Taxi Service
Final Project Report

Version 1.0

Doc. No.:

Revision History

Date	Version	Description	Author
2013-01-20	0.01	Initial Draft	Luca Zangari
2013-01-20	0.02	Chapter 2, 5, 6 edited	Luca Zangari
2013-01-20	0.03	Chapter 8.2, 9 edited	Luca Zangari
2013-01-20	1.0	Chapters 8.2, 9.2, 9.3 added General review	Lyudmil Angelov

Table of Contents

1. Introduction

1.1 Purpose of this document

The purpose of this document is to display an overview of the results and metrics of the Taxi Service project, as performed during the Distributed Software Development course 12/13. This course is joint course between Politecnico di Milano University (POLIMI) in Italy and University of Zagreb (FER) in Croatia.

This document provides information about the Taxi Service project, global team members and their performance.

1.2 Intended Audience

Following stake holders are the intended audience for this document

- Team members
- Supervisor
- Customer
- Future developers

1.3 Scope

This document covers only the result of the project. It will not cover any assumptions made in the beginning of the project, and will barely cover differences between the assumptions and results.

1.4 Definitions and acronyms

1.4.1 Definitions

Keyword	Definitions

1.4.2 Acronyms and abbreviations

Acronym or abbreviation	Definitions

1.5 References

This is a summary of all documents uploaded under the Final Documentation section on the following website: http://www.fer.unizg.hr/rasip/dsd/projects/taxi_service/documents

2. Background and Objectives

The main goal of this project is to develop functional Taxi Service system. The system will consist of several basic elements:

- Main server
- Android Client application for taxi drivers
- Android Client application for customers
- Web Application for customers

Customers will be able to order a taxi to a position via an Android mobile application or a web interface. After the order has been received, the main server will determine the zone from which the order has been made and then select the first taxi from the virtual queue of the appropriate zone to dispatch to the order location. The customer will receive various information about the taxi which will pick him or her up (position in the map, name estimation time of arrival), as soon as the driver of the taxi accepts the order from the main server through their custom Android application. If the taxi driver rejects the order or doesn't respond to it in a certain amount of time, the taxi will be put at the end of the queue and the order will be forwarded to the next taxi in the queue. While driving through the city, the taxies will change virtual queues as they change zones they are driving in. The taxi will be removed from the old queue and put at the end of a new one.

The integration strategy of the system will be feature – based. The development will begin with the core functionality and new features will be added with time. There will be several milestones and new features will be introduced in each of those. After the feature is developed, first it will be tested standalone and then it will be integrated in the system. After the integration, new series of testing will take place. After the system is fully developed and tested, it will be delivered to project supervisor in 3 parts: Web application for server, Android client application for Taxi, and Android client application for costumers. The Web Application is an extra feature delivered after the beta prototype, it allows the customer to book a taxi from a web page after registering and covers all functionalities offered from the taxi client except of course the GPS localization (the customers has to choose manually the point meeting point). The system software will be followed with the necessary project documentation.

3. Organization

Although all team members are enrolled to one of the two universities (FER and POLIMI), the team is actually geographically divided in three locations:

- Croatia (5 team members)
- Italy (2 team members)
- Finland (1 team member)

The work on the project is divided in three categories: Organization, Documentation and Presentations, and Implementation. It is decided that all team members equally participate in every project part.

1. **Organization**
    - **Project leader**
      Project leader is responsible for the team in general. His responsibility is to always be informed about every important issue. His responsibility is also to inform others about those issues. He should also be monitoring the work of POLIMI students.

    - **Team leader**
      Team leader's responsibility is to monitor FER students and inform team leader about important issues that are taking place on Croatian side.

    - **Others**

All team members share responsibility of organizing internal meetings, meetings with the project supervisor, dividing project tasks and delivering documents on time.

Tools: Google groups, Skype, Google calendar, Doodle

2. **Documentation and Presentations**
Documentation and Presentations are both responsibility of every team member. Every document that is required to deliver is entrusted to several team members (number depends on the document). After they write the document, other team members should check it and make corrections if necessary. The content of the documents is discussed on weekly meetings.
Presentations should be made by team members who are going to present them, and checked and corrected by other team members. It is agreed that two or more team members will be presenting.

Tools: Google docs, Dropbox, SVN

3. **Implementation**
Since the project is divided in three major parts, the project roles are defined similarly:

- **Taxi Mobile Application developer** (2 team members)
Responsibility: developing mobile application that will be used in Taxis.
Communication: with server side developers

- **Client Mobile Application developer** (2 team members)
Responsibility: developing mobile application that will be used by clients who want to order a Taxi
Communication: with server side developers

- **Server side developer** (3 team members)
Responsibility: developing a web service which will be the communicating with mobile applications, developing web application for clients who want to order a Taxi
Communication: with mobile application developers

- **GUI developer** (1 team member)
Responsibility: developing graphical user interface for mobile and w
Communication: with mobile and web application developers

Tools: SVN, Trello

3.1 Project group

Name	Initials	Responsibility (roles)
Luca Zangari	LZ	• Project leader • Taxi Mobile Application development
Leon Dragić	LD	• Team leader • Client Mobile Application development
Lyudmil Angelov	LA	• GUI development • Helping other team members when needed
Marko Coha	MC	• Client Mobile Application development
Jelena Jerat	JJ	• Server side development
Fabio Kruger	FK	• Taxi Mobile Application development
Igor Piljić	IP	• Server side development
Karlo Zanki	KZ	• Server side development

3.2 Customer

Comune di Milano (eng. *City administration of Milan*)

3.3 Supervisor

Prof. Elisabetta Di Nitto (POLIMI)

3.4 Others

Prof. dr. sc. Mario Žagar (FER)
Prof. Ivica Crnković (MDH)
Prof. Raffaella Mirandola (POLIMI)

Professors from FER, POLIMI and MDH responsible for the DSD course.

Marin Orlić (FER) – SVN administrator, Virtual machine administrator

4. Development process

We follow a modified SCRUM methodology for Taxi Service. We do project planning on the milestone level and deliver the project on a feature-by-feature basis.

Planning and Delivery Schedule

Planning is done by defining milestones and calculating the time to deliver each.

To build a milestone, first we break down the problem into vertical features, meaning things that make sense to the users of the system. So, for example, since implementing a part of the database on the server is not something that would be visible to any of the stakeholders of the project, it isn't considered a feature. An example feature is "A taxi reports its current location to the central server continuously," implying that user interface and back-end work need to be completed on both the taxi device and the server and integrated before it would be considered done.

Once we have a set of features that covers the functionality we want to cover in the next milestone, we estimate the complexity of each feature in complexity points, assigning an integer value between one and three. The complexity measure is only relative, so a lower score for Feature A compared to Feature B means that Feature A is relatively simpler to implement than Feature B.

When all milestone features are estimated, we sum up their complexity values to get the total complexity of the milestone. We then use our current velocity (measured in complexity points per week) to estimate how long it will take to complete the milestone (total complexity divided by velocity).* The result is a release schedule.

* The current velocity is measured throughout the project, averaging the velocities of past iterations. The initial velocity (for the first milestone) is a matter of agreement between the team members. In this case, we have picked a target initial velocity of two complexity points per week.

Development Process

Development is done on a feature-by-feature basis.

Once we have scheduled a milestone, we begin work on it in weekly iterations (what is commonly known within SCRUM as "sprints"). Our current velocity provides us with an easy way to calculate the capacity of the team for any given week (velocity multiplied by iteration length in weeks). At the beginning of the week we pick enough features to work on to fill that capacity and start developing. We implement, test, and integrate each component throughout the week in an ad-hoc manner. At the end of the week, we check how many of the features we have worked on are completed, show them to our product owner, and sum up their complexities to give us our new current velocity. We then check to see if that velocity keeps us on track to finish the milestone on time and make appropriate adjustments (simplifying features, adding more features, postponing the milestone delivery date, etc.). We then move on to plan the following iteration, repeating the same process until all the features of the milestone are completed.

Once a milestone is completed, we start again, building a list of features that will constitute the following milestone. This process continues until the final delivery deadline for the course.

Project Roles

1. Product owner: Prof. Elisabetta Di Nitto

The development team with her in order to define requirements and features. She also reviews and signs off on each feature the development team delivers.

2. Scrum Master: Luca Zangari

The development team communicates obstacles and difficulties they are experiencing that prevent them from doing the work required to deliver the work on time and he tries to remove said impediments.

3. Development team: Luca Zangari, Fabio Kruger, Karlo Zanki, Leon Dragić, Igor Piljić, Marko Coha, Jelena Jerat, Lyudmil Angelov

The development team members gather requirements, design, implement, test, and integrate features.

5. Milestones

Id	Milestone Description	Responsible Dept./Initials	Finished week				Metr	Rem
			Plan	Forecast		Actual		
				Week	+/-			
M-001	Project Vision Presentation	LZ / LA	43	43	0	43		
M-002	Project Plan	LD / JJ / LZ	44	44	0	44		
M-003	Project Plan Presentation	LD / JJ	44	44	0	44		
M-004	Requirements definition	KZ /LZ /LD /JJ	44	44	0	44		
M-005	Design Description	LA	45	45	0	45		
M-006	Alpha prototype	All team members	47	47	0	47		
M-007	Beta prototype	All team members	50	50	0	50		
M-008	Acceptance Test Plan	All team members	52	52	0	52		
M-009	Test Report	All team members	02	02	0	02		
M-010	Final Product	All team members	02	02	0	02		

6. Project Results

6.1 Requirements

6.1.1 Requirement Compliance Matrix

Id	Requirement Description	completed	Rem
TAXI-1	Communicate with the server side, sharing constantly current position and status	Yes	R-001
TAXI-2	Receive from the server side the booking requests, accept or decline it	Yes	
TAXI-3	Receive information about the booked destination	Yes	
SERVER-1	Receive constantly the taxi positions and statuses	Yes	
SERVER-2	Divide the city in different areas and make a queue for every area	Yes	R-002
SERVER-3	Maintain the queues up to date with the taxi positions and statuses	Yes	
SERVER-4	Receive orders from customer clients, and dispatch them to the nearest queues	Yes	
CUSTOMER-1	Communicate with the server side, sending a booking request of a taxi	Yes	
CUSTOMER-2	Receive information about the booked taxi and its time of arrival	Yes	

Completed: Yes (completely implemented)
No (not implemented at all)
Partially (partially implemented, more description under Remarks subsection)
Unknown (completion status not known)
Dropped (requirement was dropped during the course of the project)

6.1.2 Requirements Compliance Summary

Summarize the requirements compliance data.

Total number of requirements	9
Number of requirements implemented	9
Requirements partially fulfilled	0
Requirements not fulfilled	0
Requirements dropped	0

6.1.3 Remarks

Remark Id	Description
R-001	We have chosen to send the taxi position every 30s because of the taxi needs to be localized constantly, in any case this time can be changed as a variable.
R-002	We have chosen to divide the city in different area corresponding to zip codes zone about 100 in Milan area.

6.2 Deliverables

To	Output	Planned week	Promised week	Late +/-	Delivered week	Rem
Steering group	Project vision presentation	W43	W43	0	W43	
Steering group	Minutes of meeting documents	*	*	*	*	1
Internal	Week reports	*	*	*	*	2
Customer	Project plan document	W44	W44	0	W44	
Steering group	Project plan presentation	W44	W44	0	W44	
Steering group	Requirements definition document	W44	W44	0	W44	
Internal	Documentation policy	W45	W45	0	W45	3
Internal	Coding policy	W45	W45	-1	W44	3
Internal	SVN policy	W45	W45	-1	W44	3
Internal	Interfaces definition document	*	*	*	*	4
Steering group	Requirements definition and system Architecture presentation	W45	W45	0	W45	
Steering group	Design description document	W45	W45	0	W45	
Steering group	Alpha prototype presentation	W48	W48	0	W48	
Steering group	Beta prototype presentation	W51	W51	0	W51	
Steering group	Acceptance test plan	W01	W01	0	W51	
Steering group	Test report	W03	W03	0	W03	
Steering group	Final product presentation	W03	W03	0	W03	
Steering group	Final project report	W03	W03	0	W03	
Steering group	User manual	W03	W03	0	W03	
Steering group	Installation manual	W03	W03	0	W03	
Customer	Final product	W03	W03	0	W03	

7. Risks

Look at the risk table from the Project Plan document and list and comment:

- *risks that have appeared but their impact was low because of preventive actions*
- *risks that appeared and had a significant impact on project work*
- *risks that appeared but were not foreseen and listed in the table (describe them and their impact on project work)*

8. Project Experiences

8.1 Positive Experiences

Write down what went well during the project work, be very specific – what, how, why!

This project generated a lot of positive experiences, which are hard to rank. One of probably the greatest was to work in a multinational team. The team members came from four different countries:

- Croatia
- Italy
- Bulgaria

It was very interesting to see how people from different cultures think and what their working habits are. Except for Croatia with five team members, there were two representatives from Italy and one for Bulgaria. Because of that it's not possible to bring general conclusions about the cultures, but still it was a nice experience to work in such an environment.
The distribution of the members of the team was interesting because one of guys from Milan was in Finland, so we had 5 people in Croatia, 2 in Milan and one member of the team in Finland.

In our project the way to distribute the work was taking that we considered the members' previous technology experience and skills. So in our case we had two members from Croatia and two members from Italy working on the Android, and tree member in Croatia working on the server side. This way, most of our communication happened online in Skype meetings, instead of in person, as it would in any other usual project and we think we handled it very well and gained some precious experience in that field.

One of the most important positive things about this project is that we experience the work on a "real" project and we have seen how our applications can work in a real context with a lot of users and taxi clients connected simultaneously to the server.

8.2 Improvement Possibilities

Initially, it was difficult to get on the same page or even identify that there was a misunderstanding taking place. Ultimately, we had to seek help from Prof. Di Nitto in order to be able to work productively.

There is very little that can be done to prevent these problems from occurring, but they were exacerbated by the fact that we were under pressure to deliver on a tight schedule and by permitting ourselves to react emotionally to the situation. Therefore, in order to better handle such situations in the future, we would think about building in slack into the schedule initially to eliminate deadline pressure and we would be sure to talk explicitly about how to resolve conflicts before starting work.

Another idea to try would be weekly process improvement meetings where team members are encouraged to share their grievances and issues and ask the team to help resolve them together. This is a popular technique on Agile teams and it provides a structured and orderly way to address any issues in a timely manner.

9. Metrics

9.1 Work per Member

.

Member	W43	W44	W45	W46	W47	W48	W49	W50	W51	W52	W01	W02	W03	Total
Luca Zangari	13	11	10	10	16	8	11	14	8	4	3	6	13	127
Leon Dragić	15	15	13	24	16	8	15	12	7	6	6	29	13	179
Lyudmil Angelov	13.5	18.5	20	10	4	9	16	13	4	3	0	15	4	130
Marko Coha	11.5	9	9.5	14	12	12	9	4	5	2	1	0	10	99
Jelena Jerat	15,5	18	22	22	8,5	13	11	19,5	7	7	1	7	10	161,5
Fabio Kruger	10,5	10	7.5	10	20	10	10	14	9	0	7	6	6	120
Igor Piljić	11	13	18	26	8	14	13	12	9	4	1	9	10	148
Karlo Zanki	9,5	8,5	12	19	15	15	10	10	6	2	1	3	10	121
Total	99,5	103	112	135	99,5	89	95	98,5	55	28	20	75	76	1085,5

Figure below shows the distribution of total working hours among the group members. Here, it can be seen that we didn't have extreme cases of one person doing most of the work or a person that's not doing anything.

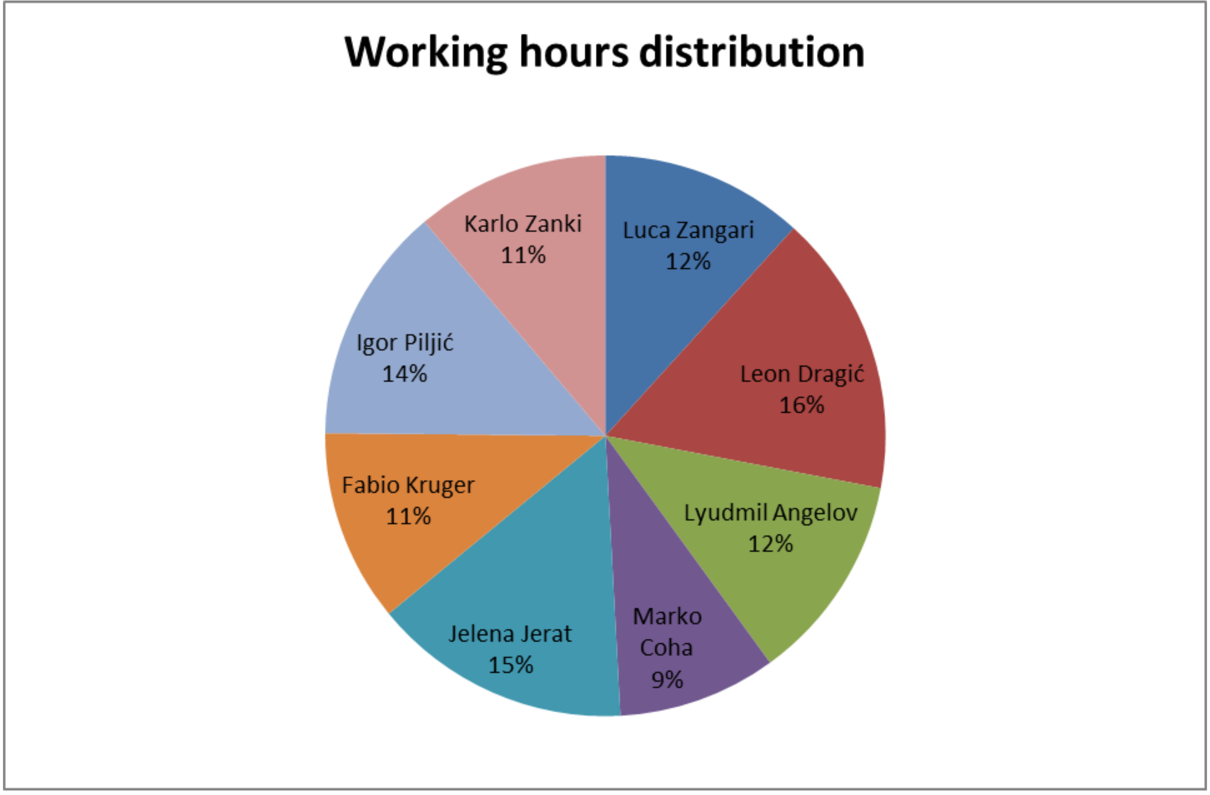
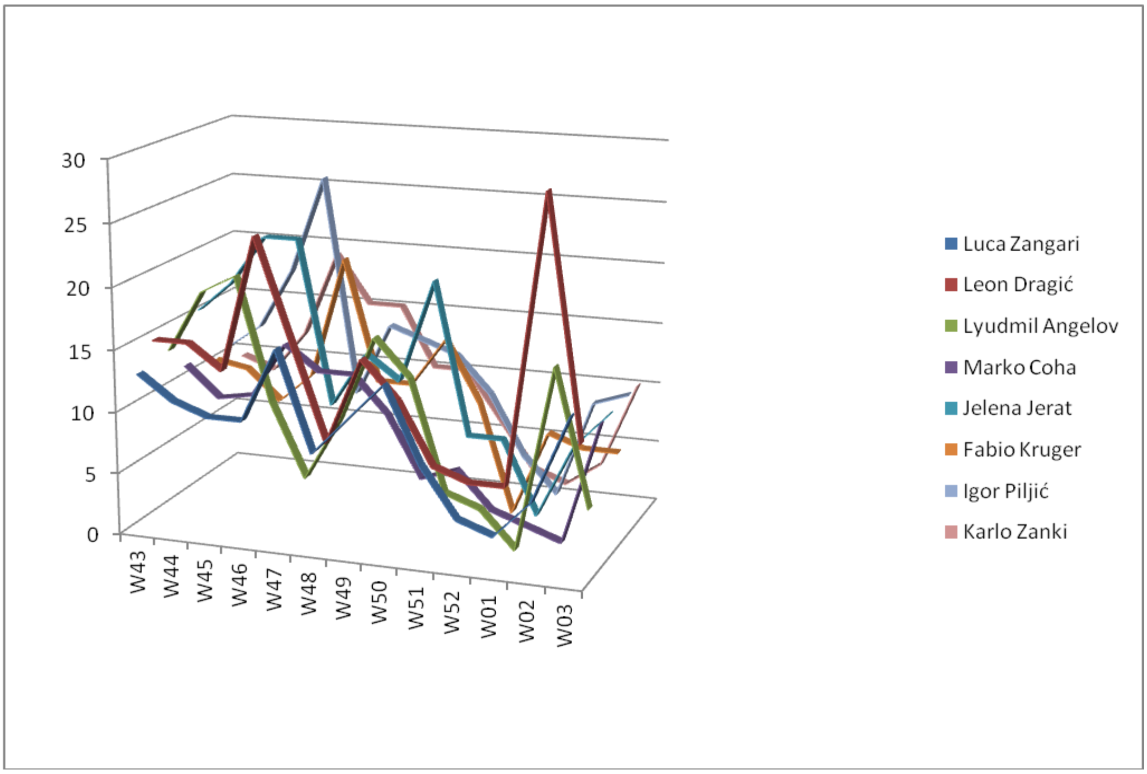


Figure below shows how the total working hours were growing during the project for each team member. It can be seen that before the important milestones the growth is steeper at most of the team members (Alpha , Beta prototype and Final product).



9.2 Milestone Metrics

Completed as planned or earlier	Total	Timeliness
9	10	90%

9.3 Effort Metrics

ID	Activity	Actual Effort	Planned Effort	Deviation (%)
M001	Dividing the tasks	7	7	0
M002	Gathering requirements	7	7	0
M003	System architecture	3	3	0
M004	Developed feature: A taxi notifies the server of its location continuously	11	7	63.6
M005	Developed feature: A taxi can get its zone information from the server	7	7	0
M006	Developed feature: A taxi can change its status	3	3	0
M007	Developed feature: A customer can place an order for a taxi	15	7	46.6
M008	Developed feature: A taxi can receive a customer order	7	7	0
M009	Developed feature: After the taxi is selected, customer gets taxi info	7	7	0
M010	Optional functionality	7	7	0
M013	Final product	7	7	0

Effort estimation accuracy (%) *(100*(1 - abs(Actual – Planned)/Actual))*	94.8%

For our project, measuring effort in workdays is not a great way to determine estimation accuracy. We measured effort in terms of relative complexity, as explained in Chapter 4. The metrics above are extrapolated from the historical development velocity of the team, so the final numbers are still meaningful, but getting a number that truly measures the team's performance during estimation meetings is a more involved and complex process.

The deviation in milestones M004 was due to team dynamics outlined in Chapter 8.2, which caused a lot of duplicated work. Milestone M007 was the most complex and risky feature, so it should not be a surprise our estimate was off.