 Alternative Scenarios

Application did not found any data matching search criteria:
1.- Application will prompt customer to change search criteria.
2.- Customer makes some changes.
3.- Use Case resumes on step 5.

0.8 Special Requirements

Searching progress should not take too long and search criteria should be easy to enter and logicaly placed.

0.9 Notes

Searching area should be easy to find in main application and fast to run through for all users.

1. See sports list

1.1 Actors

Sport@Unibe application user.

1.2 Description

As a user I want to see the list of all available sports.

1.3 Trigger

User launches Sport@Unibe app or clicks on sports list tab.

1.4 Pre-conditions

1.- User has an Android device with at least version 2.1.
2.- User has downloaded and installed Sport@Unibe app.

1.5 Post-conditions

1.- User is able to see the sports list.

1.6 **Main Scenario**

1.- User launches app.
2.- User navigates to sports list tab.
3.- System fetches list from the internet/internal cache.
4.- User sees the available sports list.

1.7 **Alternative Scenarios**

a) Application crashes:

1.- System will return to dashboard.
2.- Use Case resumes on step 1 of normal flow.

b) First time using the application with internet connection available:

1.- System fetches list from the internet database.
2.- System stores the list in user's device.
3.- Use Case resumes on step 4 of normal flow.

c) First time using the application without internet connection available:

1.- System cannot fetch the list from the internet database.
2.- Action is terminated and the system prompts an error.

1.8 **Special Requirements**

The user's device should have available storage space for the sports database and should be able to comply with the minimum technical specifications. If the list is not found on the device's cache, internet connection should be available in order to download it from the web.

1.9 **Notes**

The sports list will be stored in the device's internal memory as cache, so that it would be available without any internet connection.

2. See course schedule

2.1 **Actors**

Application should have public access. Actors might be students at University of Bern, locals who want to use some offered sport courses or other interested persons who want to get information about Universitätssport Bern in general or different sports offered by them.

2.2 **Description**

As a customer I want to know which days and at what time a sport takes place. I want to be able to see a list of all courses and information about day, time and place I can join them.

2.3 **Trigger**

A customer wants to know at which days and at what time a sport he wants to join takes place.

2.4 **Pre-conditions**

1.- Customer has an Android device with at least version 2.1 on it he is using any Android emulation software.

2.- Customer has downloaded Sport@Unibe application and can run it.

3.- Customer needs internet access from time to time to update course information from Universitätssport Bern Homepage.

2.5 **Post-conditions**

1.- Customer can see and browse all available courses.

2.- Customer wants to be able to access other use cases based on schedule list.

2.6 **Main Scenario**

1.- Customer opens application.

2.- Customer enters the sports overview list and selects a specific sport.

3.- Application enters into new view with schedule list.

4.- Customer can see all courses in a list with day, time and place row.

5.- Customer can choose one course and in a fast option bar he can show place of course on map or add a specific course to his favorites.

6.- Customer returns to main applications, to the sport overview list or exits as he has received all necessary information.

2.7 **Alternative Scenarios**

a) Customer enters the schedule list via his favorite list:

1.- Continues at step 3.

b) Customer pushes show on map bottom in fast option bar:

1.- Application opens new view and shows place on Google maps.

2.- As user has finished locating the courses place on map he can return to step 5.

c) Customer pushes add to favorites bottom in fast option bar:
1.- Application adds the course to users favorites.
2.- User can continue at step 5.

2.8 **Special Requirements**
Schedule list should be easy to read. User needs internet access to have access on Google maps information.

2.9 **Notes**
Entering schedule list is easy to access via pushing an element in sport list overview.

3. Define favorite courses

3.1 **Actors**
Sport@Unibe application user.

3.2 **Description**
As a user I want to set my favorite courses.

3.3 **Trigger**
User click on the ”star” to set a course as a favorite.

3.4 **Pre-conditions**
1.- User has an Android device with at least version 2.1.
2.- User has downloaded and installed Sport@Unibe app.
3.- User has already has the sport list.

3.5 **Post-conditions**
1.- Selected course is available in the favorites list.

3.6 **Main Scenario**
1.- User launches app.
2.- User navigates through the different courses.
3.- User ’stars’ the desired course.
4.- System adds the course to the favorites list.
5.- User is able to see the selected course in the favorite list.

3.7 **Alternative Scenarios**
a) Course already is a favorite:
1.- As the user touched the ’star’ button, system removes the starred status of the course.
2.- System removes the selected course from favorites list.

3.8 **Special Requirements**
None.

3.9 **Notes**
The user is able to add whichever number of courses to the favorite list.

4. See favorite courses schedule

4.1 **Actors**
Application should have public access. Actors might be students at University of Bern, locals who want to use some offered sport courses or other interested persons who want to get information about Universitätssport Bern in general or different sports offered by them.

4.2 **Description**
As a customer I want to be able to see timetables of my favorite courses.

4.3 **Trigger**
A customer has favorites curses added to his favorite list and wants to have an overview of them including day and time they will take place.

4.4 **Pre-conditions**
1.- Customer has an Android device with at least version 2.1 on it he is using any Android emulation software.
2.- Customer has downloaded Sport@Unibe application and can run it.
3.- Customer has already added favorites to his favorite list.

4.5 **Post-conditions**
Customer can see and browse all his favorites in one list with day and time they will take place.

4.6 **Main Scenario**
1.- Customer opens application.
2.- Customer enters the searching area.
3.- Customer can slide to Favorites page.
4.- Application lists all courses added up to now.
5.- Application day and daytime to each course.

6.- Application offers fast bottom menu to remove a curse from favorites or to show curse place on map.
7.- Customer returns to main applications or exits as he has received all necessary information.

4.7 **Alternative Scenarios**
a) User decides to remove a curse via fast bottom menu from favorites list:
1.- User push remove bottom.
2.- Application deletes selected curse from list.
3.- Application returns to favorite List Page and scenario continues at step 13.
b) User decides to show a favorite curse on map:
1.- User push show on map bottom.
2.- Application opens new view with curse location on Google maps.

4.8 **Special Requirements**
None.

4.9 **Notes**
For task **b)**, Device or Emulation needs internet connection to access Google maps. If he already opened the map once, it might be saved in cache and internet connection is not necessary.

5. **<u>Get course location</u>**

5.1 **Actors**
User of Sport@Unibe application.

5.2 **Description**
As a user I want to find a location of selected course.

5.3 **Trigger**
User opens course description.

5.4 **Pre-conditions**
User should have internet connection.

5.5 **Post-conditions**
1.- User sees location of course.
2.- User found place location on map.

5.6 **Main Scenario**
1.- User opens sports list.

2.- User selects sport.
3.- User sees list of courses.
4.- In table user finds place column.
5.- In place column user sees location.

5.7 Alternative Scenarios

a) User clicks on course:
1.- User sees location in course info.

b) User long clicks on course:
1.- User sees opened extra panel. 2.- User clicks ”locationïcon. 3.- User sees location on map.

5.8 Special Requirements

User needs internet to get course data and list of sports.

5.9 Notes

For Google Maps API OpenGL ES 2nd version is necessary.

6. Filter courses

6.1 Actors

Customer.

6.2 Description

As a user I want to see a list of courses that match with a certain criteria e.g. start at a given day/time.

6.3 Trigger

Customer enters criteria in the search field and hits the search button.

6.4 Pre-conditions

1.- Customer has a running Sport@Unibern - App

6.5 Post-conditions

1.- Customer sees a new window.
2.- Window shows results for the costumers search.
3.- Customer sees an error dialogue if his searching criteria was invalid.

6.6 Main Scenario

1.- Customer opens Sport@Unibern – App on Android Phone.
2.- Customer clicks on “Search” – Field.
3.- Customer enters search criteria.

4.- Customer hits "Search" – Button.
5.- Customer sees a sign indicating that the search is being processed.
6.- Customer sees search result.

6.7 **Alternative Scenarios**

a) Customer sees error dialogue:
1.- App provides other possible search criteria ("You wanted the enter: . . . ?")

b) Customer sees error dialogue:
1.- Could not process search.

6.8 **Special Requirements**

Search processing should not take more than 2 seconds.

6.9 **Notes**

1.- Should there be separate search options for keywords and for dates/times (calendar, clock)?
2.- If 1 is accepted, should there be a possibility to combine search criteria like keywords + dates?
3.- Must internet connection be established for searching?
4.- Should searching for a specific person be provided (stalking issues).

7. Get notifications

7.1 **Actors**

Customer.

7.2 **Description**

As a user I want to have notifications for upcoming course e.g. 2h before the course starts.

7.3 **Trigger**

This event is triggered by the App itself.

7.4 **Pre-conditions**

1.- Customer has a running Sport@Unibern – App.
2.- Customer has marked a course as favorite.
3.- It's only 2h left 'till course starts or an employee of the university has added a notification.

7.5 **Post-conditions**

1.- Customer receives a push notification.

7.6 Main Scenario

1.- Customer has Sport@Unibern – App running in background.
2.- Notification appears.

7.7 Alternative Scenarios

None.

7.8 Special Requirements

None.

7.9 Notes

None.

8. Rate course

8.1 Actors

Customer.

8.2 Description

As a user I want to rate an attended course.

8.3 Trigger

Customer clicks on “Rate this course” - button.

8.4 Pre-conditions

1.- Customer has a running Sport@Unibern – App.
2.- Device is connected to internet.

8.5 Post-conditions

1.- Dialogue shows what the customer rated.
2.- Rating of the course changes.

8.6 Main Scenario

1.- Customer opens Sport@Unibern – App.
2.- Customer selects the course.
3.- Customer clicks on “Rate this course” – button.
4.- Customer rates the course.
5.- Customer hits “OK” or “Enter”.
6.- Dialogue shows the customer’s rating.
7.- The previous window appears.

8.7 Alternative Scenarios

None.

8.8 Special Requirements

Internet Connection.

8.9 Notes

1.- Is internet connection necessary or can the app save the rating until next sync?
2.- Is the rating with marks (1 – 6 maybe) or/and with words?

9. See course ratings

9.1 Actors

Customer.

9.2 Description

As a customer I want to see how other people who have already been at a curse would rate it. It helps me to decide whether to join a curse or not and gives some kind of feedback for teachers who guide certain curses.

9.3 Trigger

A customer wants to rate a curse and wants to be able to see how other users rated the curse.

9.4 Pre-conditions

1.- Customer has an Android device with at least version 2.1 on it he is using any Android emulation software.
2.- Customer has downloaded Sport@Unibe application and can run it.
3.- Application needs to update from time to time to have actual statistics about rating ready.

9.5 Post-conditions

4.- Customer can see if a curse is rated bad or good on a scale.
5.- New rated curses from a user is updated to the backend server of the Application.

9.6 Main Scenario

1.- Customer opens application.
2.- Customer enters the searching area.
3.- Customer can see rating of a curse via fast bottom panel.
4.- If customer wants to rate a curse he push on rate bottom.
5.- Application shows a scale where user can place his rate from bad to good.
6.- Application saves the new rate value on backend server.
7.- Application updates.

8.- Application returns to the previous screen.
9.- Customer can now see an actual rate of the previous rated curse.
10.- Customer returns to main applications, makes other rates or exits as he has received all necessary information.

9.7 Alternative Scenarios

a) User is going to rate a curse a second time:
1.- User clicks rate bottom of a curse he did already rate before.
2.- Application shows warning if user really wants to rate again.
3.- User decides if he wants to rate again or not.
4.- Scenario continues at step 23.

9.8 Special Requirements

To make rates user need internet connection, if he only wants to see rates he only needs it to get the very latest updates, otherwise it's enough to use Application in offline mode.

9.9 Notes

We need to talk with our customer Universitätssport Bern if he wants to rate from bad to good or if he wants to be able to get different points rated. Like: Teacher, Sport itsef...

10. <u>Share schedule</u>

10.1 Actors

Sport@Unibe app user and friends (also users).

10.2 Description

As a user I want to send my courses schedule to my friends, in order for them to join me if they want.

10.3 Trigger

User clicks on the 'share schedule' option.

10.4 Pre-conditions

1.- User has an Android device with at least version 2.1.
2.- User has downloaded and installed Sport@Unibe app.
3.- User has already has the sport list.
4.- User has already defined at least one course as 'attending'.
5.- User has at least one friend.

10.5 Post-conditions

1.- User's friends are able to see which courses is the user going to attend and at which date and time.

10.6 **Main Scenario**
1.- User launches app.
2.- User navigates to the friends tab.
3.- User clicks on the 'share schedule option'.
4.- User decides which friends is he/she sending his/her schedule to.
5.- System sends notifications to friends.

10.7 **Alternative Scenarios**
a) User has no friends:
1.- User receives an error, as its message has no recepient.

10.8 **Special Requirements**
The user has to have an internet connection available in order to send the messages.

10.9 **Notes**
Push notifications have to be enabled in friend's devices for them to receive the messages.

2.2. Actor characteristics

- The User:

The user main characteristics are its ability to interact with the device, which runs the Sport@Unibe app, and its interest in taking part in the university's sports activities.

- The User's Friends:

The characteristics of the User's Friends are the same as the users, since the user is at the same time, seen from another perspective, someone else's friend.

- Back-End Server:

Stores tables with all the information from all users and all courses. Provides service for pushing notifications, rating courses, attending courses, managing friends, and obtaining courses locations.

- Sport@Unibe Homepage:

Back-End server gets sports up-to-date information from the university's homepage.

- Google Maps:

Shows the user the events' locations and provides him/her with directions.

3. Specific requirements

3.1. Functional requirements

A brief list of some of the functional requirements of the application is given here:

- see a list of all sports in campus
- see courses schedules for a selected sport
- define favorite courses
- see schedule of favorite courses
- see the location of a selected course
- show directions of a selected course
- see a list of courses that match a certain criteria (extended search)
- have notifications for upcoming courses
- rate an attended course
- see the ratings of a certain course
- let friends know which courses I plan to go to
- provide user sport description (photos and text)

3.2. Non-functional requirements

The most important non-functional requirements are given on the following list:

- fast performance in every possible Android version
- different landscape and portrait view

- automatic adjustment to any screen size
- utilize UniBern colored theme
- the protection of the user data (friends, account, passwords, etc)