



GiftCase

GiftCase

Final project report

Version 1.05



Revision History

Date	Version	Description	Author
2015-01-08	1.01	Initial draft	Vlatko Klabučar
2015-01-16	1.02	Chapter 3, 4, 5, 7, 8, and 9	Gijs Bos
2015-01-16	1.03	Github Statistics added	Vlatko Klabučar
2015-01-16	1.04	Final chapters added	Vlatko Klabučar, Gijs Bos, Giulio Cattivera
2015-01-23	1.05	Review, formatting and general fixing	Giulio Cattivera

Table of Contents

- [1 Introduction](#)
 - [1.1 Purpose of this document](#)
 - [1.2 Intended audience](#)
 - [1.3 Scope](#)
 - [1.4 Definitions and acronyms](#)
 - [1.4.1 Definitions](#)
 - [1.4.2 Acronyms and abbreviations](#)
 - [1.5 References](#)
- [2 Background and Objectives](#)
 - [2.1 Overview](#)
 - [2.2 Customer](#)
 - [2.3 Supervisors](#)
 - [2.4 Project vision](#)
- [3 Organization and Communication](#)
 - [3.1 Team organization](#)
 - [3.2 Communication](#)
 - [3.3 Team meetings](#)
- [4 Development Process](#)
 - [4.1 Requirements gathering](#)
 - [4.2 1st sprint](#)
 - [4.3 2nd sprint](#)
 - [4.4 3rd sprint](#)
 - [4.5 4th sprint](#)
- [5 Milestones](#)
 - [5.1 Internal](#)
 - [5.2 External](#)
- [6 Project Results](#)
 - [6.1 Software overview](#)
 - [6.2 Backend](#)
 - [6.3 Fontend](#)
 - [6.4 Missing features](#)
 - [6.5 Possible improvements](#)
 - [6.6 Deliverables](#)
 - [6.7 Documentation](#)
- [7 Project Risks](#)
- [8 Project Experiences](#)
 - [8.1 Environment](#)
 - [8.2 Technologies](#)
- [9 Metrics](#)
 - [9.1 Work per member](#)
 - [Figure x. Commits on Github](#)
 - [9.2 Milestone metrics](#)
 - [9.3 Effort metrics](#)
- [References](#)

1 Introduction

1.1 Purpose of this document

The purpose of this document is to show an overview of the project GiftCase so that it provides detailed documentation for the parties interested in examining this project. This document contains information about objectives, team organisation and communication, development process, important milestones and overall project results.

1.2 Intended audience

The intended audience is:

- Team members
- The customer
- Project supervisors
- Anyone who wants to get familiar with the project

1.3 Scope

The scope of this document is overall description of project progress and results. It includes background and objectives, internal and external product milestones, development process, project results relative to the requirements. It also includes work metrics for this project.

1.4 Definitions and acronyms

Acronyms contained in this document are provided in the next few sections. This sections is included to clarify specific abbreviations or acronyms relevant to the project.

1.4.1 Definitions

Table 1. Definitions used throughout the document

Keyword	Definition
GiftCase	The name of the project
Back-end	Application which is used to provide the functionality communicated with databases and other information sources.
Mobile application	An application which is targeted to be executed on a mobile device
App42	Is a specific BaaS provider we have chosen to put in the system

1.4.2 Acronyms and abbreviations

Table 2. Acronyms and abbreviations used throughout the document

Acronyms or abbreviations	Definitions
FER	Fakultet Elektrotehnike i Računarstva (Faculty of Electrical Engineering and Computing)
MDH	Mälardalens Högskola (Mälardalen University)
BaaS	Back-end as a Service. Model for providing web and mobile app developers with a way to link their applications to back-end cloud storage and APIs.
GUI	Graphical User Interface
REST API	API that adheres to the REST architectural constraints

1.5 References

More information about GiftCase like more documentation and team members can be found here:

<http://www.fer.unizg.hr/rasip/dsd/projects/giftcase>

Requirement definition document:

http://www.fer.unizg.hr/_download/repository/Requirements_Definition_Document%5B1%5D.pdf

Design description:

http://www.fer.unizg.hr/_download/repository/Designdescription_v1.09%5B1%5D.pdf

Acceptance test plan:

http://www.fer.unizg.hr/_download/repository/AcceptanceTestPlan.pdf

2 Background and Objectives

2.1 Overview

The GiftCase project is initiated by Ericsson Nikola Tesla, located in Croatia. The purpose is to create a mobile application with which registered users can send gifts to their contacts. When a user wants to give a gift to another, it is suggested directly by the application that is able to understand the preferences of users mainly through the investigation of its behavior in social networks (Facebook, Linked In). As for gifts, the application performs the search on Ebay or Amazon and suggests a product to the user that wants to donate a gift. The application also suggests gifts individually based on events in the life of registered users, such as birthdays.

2.2 Customer

The customer is Marin Orlić from Ericsson Nikola Tesla, which is the leading regional provider of innovative ICT products, solutions and services in telecommunications, health care, transport, state administration, municipal services and multimedia. It is also Croatia's leading knowledge exporter through its Research and Development activities and Service Delivery Center. Being a member of the Ericsson Group, the Company operates within the Region Western and Central Europe (RWCE).

Website: <http://www.ericsson.hr/>

Contact: marin.orlic@ericsson.com

2.3 Supervisors

Igor Čavrak is the local supervisor of the GiftCase project. He is an assistant professor on Faculty of Electrical Engineering and Computing (FER), University of Zagreb, Croatia.

Website: <http://www.fer.unizg.hr/rasip/icavrak>

Contact: igor.cavrak@fer.hr

Juraj Feljan is the remote supervisor of the GiftCase project. He is a PhD student at MdH and at the Faculty of Electrical Engineering and Computing (FER), University of Zagreb, Croatia.

Website: http://www.es.mdh.se/staff/192-Juraj_Feljan

Contact: juraj.feljan@mdh.se

2.4 Project vision

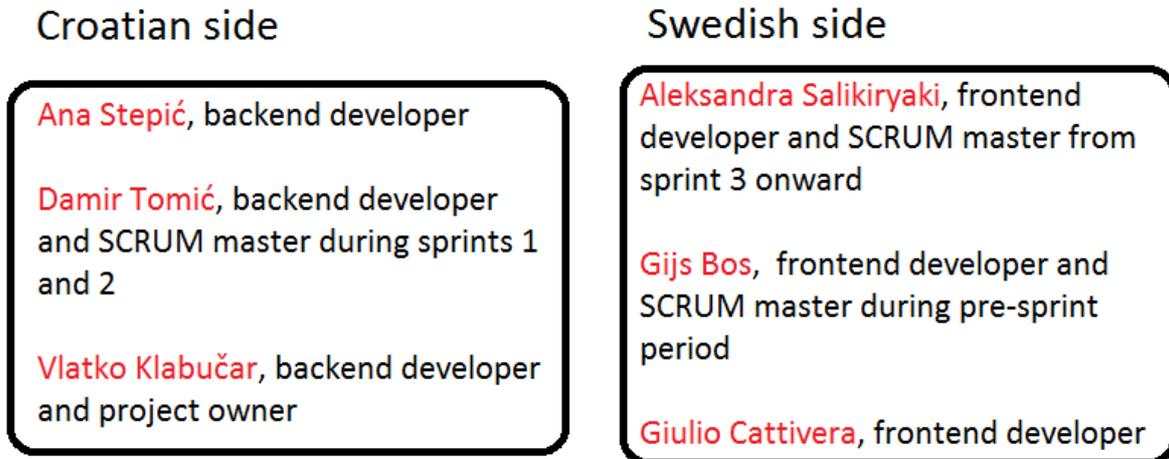
The vision of the project is to develop a recommender system. The recommender system should recommend the gifts to the user for his friends based on their preferences. The preferences of the users can be fetched from social networks, and various gifts can be queried and purchased from online services. This recommender system consists of mobile application and a back-end. Application servers as a front-end for the user to interact with the system. Back-end connects to social media and other content providers to fetch data about the users, gifts etc.

3 Organization and Communication

3.1 Team organization

Our team is organized according to the SCRUM methodology. We had a SCRUM master role, which we decided to rotate every few weeks. We had a product owner, in charge of communicating with the customer and accepting or rejecting requirements from our backlog. And finally we had a team of developers which were later organized in a remote Croatian and Swedish team to work on two separate parts of the application. The Croatian team was responsible for implementing the back-end including REST interface, Recommender system, and content provider implementation. The Swedish team was responsible for the front-end and graphical design of the application. The documentation was provided by both sides of the team, and was managed by our document manager to have a consistent format.

Figure 1 team responsibilities



3.2 Communication

Our primary means of communication consisted of having weekly Skype meetings. We intended to have daily stand-up meetings but we found it more convenient to have ad-hoc meetings whenever we needed to discuss a topic. We used Slack as a written communication tool, important decisions were reported on Slack so that we would have decent traceability of our project and decisions made.

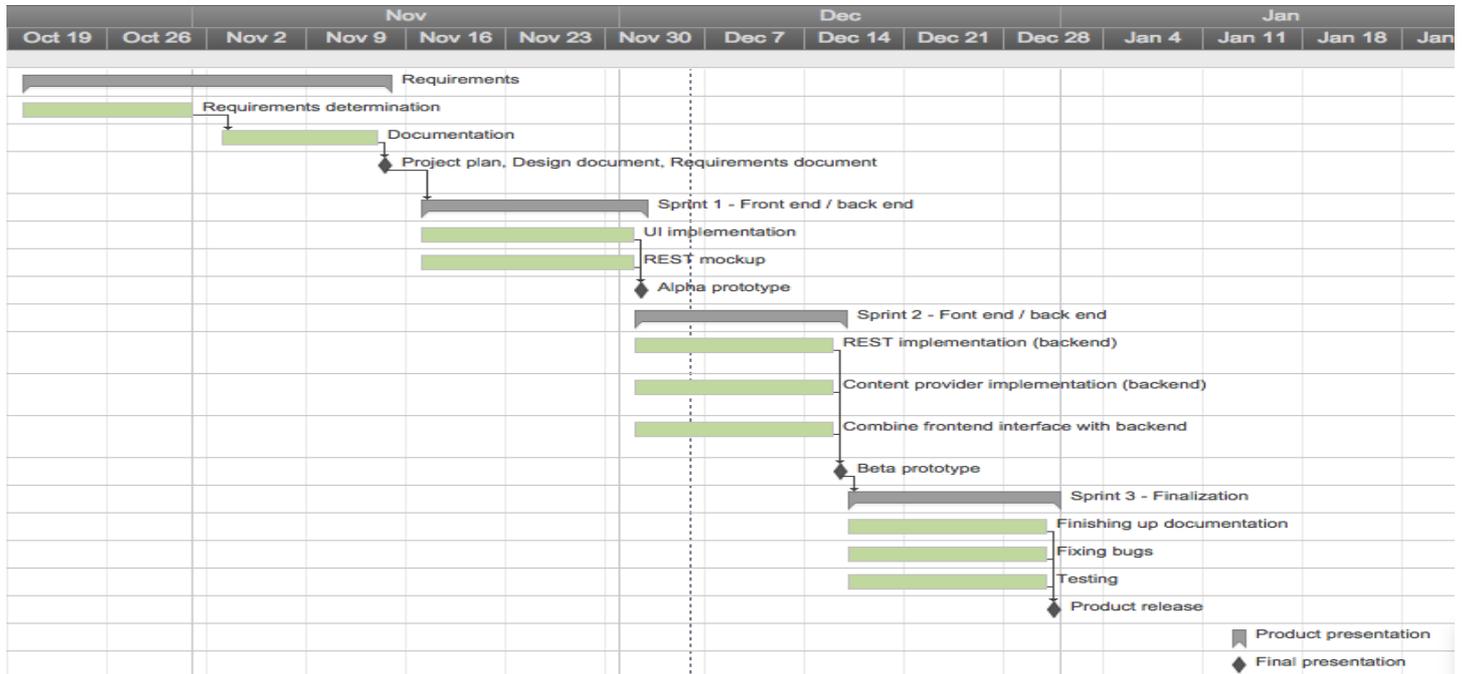
3.3 Team meetings

We had weekly meetings in which we talked about our current sprint process, the problems we encountered, and what could be improved for the next sprint. We had a modified version of a stand-up meeting in which we had a retrospective of the past activities, and what plans we had for future activities. We wrote down what we discussed in our minutes of meeting and made sure we completed the required documentation. Our meetings we're held with Skype.

4 Development Process

As shown in our Gantt chart, every sprint had several milestones which we aimed to achieve after every sprint. The sprints are described in detail in the sections below.

Figure 2. Gantt chart



4.1 Requirements gathering

Since the formation of our team, we appointed Vlatko to be our product owner. He was in charge of the communicating with the customer and communicating the requirements to the team. Before our first sprint, we had a video conference meeting with our customer and supervisors to get an idea of what the application is supposed to do. We wrote down our initial requirements and drafted follow up questions when matters were uncertain.

Vlatko had several face-to-face meetings with the customer and managed to draft a list of requirements with the help of the team. The initial requirements were agreed upon and we were ready to fill the backlog and plan our first sprint. The product of this phase was a Requirements Definition Document.

4.2 First sprint

The first sprint was all about establishing the most fundamental components required for the system. Our architecture was established in our architecture design process and we came to conclude that we could develop the back end and front end simultaneously. In order to prevent working too much apart from each other while working on both, we decided that we wanted to have an early back-end front-end integration. This resulted in a REST interface mock up that could process fake rest calls from the front end. After the first sprint, we had a basic REST interface and processed some dummy rest calls from the front end. Also a RESTful service TelcoMockup was implemented using S-CASE MDE tool, a tool for generating Java REST services. TelcoMockup was successfully deployed on Heroku.

Swedish front-end team set up the skeleton of the application and the MVC pattern structure. also the first rude version of the GUI was initialized and implemented.

4.3 Second sprint

The second sprint continued in developing the REST interface and we started to produce the engine to recommend gifts to the application users. We implemented several service and content providers such as amazon and steam. Also fetching TeloMockup data was implemented. The first interaction between backend and fronted was established in this sprint. During this sprint we experienced several problems, most notably inability to gather Facebook user information and Amazon security enforcement. It turns out that the Facebook is very restrictive about the data of it's users. If the application wishes to gather the information like email and user preferences it must submit a formal description of the application for review. Since this was a serious risk, considering that this review could go on for weeks and the grant of the permissions to gather user data was questionable we found a way to gather data without the delay. If the user has a confirmed Facebook account and is a Facebook developer the application can gather its data. During the development of the backend we used certain credentials to access the Amazon services, and they were hardcoded as plain text in our code. Amazon found out that our credential were exposed on a public github repository so we had to move them in the database.

The swedish team has implemented the login view, main view, contact view, and the events view. Implementation of the mechanism to read from the REST API. Mockup of the Facebook login. Implementation of the GiftBox view, inbox and outbox view, profile view, and gift details view.

4.4 Third sprint

We continued our front-end back-end implementation and integration with the content providers, and Damir finished implementing the recommendations engine. The Acceptance Test Plan document was created and the design of the front-end was being continued. Since the front-end was not ready yet, we decided to have another sprint.

The Swedish team did bug fixing and implemented the downloading gift functionality and the buying functionality. The Facebook integration was improved and the log out functionality was included in the application.

4.5 Fourth sprint

The fourth and last sprint was dedicated to debugging, and making final adjustments to the application. The front-end was finalized and the GUI was implemented during the last sprint. Several test have been run on mobile devices to make sure the application was consistent with the results from the emulators used to test the application.

The send invitation functionality was developed, filtering functionality and sorting functionality was developed, show contact gender functionality was developed. We performed the last code testing as well as usability testing.

5 Milestones

5.1 Internal

Our milestones consist of:

- Project documentation
 - Project plan
 - Design document
 - Requirements document
- REST mockup
- REST implementation
- UI Implementation
- Content provider implementation
- Front-end back-end integration
- Alpha testing

5.2 External

Our external milestones consisted of providing our customer with the following:

- Requirements document
- Alpha test
- Beta test
- Acceptance test
- Product release

6 Project Results

All members of the team got positive experience from this project. We learnt how to work on a distributed team, and especially how to handle communication and organizational issues when the team is unable to meet in person and is distributed in different part of the Europe. At the beginning we had some problems regarding relations among team members that could have lead several issues, but after some clarifications and discussions during the burn in phase we didn't encounter any particular problems.

6.1 Software overview

We learnt how to create a distributed system, since we developed a back-end and a front-end that are deployed in different machine. Moreover we also managed how to integrate a COTS in our system, since we used also a back-end-as-a-service.

6.2 Backend

The backend consists of a REST server bundled with the recommendation engine. It had been integrated with a number of service and content providers , like Amazon, Steam and Facebook. The backend was written in C# using a .NET framework. Its code is available on github and is currently deployed on Microsoft's Azure cloud.

6.3 Frontend

The application provides all the major and almost all the minor requirements. It offers gift recommendations, events proposals, integration with Facebook account, downloading of

received and sent gifts, many features of sorting and filtering of the results. We deployed the application for Android but since we used backbone it is also possible to deploy for different platform. All the features of the application are offered to the end user by a simple and usable GUI.

6.4 Missing features

They are all minor requirements, implement "Update item status" when certain present has been received or reviewed by its receiver. Implement highlighting of the received presents which have not been reviewed until now. Implement highlighting of the events which have not been reviewed until now. Show presents that a contact has received. Implement notification configuration as one of the buttons in the configuration menu and make a screen in which the count of notification and the amount of time before an event in which the notification should be sent.

6.5 Possible improvements

Back-end

The back-end could be improved by caching of user data and the data of content providers relevant to the users.

Front-end

Since the customer had a primary interest in the functional requirements, topics such as the look of the application was of less priority. The GUI of the interface could be improved to make the application look more appealing to the user.

The user interface of the application does not adjust when different resolutions are presented. Moreover, since we didn't have time to do a formal testing like whitebox testing then there could be some bugs that we were unable to find during our manual testing of the features. Improve the functioning of the sorting and filtering features, the behavior sometime is not precise.

6.6 Deliverables

Alpha prototype: 2 December 2014

Beta Prototype: 16 December 2014

Final version v1: 13 January 2015

Final version v2: 23 January 2015

6.7 Documentation

During the GiftCase project, the following documents were produced:

- Project plan: containing organizational aspects, and approach to the project
- Requirements document: contains the requirements of the application
- Design document: contains details about the architecture and application design
- Acceptance test document: contains test about the requirements and functionality
- User documentation: contains instructions about how to use the final application

At a regular interval the following documents were produced:

- Weekly report: information about the work done in the past time
- Minutes of meeting: information and agreements gathered during a meeting

7 Project Risks

The risks we identified were either related to our distributed team or the development of the application.

Team related risks:

- Differences in daily occupations
- Planned exams in both locations

Project related risks:

- Bad integration front-end and back-end
- Sprint process delays because of different occupations

8 Project Experiences

8.1 Environment

The distributed environment was a challenge in itself. Our team members were located in Croatia and Sweden. In terms of communication, we had to invest some effort into fine tuning our ways of communication with keeping cultural differences in mind. As soon as we established a good understanding of our working environment, we managed to become very productive in both development and communication.

8.2 Technologies

Beforehand we decided to use a variety of tools based on the experience of our team members. These consisted of:

- Slack for written communication: we experienced this tool to be very efficient because it allowed us to keep track of every major decision we made.
- Skype for vocal communication: we discovered that the best way of getting the team up to speed on current developments was by having Skype conference calls. This allowed us to quickly update every team member at the same time without the delay that could be found into reading messages on slack.
- Google drive: this tool provided us with a working environment for sharing and documenting project related content. We experienced a very efficient work style because Google drive allowed us to work on documents together.
- Kanbanize: this tool allowed us to create a backlog and prioritize our tasks and assign them to individual team members. This tool was helpful since we decided to use SCRUM.
- OneDrive: Microsofts OneDrive allowed us to create online presentations. This tool was somewhat slow but it got the job done.
- Visual studio: This developers tool provided us with the means of developing software fast.
- Sublime text 2: This developer tool is a simple text editor with several features to speed up development.

- Github: Github was essential to our distributed development process. This allowed us to work on the same code. We decided to split the project into two master branches which we all found very convenient.

9 Metrics

9.1 Work per member

Our work per member has been carefully documented in the weekly reports which clearly state every activity and effort made to complete the activity. An overview of the work per member is provided in the following subsections that consist of a table of hours worked per week and an overview of programming activities, captured by Github statistics.

Table 2. Hours worked per week

Member	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	Total
Gijs Bos	8	5	8	10	8	23	11	15	10	11	14	9	131
Giulio Cattivera	8	5	7	8	11	21	13	15	14	8	2	8	120
Vlatko Klabučar	8	10	8	8	9	12	6	13	10	11	7	21	123
Aleksandra Salikiryaki	8	5	10	10	6	23	15	21	7	10	12	26	153
Ana Stepić	8	7	7	22	6	19	15	14	12	10	10	26	156
Damir Tomić	8	8	7	9	12	11	16	16	10	12	9	26	144
Total	48	40	47	67	52	109	76	94	63	62	54	115	827

Figure 3. Frontend commits on Github

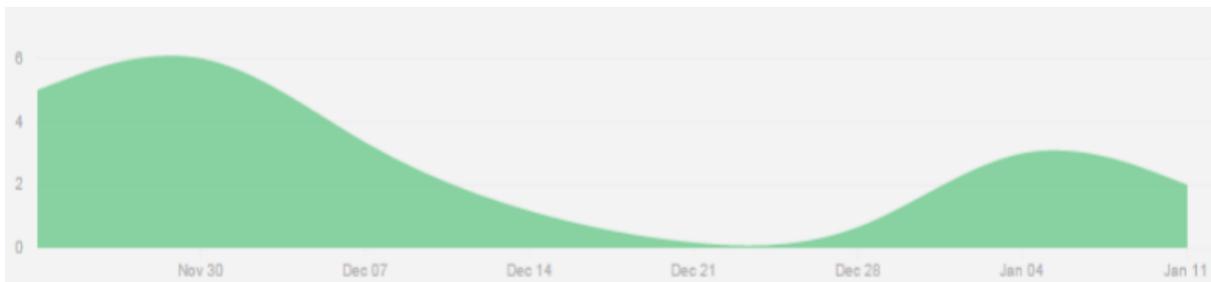


Figure 4. Backend commits on Github

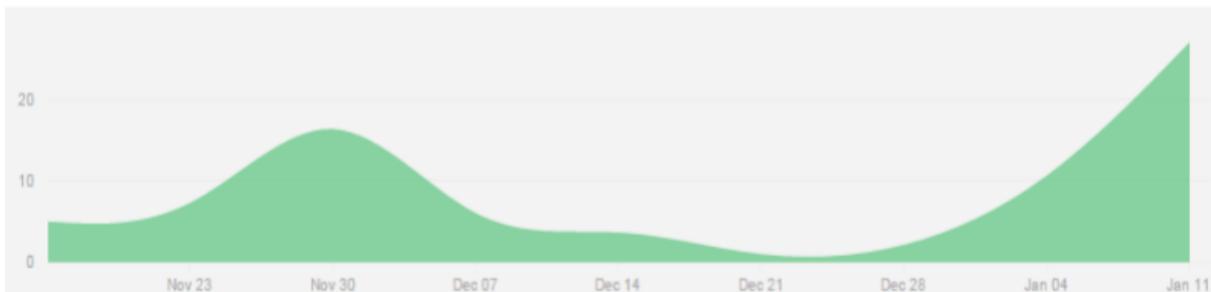


Figure 5. TelcoMockup commits on Github



Figure 6. Ana's commits on Github



Figure 7. Vlatko's commits on Github

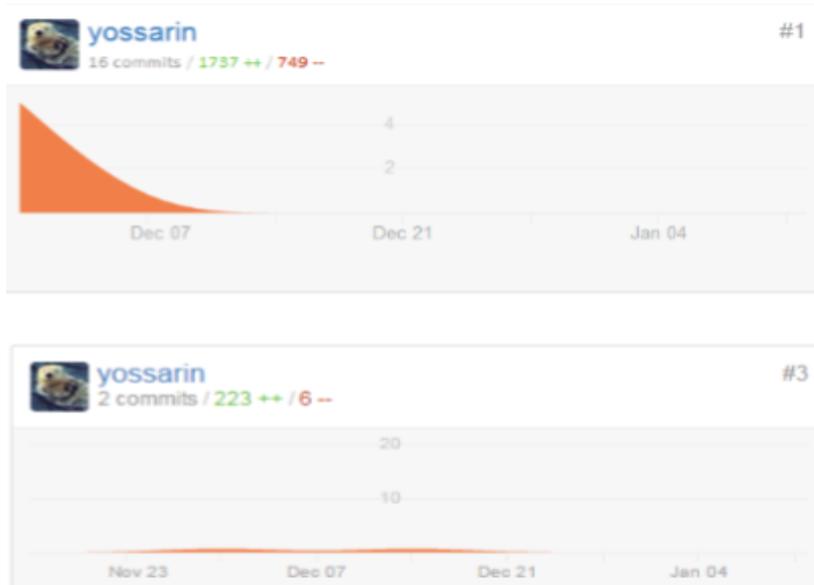


Figure 8. Damir's commits on Github



Figure 9. Alexandra's commits on Github



Figure 10. Gijs's commits on Github

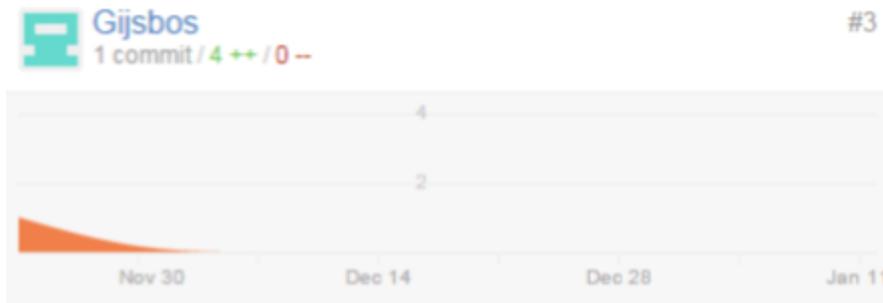


Figure 11. Giulio's commits on Github



9.2 Milestone metrics

Every sprint resulted into a milestone and deliverable. The deliverable was used as a metric of our milestone process. Our customer approved of major releases and every iteration of our product was considered to be a milestone metric, in terms of overall progress.

9.3 Effort metrics

The effort was measured by keeping track of the hours worked by our team members. We obtained statistics from Github to measure effort in terms of programming. And we kept track of working hours per week by filling in the weekly report. This weekly report described in detail what tasks were performed and how much time was estimated in order to complete the task followed by the actual time spend of completing the task.

References

- GiftCase Project - <https://www.fer.unizg.hr/rasip/dsd/projects/giftcase>
- Website FER - <https://www.fer.unizg.hr/rasip/dsd/>
- Website MDH - <http://www.mdh.se>
- Website Polimi - <http://www.polimi.it/en/>
- Requirements Definition Document - http://www.fer.unizg.hr/_download/repository/Requirementsdefinition.pdf