UNIVERSITY OF ZAGREB

FACULTY OF ORGANIZATION AND INFORMATICS

VARAŽDIN

Marko Bartolić, mbartoli@foi.hr

Rene Škuljević, rskuljev@foi.hr

Tomislav Vunak, tvunak@foi.hr

NAVIGATION ON A SKI SLOPE

PROJECT DOCUMENTATION FOR SOFTWARE ANALYSIS AND DEVELOPMENT PROJECT

Varaždin, 2016.

UNIVERSITY OF ZAGREB

FACULTY OF ORGANIZATION AND INFORMATICS

VARAŽDIN

Team number: T10

Team members:

Marko Bartolić, 0016092146

Rene Škuljević, 0016092375

Tomislav Vunak, 0016091502

NAVIGATION ON A SKI SLOPE

PROJECT DOCUMENTATION FOR SOFTWARE ANALYSIS AND DEVELOPMENT PROJECT

Mentors:

Doc. dr. sc. Zlatko Stapić

Ivan Švogor, mag.inf.

Evolaris mentors:

Gerald Binder

Thomas Rößler

Varaždin, 2016.

Table of contents

1. Introduction

We are developing a mobile application for the Evolaris company. They had already worked on similar application for navigation on ski slope. We are going to develop a software which will focus on a more fulfilling user experience while skiing. Our main goal is to implement the algorithm for navigation on a ski track. Evolaris will provide us with the existing part of their application. We will implement new algorithm for calculating if the skier is on the right course and where should he turn next.

Evolaris is Austrian company founded in 2000 and is based in Graz. Their primary focus rests on the conception and development of digital assistance systems in the industrial and commerce sector. They guarantee individual solutions with a unique user experience. Also they are very proud on their efficient and attractive interfaces that enable end users to make the best use of information, which adds value for businesses. In addition to that, they advise companies in their selection of suitable technologies, design individual application scenarios and develop assistance systems on the basis of mobile and wearable technologies.

The purpose of this application is to help the skier determine if he is skiing on the track and to alert him if he leaves the track. Also, the application will calculate the position of the skier on the track and when he approaches a junction on the track, notify him about which turn he should take.

We were given the Recon Snow2 ski goggles for testing purposes for our application. Also Evolaris provided us with their web server and database which are used for storing information about tracks.

Recon Snow2 are ski goggles from Canadian company Recon Instruments. They produce smart glasses and heads-up displays for sports.

Application code and project documentation will be stored in our public GitHub repository https://github.com/rskuljev/Navigation-on-a-ski-slope/ .

This document contains information about software development methodology and how we implemented it in our project. Other than this document, the project characteristics are explained in technical documentation.

2. Software development methodology

For the development of this project we are going to use agile software development methodology. There are few agile methodologies, like Scrum, Extreme Programming, etc. Since the requirement of our mentors is to use Scrum, that is what we will be using.

Scrum methodology has demands and boundaries which we need to respect. First one is having a Scrum master, a person who will ensure that team achieves all sprint goals. In our team Scrum master needs to be part of the development team, and will be responsible for delivering appropriate project documentation on time. We have three iterations, but not all of them are the same length, mostly because we depend on availability of information that we received from Evolaris company and deadlines determined by our mentors.

The tools we chose to accomplish this task include Android Studio and Java programming language for the application itself. The tracking algorithm was first implemented in C# using Visual Studio, but was later rewritten in Java. For the mock up design, we used web application called NinjaMock. The mock up design is shown at the end of this document. As for the implementation of Scrum, we discovered and used beta web application called Taiga.io which can be accessed here https://tree.taiga.io/project/mbartoli-navigation-on-a-ski-slope/ .

Scrum methodology has so far helped us in organizing our time better and giving and tracking responsibility to all team members. Two main problems which we had to solve in this project was to determine the position of the skier and alert him if he leaves the track and to show the appropriate arrow in which direction the skier should turn while arriving at junction. This project was finished in three iterations. In first iteration our goal is to discuss and agree on functionalities of the application, implement the main menu and prepare everything for future implementations. In second iteration we have to focus on implementing the tracking algorithm in C# (which will later be rewritten in Java) and to solve the problem of tracking the course on a ski track. Finally, in third iteration we have to translate the tracking algorithm in Java, connect with Evolaris' web server and show the appropriate arrow on the display when the skier arrives at junction.

2.1. Scrum team

Our Scrum team consists of three roles, which are Product Owner, Scrum Master and Development team. We assigned roles to all team members and every team member has responsibilities which depend on their assigned role. Still, as our mentors requested, each team member will be working on the development of the application. Scrum roles can be seen in figure 1.

Product owner	Scrum Master	Development Team
Gerald Binder Evolaris company	Marko Bartolić	Rene Škuljević Tomislav Vunak

Figure 1. Scrum roles

2.2. Product backlog

Product backlog is a list of work that should be done by the development team. Items in the product backlog are defined by product owner, but the Scrum Master sorts them into sprints depending on their importance. Our product backlog was refactored a few times because it took some time until we finally agreed what Evolaris demands from us and what we can do. This tasks are distributed in three sprints. As you will see, not all of them are completed. Product backlog for this project is shown in figure 2.

Figure 2. Product backlog

2.3. Tasks

Each task in a sprint can be in one of the following states: “New”, “In progress”, “Ready for test”, “Closed” and “Needs info”. When defining a new task, it is assigned to one person in the development team and it is defined how much work it requires. Each member of the team has to update the status of his own tasks. Example is shown in figure 3.

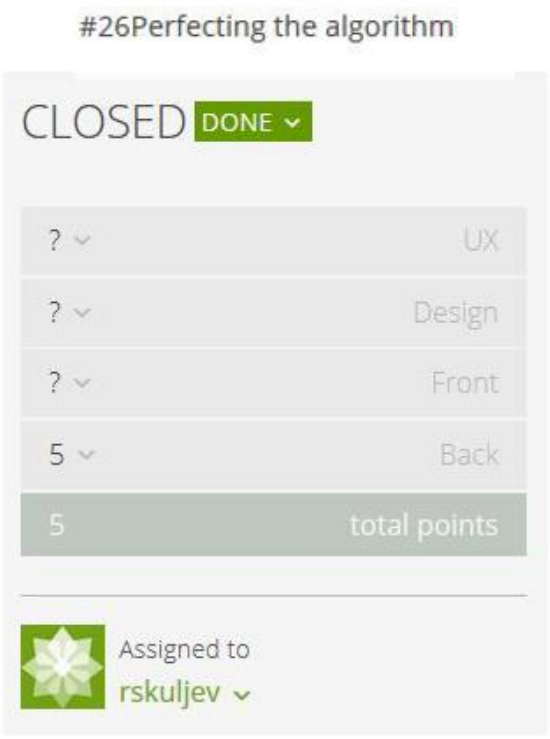

Figure 3. Example of a task

2.4. Sprint 1

Sprint one began on 12th October 2015 and ends on 27th October 2015. It has 35 completed points and 8 tasks. Since Sprint 1 started before our final agreement, during this first sprint we focused on basic functionalities of the application, which can be seen in figure 4.

Figure 4. Tasks in sprint 1

Performance of the development team can be analyzed by using burndown charts. The gray line shown on the graph represents the ideal distribution of finishing tasks. The green line shows the actual work done by the development team. If the green line is above the gray one, it means that the development team is behind schedule and the project is late. Contrary, if the green line is below the gray one, it means that the project is ahead of schedule. Sprint 1 burndown chart of our project can be seen in figure 5, and it shows that throughout the whole sprint the project was a bit late.

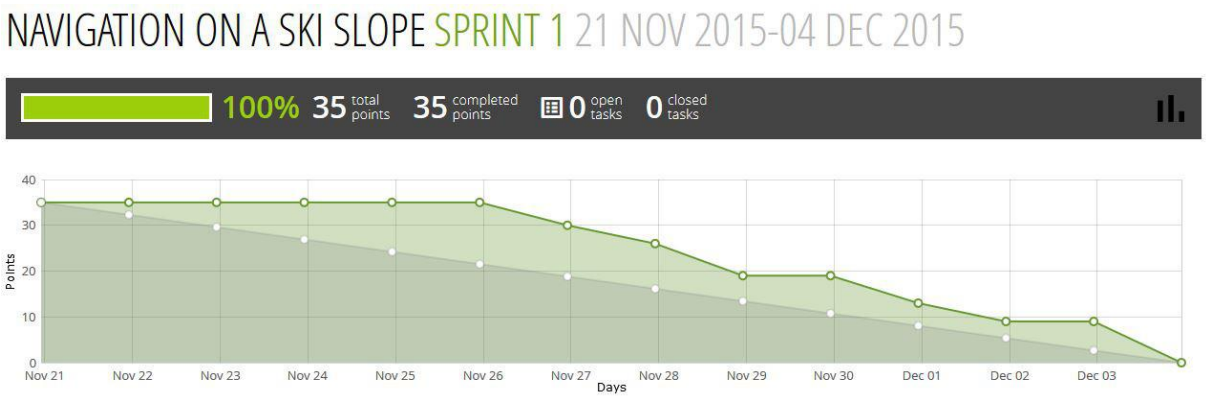

Figure 5. Sprint 1 burndown chart

2.5. Sprint 2

Our second sprint started on December 5th 2015 and ended on December 19th. Here we had a lot of issues. The first task, the implementation of Retrofit was the only task we managed to finish on time. Next we started the implementation of the local database using ActiveAndroid, but halfway through we realized that we may not need the local database and that we can store the data from the web server in lists and directly display them on screen without saving in the local database. Thus the second task, local database, remained unfinished. The final task for the second sprint was the implementation of the tracking algorithm, used for tracking the skier's movement. In the beginning, we experimented with so called RANSAC algorithm, but it was too complicated and it didn't produce the result we wanted, so we found another algorithm which better suited our needs. Still, we haven't managed to finish that task on time, so it was continued in the third sprint. Task for Sprint 2 can be seen in figure 6.

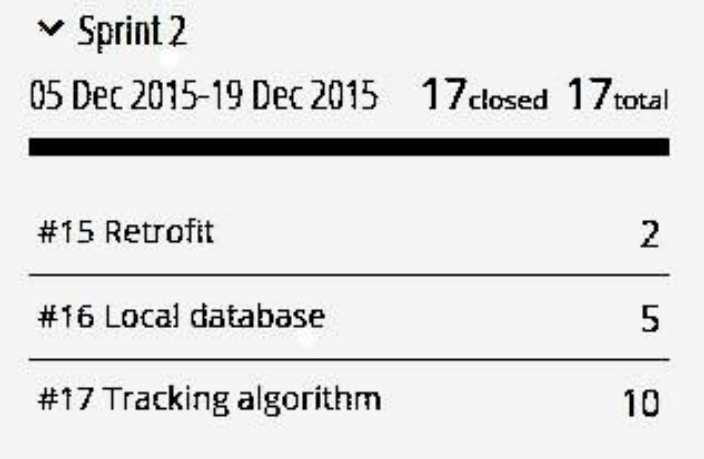

Figure 6. Tasks in sprint 2

As for the burndown chart of the Sprint 2, which can be seen in figure 7, it shows that green line was below the gray one in the beginning. That is because we had to remove some tasks which we agreed not to do after the sprint had already started. One task was finished, which can be seen on the chart, but the other were not finished during the second sprint, meaning the green line representing the actual work remains flat until the end of sprint, leaving the project a bit behind schedule.

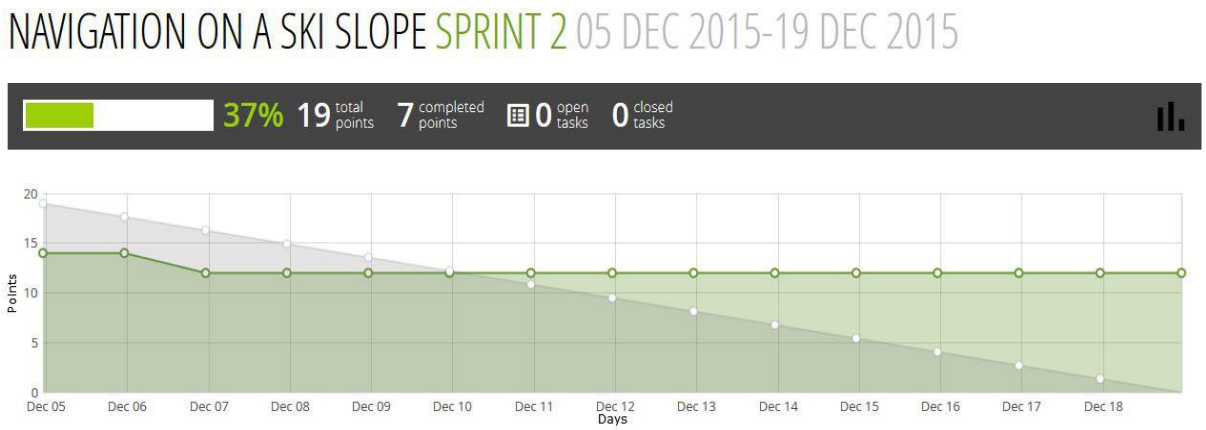

Figure 7. Sprint 2 burndown chart

2.6. Sprint 3

Third and final sprint began on December 19th 2015 and finished on February 3rd 2016. Task for the third sprint can be seen in figure X. During this sprint, we were working on finalizing the application, connecting with the web server and perfecting the tracking algorithm. Only one task was not done during this sprint, "Routes search" which we agreed not to implement.

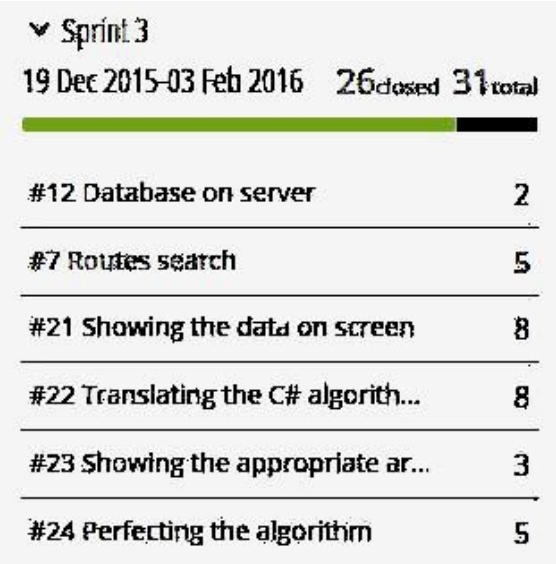

Figure 8. Tasks in sprint 3

Burndown chart for Sprint 3 can be seen in figure 9. As you can see, the sprint is pretty long, and the graph remains flat most of the time. That is because the development team of the web application we were using for SCRUM (Taiga.io), was performing an update to their application in late December, which altered some of the functionalities of their application, causing our graph not to change shape despite tasks being finished. That problem was later resolved by contacting their support team and then we proceeded as planned.

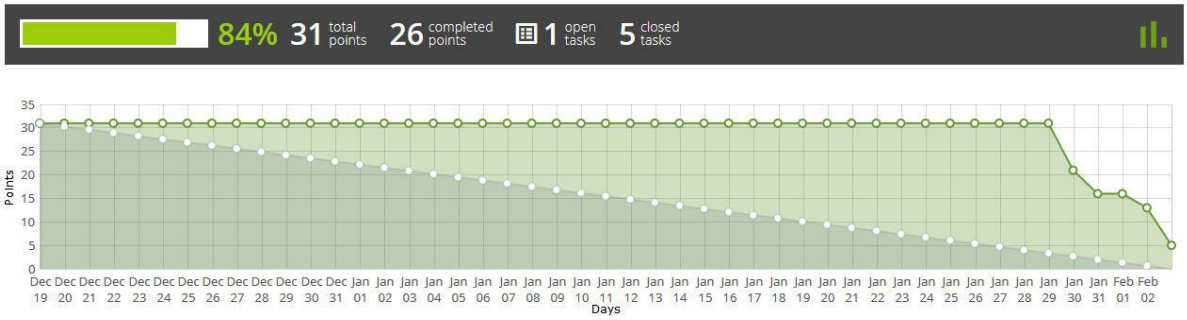

Figure 9. Sprint 3 burndown chart

3. Mock up

Mock up is a model of the application design used for demonstration and evaluation purposes. Using mock ups, designers and developers can analyze design, layout and functionality. Figures X show our idea of the application at the beginning of the project. As you will see, it differs greatly from the final product.

Figure 10. Mock ups 1 and 2

Figure 11. Mock ups 3 and 4