JustMeet
Final Project Report

Version 1.0

Revision History

Date	Version	Description	Author
15-01-2016	1.0	First version	Armindo Carvalho
20-01-2016	1.0	Metrics & Final review	Danijel Sokač

1. Introduction

1.1 Propose of the document

The present document intends to make an overview of all the project that has been developed during the DSD course at Faculty of Electrical Engineering and Computing of University of Zagreb and Information Engineering School of Politecnico di Milano. In this report you will find some information regarding the project and its goals, process of development, all the materials produced, and some reflection about the lessons learned during this distributed project.

1.2 Document Organization

The document is organized in this: chapters 2,3,4,5 and 6 will describe all the project and chapters 7,8 and 9, will show you how the work was organized and what the team learnt from the project.

1.3 Scope

The scope of this document is to summarize all the activities of the DSD project and at the same time make some reflection about the all the development process and provide some metrics that will help to analyze the developers performance and level of organization.

1.4 Intended Audience

The intended audience of this document are all the people that are involved in DSD course, namely, the project team, the supervisors of the project(Ivana Bosnić and Elisabetta Di Nitto) and all the DSD staff and students.

1.5 Definitions and acronyms

FER	Faculty of Electrical Engineering and Computing
POLIMI	Politecnico Di Milano
MDH	Mälardalen University
DSD	Distributed Software Development
JustMeet	The name of the application that was developed

1.6 References

Project DSD page: http://www.fer.unizg.hr/rasip/dsd/projects/meetme_2
Project application page: http://justmeetdsd.herokuapp.com
Project code: https://github.com/jnadal/MeetMeDSD
Browser automation:
https://github.com/kacso/JustMeetDSD-browserAutomation

2. Background and Objectives

2.1 Background

Today's planners do not meet the needs of a user. Person willing to easily schedule an event can stump into many problems in the process of organizing a meeting. The problem can become even more complex when the number of people willing to attend the meeting grows. The proposed timeslots can be spread over few different meeting times, because each of the attendants have their own schedules they have to cope with and they have scheduled before. Because of the many constraints force upon the other attendances it is hard to find appropriate time slot that can work out for everyone. Usually in this situation the meeting is held over email conversation, which can become quite messy.

2.2 Project Goals

The main goal of the project that is presented in this document was to develop one application that allows the users to schedule events. The application should use different types of sources to get the user calendars (for instance, Google calendars) and should have a user friendly interface so the user can easily use it. Another goal of this project was to develop all this features using disruptive technologies (like NodeJS as we used).

Another important goal of the project was the distributed part of development, so the members of the team should organize themselves to achieve the goals of the project developing in a distributed way.

2.3 Project Requirements

The project main requirements were to build an application to manage the scheduling of events with new technologies and this application should work better than the applications that exist already in the internet (Doodle, for instance). Then you have a list of all the requirements of this project

The system should draw from Google Calendar and/or widely used calendaring applications to help choose prospective meeting times.
The system should respect privacy and security preferences of users.
The system should support web-based responses from prospective meeting participants, on both desktop/laptop and mobile platforms.

The system should be able to take care of different event time zone in a strong and appropriate way.
The system should provide guest users to register for using the planner
The system should provide users with option to invite people to join the created event
The system should provide users possibility to create meeting event
The system should be able to provide to the user possibility to add an event to his personal calendar
The system should be able to get availability information of invited users and integrate it to the calendar of the proposed event respecting users privacy

2.4 Project Milestones and Deliverables

Project Plan & Vision Document	This document intended to plan the activities for the project and provide some more information about the goals
Requirement Definition Document	In this document the team presented the requirements to the project and the strategy to implement it
Design Description Document	In this document team presents the architecture of the system
Alpha Prototype	First presentation of the application and basic features
Beta Prototype	Presentation the main features of the application
Acceptance Test Document	This document have the test that were made to guarantee the acceptance of the project
Final Product	The final product implemented with all the proposed features

2.5 Project Testing

In this project, the team adopted two ways of testing, the manual testing and automatic testing. The manual testing was made by the team members and whenever some feature was implemented, all the members tested it trying to find some error or testing different scenarios when the feature is used.

The automatic testing, was implemented in the server side with the tool “Cucumber” plus other tool named “Watir WebDriver”. By this tests basic functionality of application was tested.

Besides that, we also had unit test which were run every time the team member deployed code, and in the case of some errors in the code, the deployed code was not accepted.

3. Organization and Communication

3.1 Project Team

FER	POLIMI
● Armindo Carvalho ● Carolina Casali ● Danijel Sokač ● João Nadais	● Francesco Giarola ● Milica Jovanović ● Mohamed Mehdi Kaabi
Supervisor	
Ivana Bosnić	Elisabetta Di Nitto
Customer	
Michal Young - University of Oregon	

3.2 Roles

As the team choose the Scrum development process the team only have 3 main roles:

- **Scrum Master**: Danijel Sokač
- **Product Owner**: Carolina Casali
- **GitMaster**: João Nadais
- **Developer**: All the remaining members of the team

3.3 Communication

Since the project is developed in the context of the DSD course and the members of the team are not in the same location, all communications was done over Internet, with appropriate tools. In our case we choose different tools so we can communicate efficiently (Facebook, Skype, email).

In the beginning of the project, we had fixed meetings every week on Monday, from 15.00 to 16.00, with the participation of the supervisors and on Thursday, from 20.00 to 21.00, with only the team members present, but after two sprints, the team changed the Thursday meeting to Monday, so the sprint start and end meet the DSD deadlines. During all the meetings one team member was assigned with the task of keeping the meeting minutes and making a summary document at the end of it, in order to minimize the possibility of misunderstandings, and we invite the reader to read some of this documents in our DSD page.

Summary meetings for getting quick status updates are done every day through short Skype chats or Facebook chats. In the middle of the semester, the team adopted to make daily reports in a Google Drive document, so all the team members know all work that is being done.

3.4 Tools

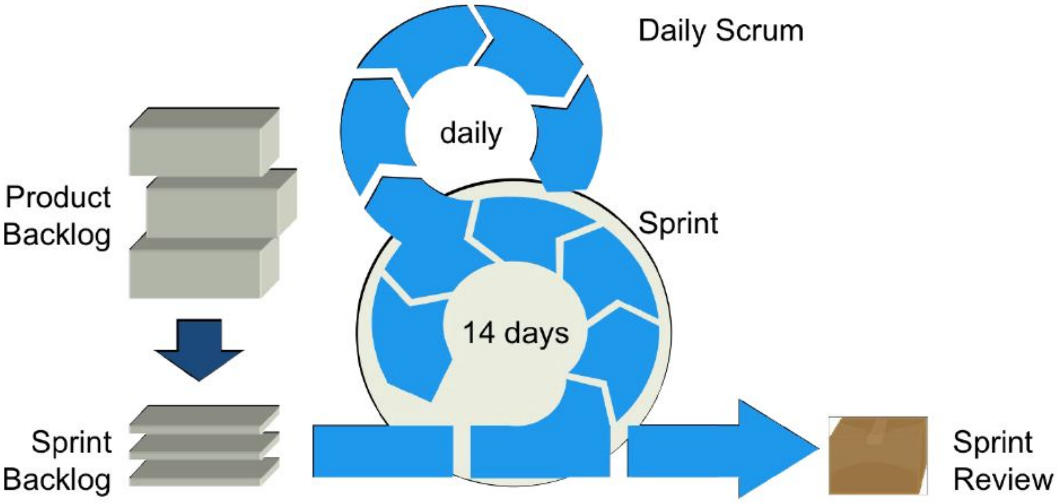
The team have choose several tools to sharing information and coordinate all the work between all the team members:

Daily Meetings	Facebook
Weekly Meetings	Skype
Task Management	Asana
Document Repository	Google Drive
Code versioning	GitHub

4. Development Process

4.1 Scrum

The team have choose to follow and agile approach using software development framework called SCRUM. Communication and customer collaboration are two key aspects on the base of this approach. The application is developed within iterations called Sprints that last for fourteen days.



4.2 Sprint

The sprints in this project had the duration of two weeks and the team met every two weeks to make the sprint retrospective and prepare the next sprint. As the team was divided in different places, the daily sprint meetings, were replaced by a daily report document in “Google Drive” or using the Facebook chat to give the team members important updated information about the current status of the project.

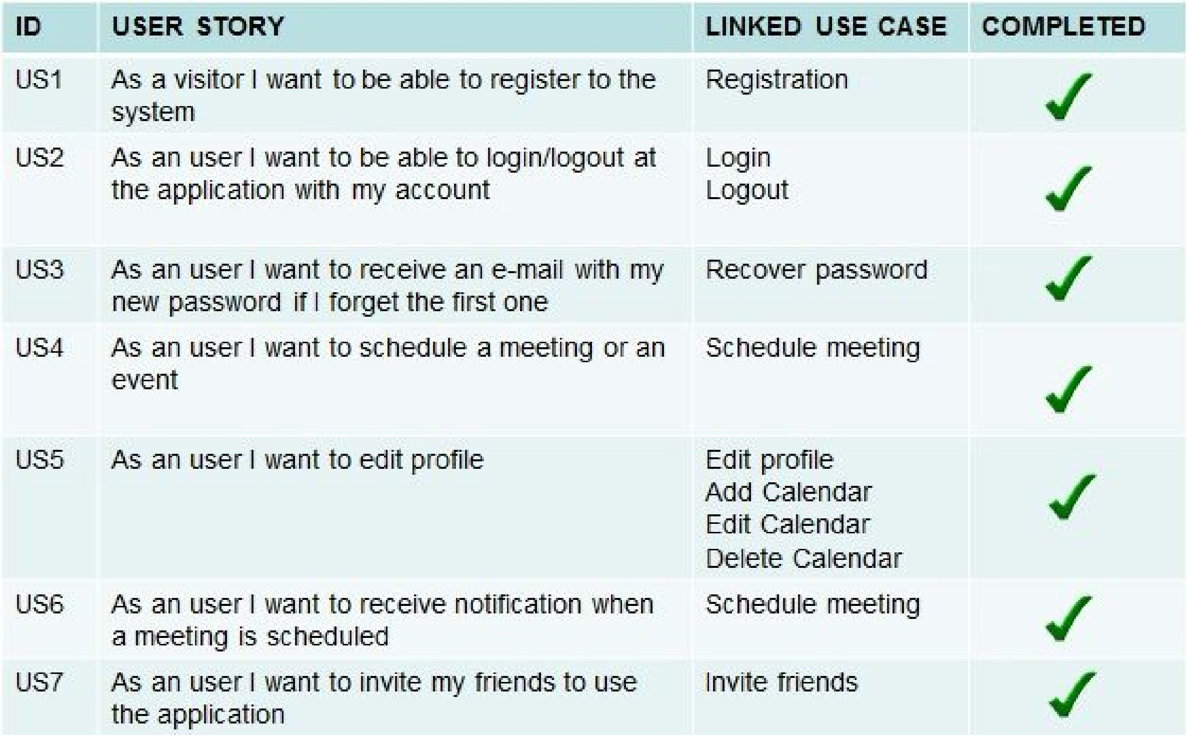
4.3 Issues with distributed development

During the semester, the team faced a diverse range of problems, that can be divided in two big group: the technical issues, that regard the issues related with the technologies and developing code in different places, and personal issues, that regard the issues related with cultural differences and personal differences between all people.

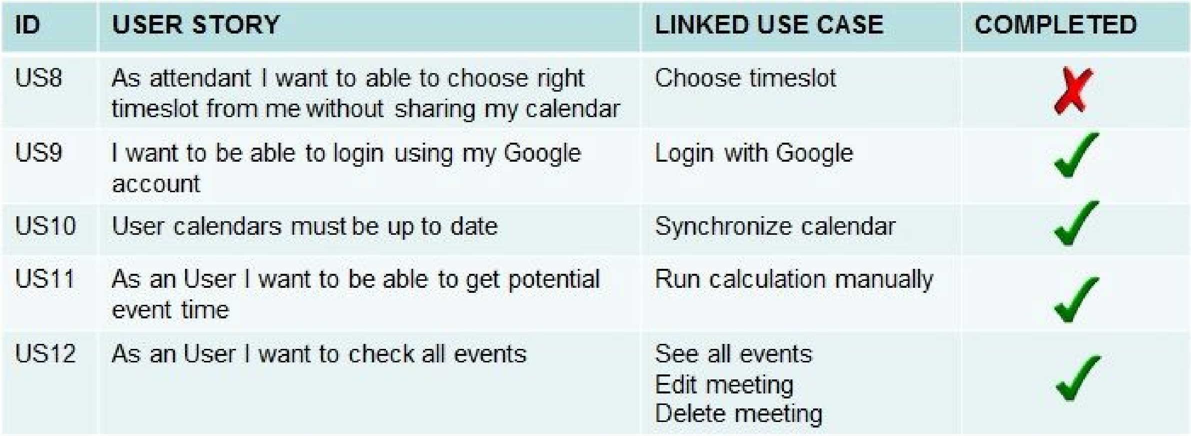
The main technical issues that the team have dealt with, were the merging code of all developers problems, problems accessing the Database, adaptation to new technologies and the fact that our host server was a free version that limits the efficiency of the server.

The main personal issues were some miscommunication between the two sides of the team and the fact that none of the team members worked together before so they were not accustomed to each other way of work.

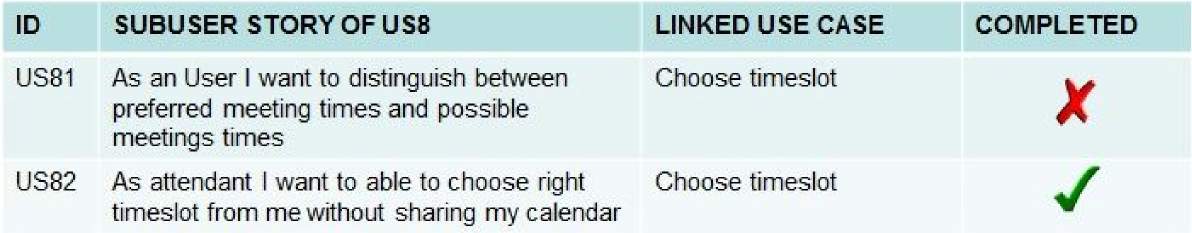
5. Project Final Product



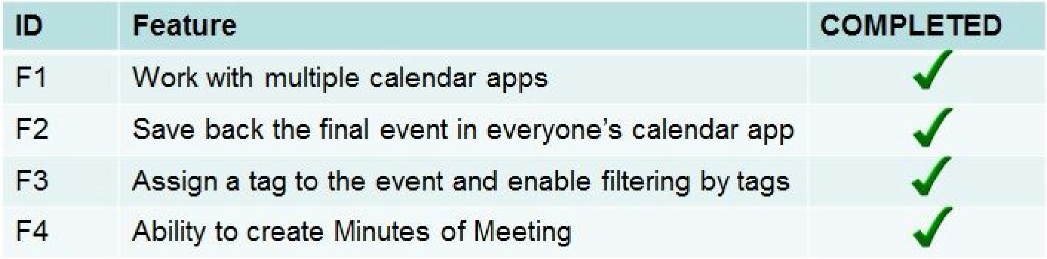
ID	USER STORY	LINKED USE CASE	COMPLETED
US1	As a visitor I want to be able to register to the system	Registration	✓
US2	As an user I want to be able to login/logout at the application with my account	Login Logout	✓
US3	As an user I want to receive an e-mail with my new password if I forget the first one	Recover password	✓
US4	As an user I want to schedule a meeting or an event	Schedule meeting	✓
US5	As an user I want to edit profile	Edit profile Add Calendar Edit Calendar Delete Calendar	✓
US6	As an user I want to receive notification when a meeting is scheduled	Schedule meeting	✓
US7	As an user I want to invite my friends to use the application	Invite friends	✓



ID	USER STORY	LINKED USE CASE	COMPLETED
US8	As attendant I want to able to choose right timeslot from me without sharing my calendar	Choose timeslot	✗
US9	I want to be able to login using my Google account	Login with Google	✓
US10	User calendars must be up to date	Synchronize calendar	✓
US11	As an User I want to be able to get potential event time	Run calculation manually	✓
US12	As an User I want to check all events	See all events Edit meeting Delete meeting	✓



ID	SUBUSER STORY OF US8	LINKED USE CASE	COMPLETED
US81	As an User I want to distinguish between preferred meeting times and possible meetings times	Choose timeslot	✗
US82	As attendant I want to able to choose right timeslot from me without sharing my calendar	Choose timeslot	✓



ID	Feature	COMPLETED
F1	Work with multiple calendar apps	✓
F2	Save back the final event in everyone's calendar app	✓
F3	Assign a tag to the event and enable filtering by tags	✓
F4	Ability to create Minutes of Meeting	✓

6. Lessons Learned

We were able to take a big amount of lessons from the development of this project, mainly related to the composition of our team. In the beginning, we were taught that managing a distributed team was not easy, but we got to find out that the biggest issue was not only managing all the people but also taking the most advantage of the background of each person. 7 people from different universities and never in the same place reveal issues mostly reaching to a common knowledge. As we were distributed, we realized the importance of communication. If you are in the same place as the rest of your team, you can just get to them and explain what have you actually been doing. However, when that is not the case, if for some reason you can not see their messages or don't reply, then it goes to a dead end.

As for the project itself, developing it made us face new kinds of problems. For some of us, new web development frameworks were learnt. For others, a new way of thinking (and therefore developing) was learnt (asynchronous programming). Finally, for those who already had experience with the language, not only they got better developers in it but they learnt how to solve problems related to scheduling events. In the end, it was for sure a joyful experience that we will remember for what comes on.

7. Metrics and Analytics

7.1 Project metrics

In the following graph completion of tasks divided by sprints is shown. Remaining tasks are some improvements which can be taken as possible upgrades for the application.

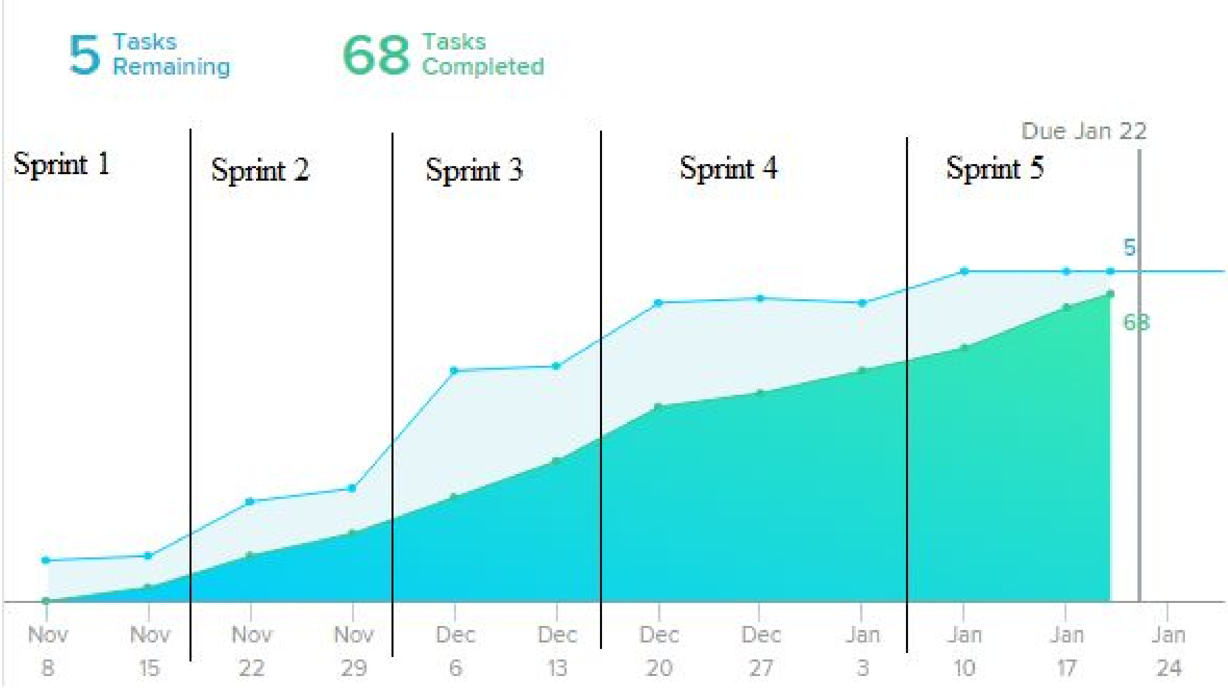


Figure 1. Project progress

7.1.1 Invested hours

In the following graph is shown our progress during the project according to working hours. It can be seen that graph is linear which means that our work was equally divided during all project.

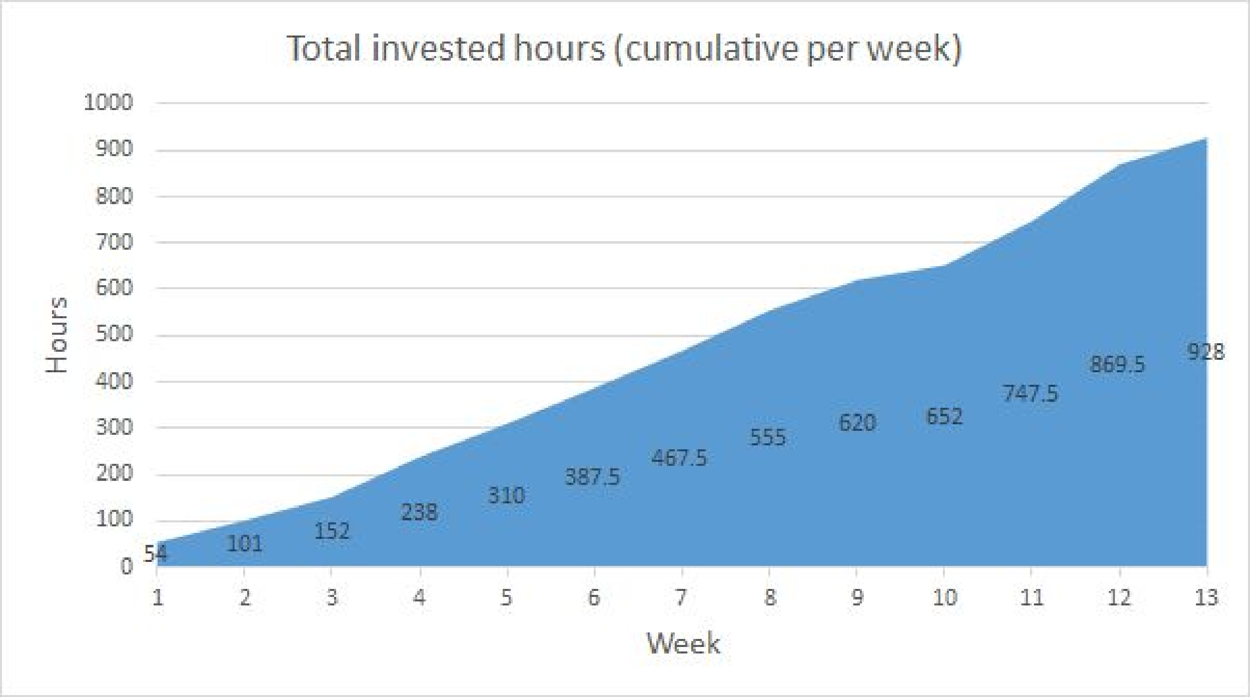


Figure 2. Total invested hours per week

7.1.2 Git metric

In the following graph is shown our git statistic. From this statistic it can be seen when we were most active in developing our project.

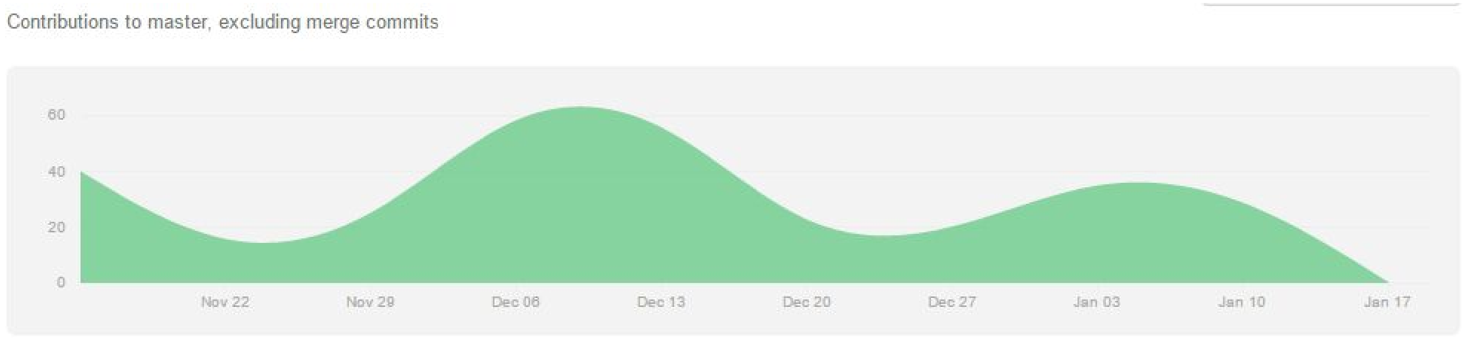


Figure 3. Contribution to master

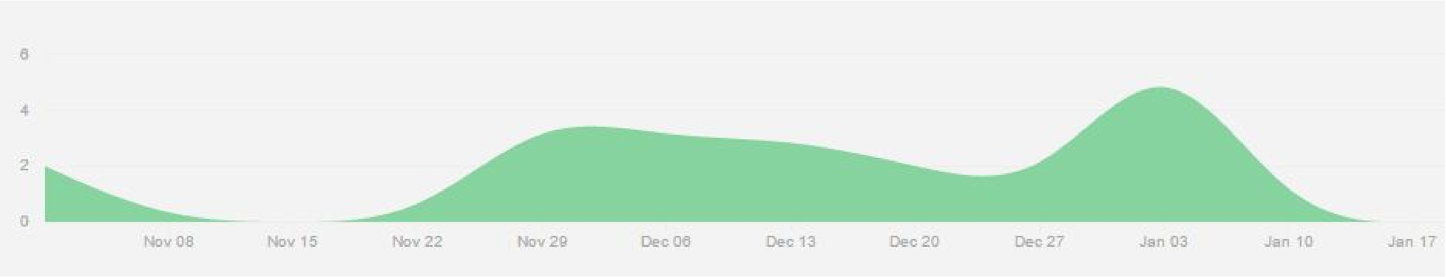


Figure 4. Contribution in browser automation

Figure 5. Code frequency

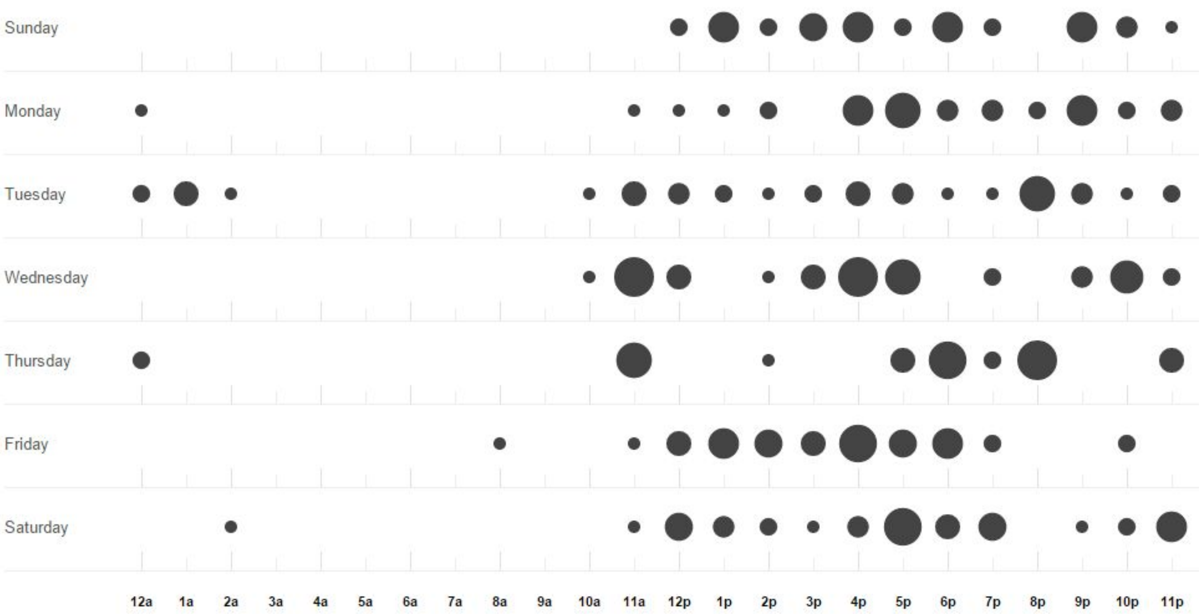


Figure 6. Active time

7.2 Per team member metrics

7.2.1 Invested hours

In the following graph can be seen our effort divided per week and also per team member.

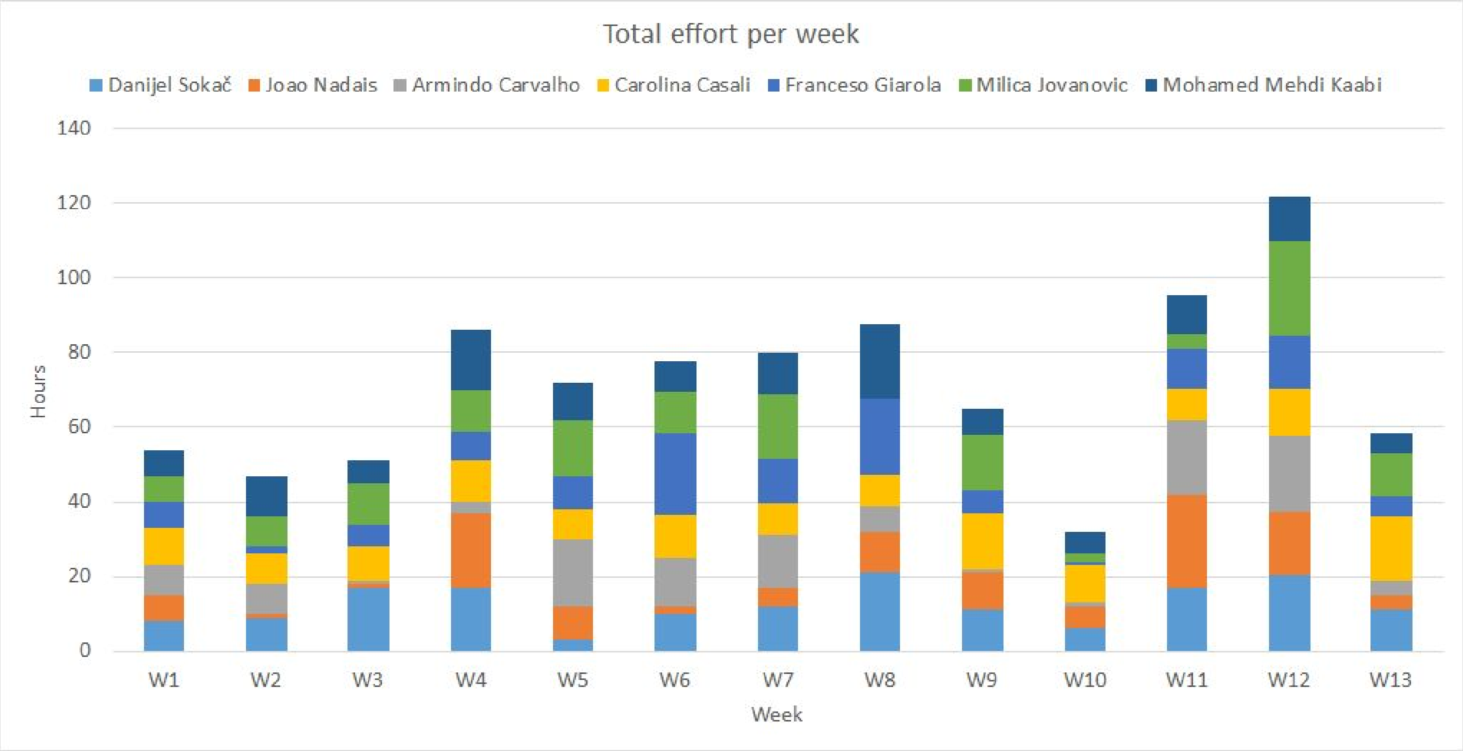
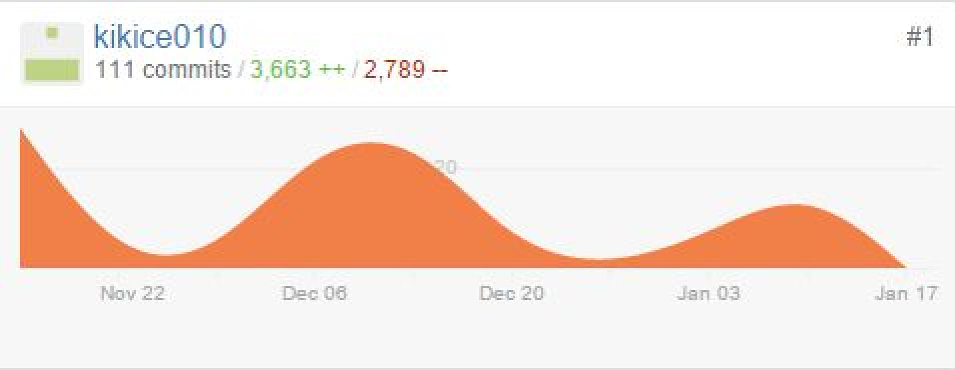


Figure 7. Total effort per week and team member

7.2.2 Git statistic

In the following graphs is shown per team member statistic on git. It can be also compared to overall git statistic shown in chapter 7.1.2.


kikice010 #1

111 commits / 3,663 ++ / 2,789 --

Figure 8. Milica Jovanović app contribution

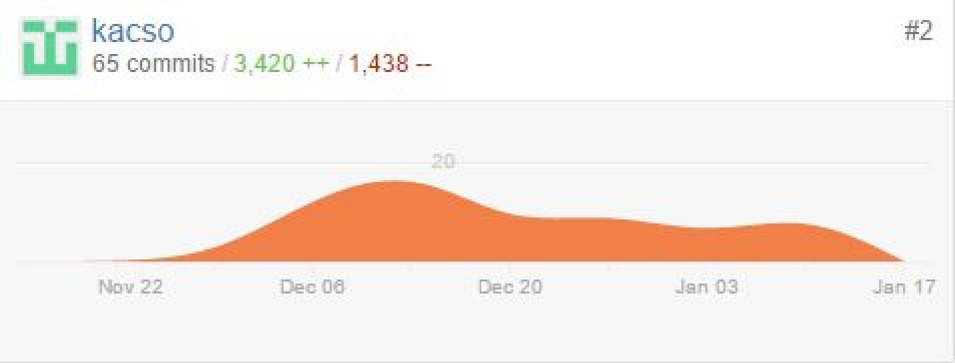


Figure 9. Danijel Sokač app contribution

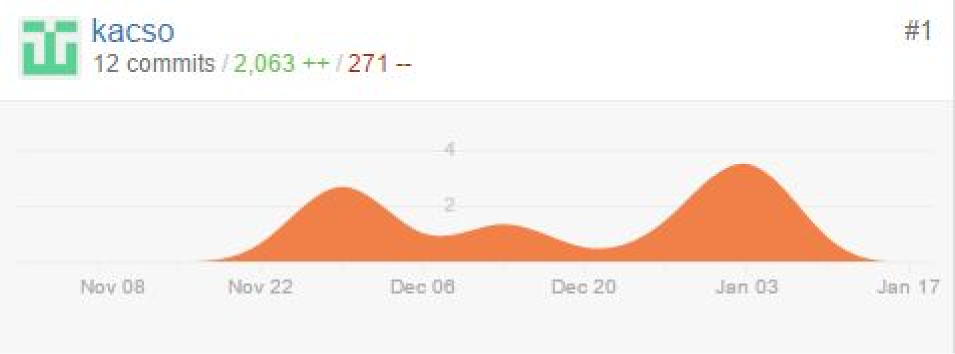


Figure 10. Danijel Sokač browser automation contribution

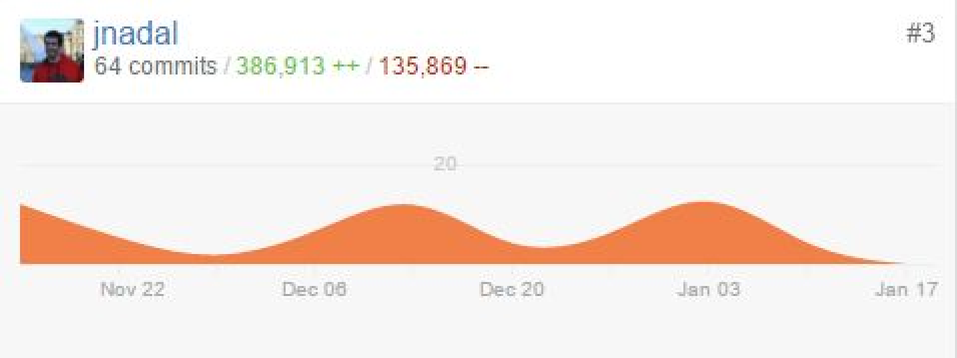


Figure 11. Joao Nadais app contribution

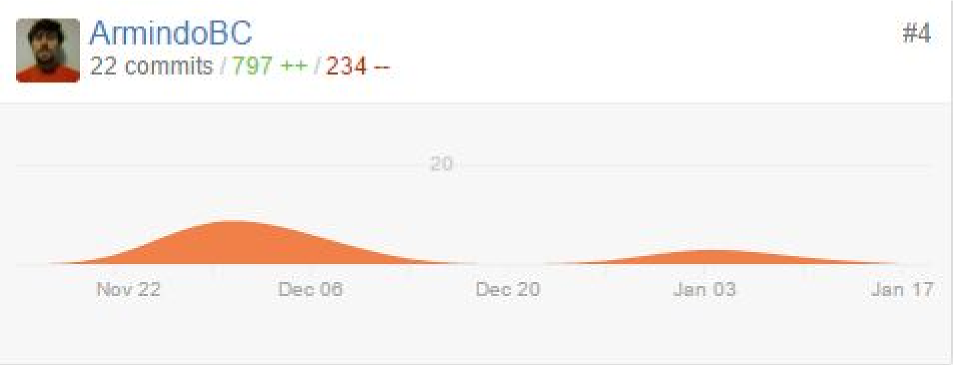


Figure 12. Armindo Barbosa Carvalho app contribution

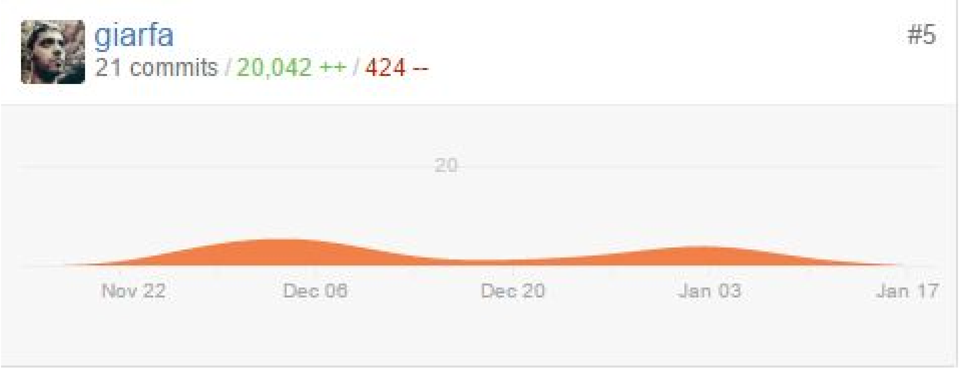


Figure 13. Francesco Giarola app contribution

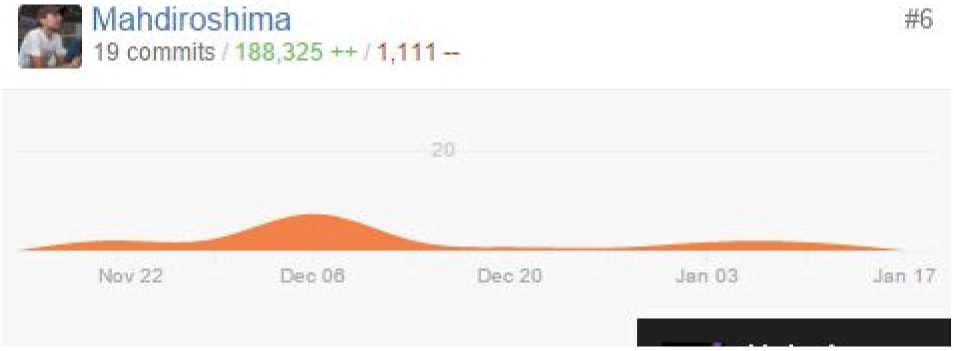


Figure 14. Mohamed Mehdi Kaabi app contribution

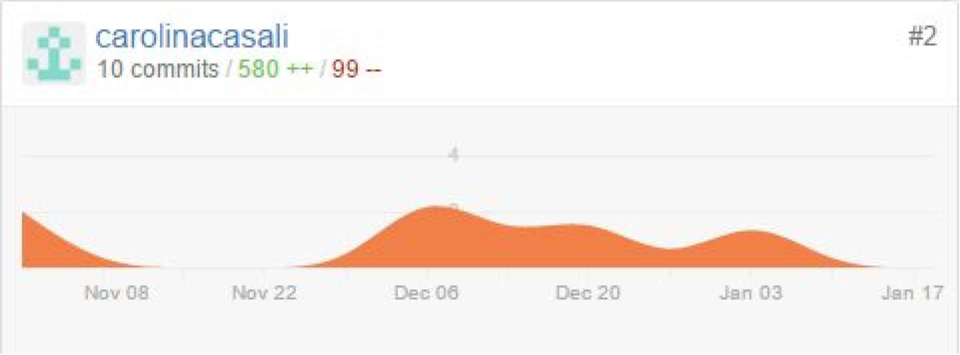


Figure 15. Carolina Casali browser automation

7.3 Per side statistic

For the end is shown statistic on per side basis with addition of comparison of the whole team. All this statistic is according to invested hours of each team member. From this statistic it can be seen that at the end, each team member was investing similar amount of time during the project. From this, it can be concluded that our division of tasks was successful.

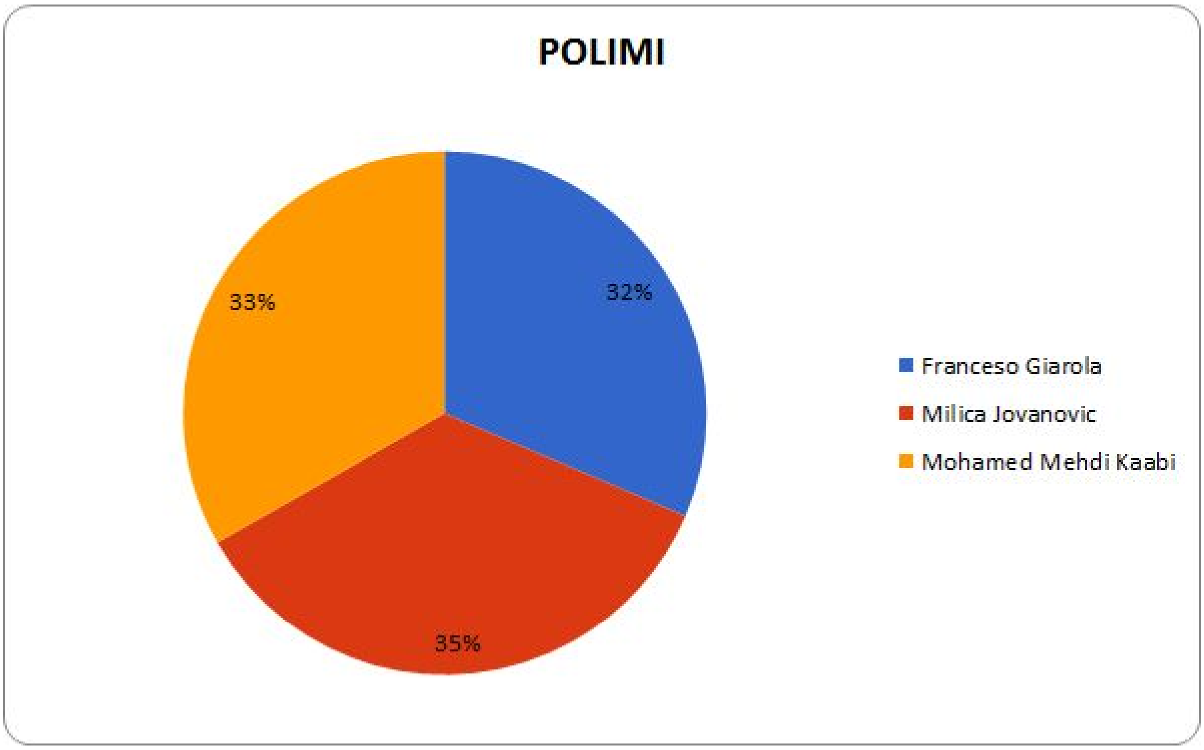


Figure 16. Polimi side division of work in working hours

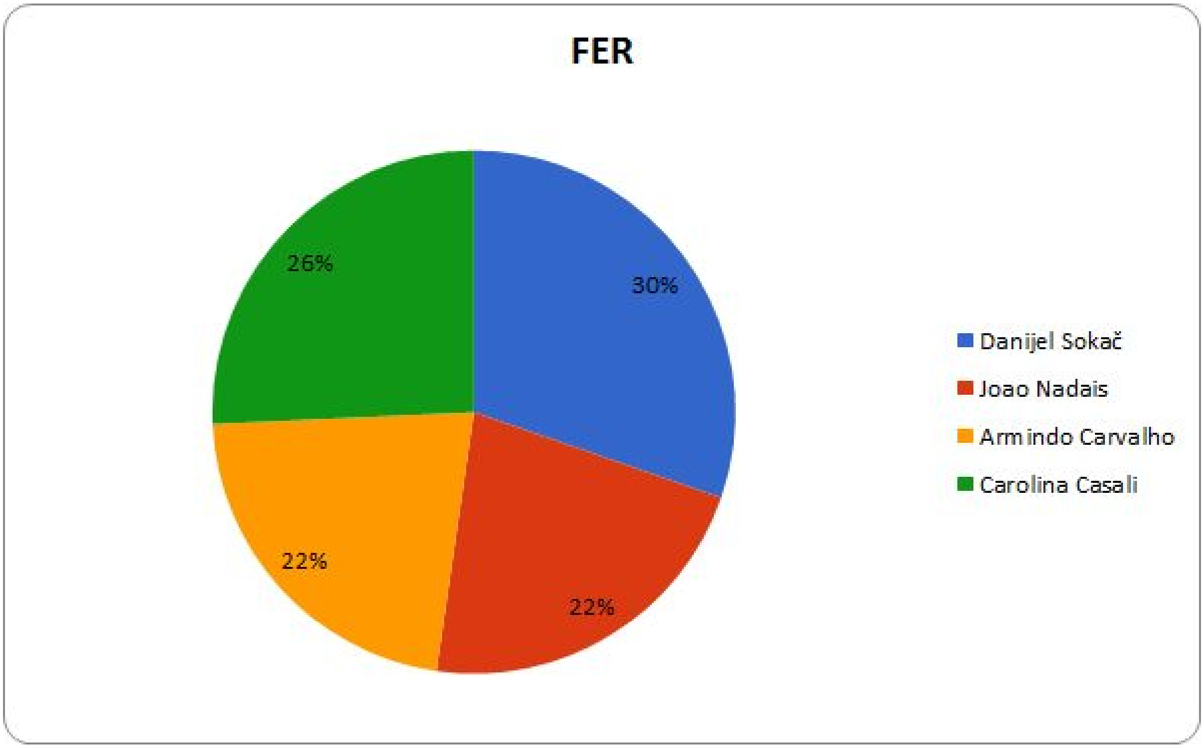


Figure 17. FER side division of work in working hours

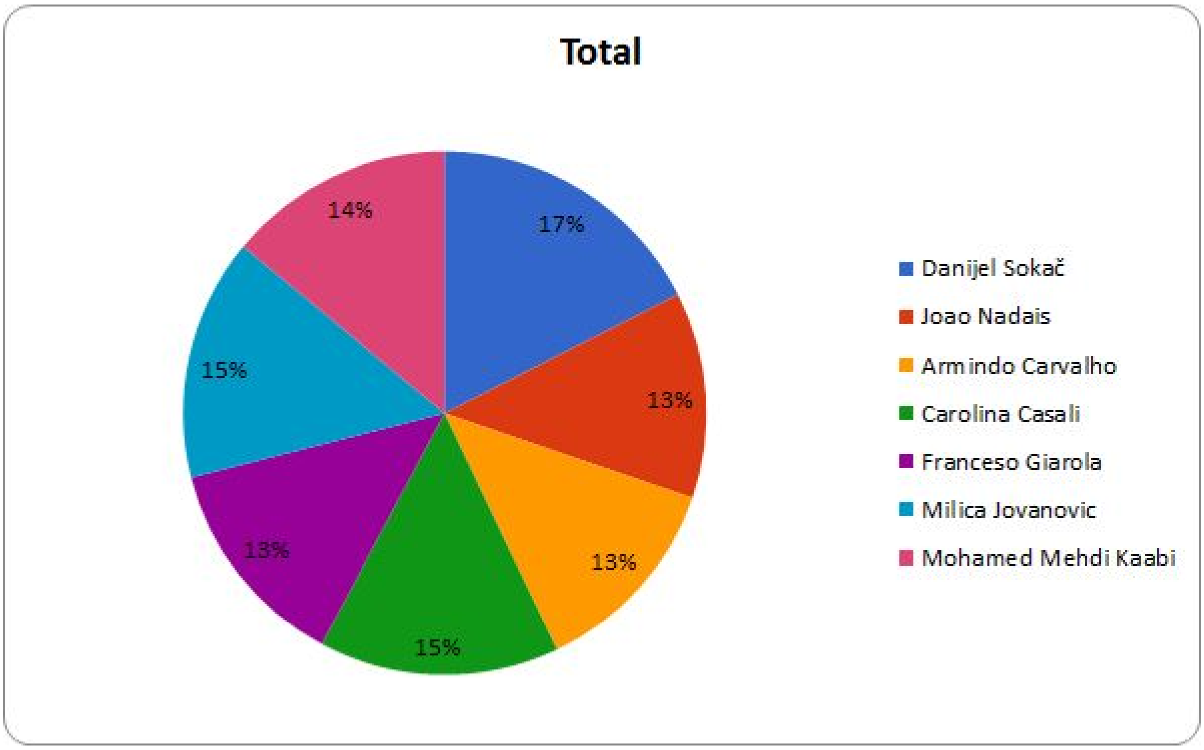


Figure 18. Team division of work in working hours

8. Discussion

After all the project, the team concludes that despite the all the work and all the issues that we had, the experience was very profitable for all the team members. The member end this experience, knowing more technologies and at the same they got more experience working in team and dealing with the different issues that can occur in a project.

If the project started today, all the team members would be more prepared to manage all the problems and challenges that this kind of projects provide. So, the team considers that all the goals of this course were achieved with success and all the team members are very proud of this experience that DSD course gave to us.