**Taxi Service
Project Plan**

Version 1.2

Revision History

Date	Version	Description	Author
2012-10-24	0.1	Initial Draft	DSD staff
2012-10-26	0.2	Chapters 1, 3, 7 Draft	Jelena Jerat
2012-10-27	0.3	Chapters 1, 8 Draft	Jelena Jerat
2012-10-27	0.4	Chapter 2 Draft	Leon Dragić
2012-10-29	0.5	Chapter 4 Draft	Lyudmil Angelov
2012-10-31	0.6	Chapter 9 Draft	Igor Piljić
2012-11-01	0.7	Chapter 5, 6, 10 Draft	Jelena Jerat
2012-11-02	1.0	First version finalized	Jelena Jerat
2012-11-13	1.1	Time tables updated to week 46	Marko Coha
2012-12-04	1.2	Time tables updated to week 49, updated chapter 9	Jelena Jerat

Table of Contents

1. Introduction

1.1 Purpose of this document

This document presents a detailed project plan of a Taxi service project. This document is defined at the beginning of the project and will be revised whenever a change in project organization occurs. Any change in deadlines, documents or parts of the program that are supposed to be delivered will also be documented in one of the revisions. Every other delivered document should be based on the content of this document.

1.2 Intended Audience

Project plan document is intended for all team members as well as the project supervisor and the customer. Team members working on the project should use this document as a base document which defines work methodologies milestones and deliverables. It is supposed to be used and revised in every phase of the project work from its beginning until the end of the project. Customer and project supervisor should use it to get the insight in the project organization.

1.3 Scope

This document gives general information about Taxi service project. It describes a team organization and development process in general. In this document are defined all milestones and deliverables significant for the project as well as the project risks. However, detailed information about some parts of the project (such as implementation details and meeting details) will be given in other documents.

1.4 Definitions and acronyms

1.4.1 Definitions

Keyword	Definitions
Taxi service	Project name
Google docs	Web-based office suite and data storage service offered by Google
Dropbox	A cloud file hosting service that offers file sharing and synchronization.
Web service	A method of communication between two devices on the Internet
Doodle	A service that facilitates scheduling meetings.

1.4.2 Acronyms and abbreviations

Acronym or abbreviation	Definitions
PL	Project leader
TL	Team leader
GUI	Graphical user interface
POLIMI	Politecnico di Milano (eng. *Polytechnic University of Milan*)
FER	Fakultet Elektrotehnike I Računarstva (eng. *Faculty of Electrical Engineering and Computing, University of Zagreb*)
SVN	Subversion

1.5 References

Taxi service website: http://www.fer.unizg.hr/rasip/dsd/projects/taxi_service

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 customers
- Android Client application for taxi drivers

Customers will be able to order a taxi to a position via Android client application or web application. After the order has been received, main server will determine the zone from which the order has been made and after that, select the first taxi from the virtual queue of the appropriate zone and dispatch it to the received location. The customer will receive the number of the taxi which will pick him up, as soon as the driver of the taxi accepts the order he got from the main server. 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 every. 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 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 and design, implement, test, and integrate features.

5. 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
Intenal	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			
Steering group	Acceptance test plan	W01	W01			
Steering group	Test report	W03	W03			
Steering group	Final product presentation	W03	W03			
Steering group	Final project report	W03	W03			
Customer	Final product	W03	W03			

5.1.1 Remarks

Remark Id	Description
1	Minutes of meeting are delivered regularly after every team meeting
2	Week reports are delivered regularly every week on Sunday
3	Delivered on the Taxi Service shared folder on Dropbox
4	Interfaces definition document is defined on the first week of implementation (W45) but is constantly (weekly) updated as new features are being implemented. It's final version will be uploaded on the DSD website.

6. Inputs

From	Required item	Planned week	Promised week	Late +/-	Delivered week	Rem
Project supervisor	Requirements	W44	W44	0	W44	
DSD staff	SVN	W42	W42	0	W42	
DSD staff	VM	W44	W44	0	W44	
Project supervisor	Windows Azure license	W44	W44	0	W44	

Comment: *Required week* = week when it is required by the project; *Promised week* indicates when the *From* expects to deliver; *Late +* indicates a discrepancy between *Required week* and *Promised week; Received week* is week when it was actually received; *Rem* is a remark index number.

6.1.1 Remarks

Remark Id	Description

7. Project risks

Possibility	Risk	Preventive action
High	Problems with system integration	Well defined interfaces between components and constant communication between members who are developing connected components.
High	Lack of time to finish deliverables on time	Planning in advance, dividing work equally between team members to avoid overloading some team members.
Medium	The need for licensed tools (because of using commercial platforms and technologies)	Planning the project from start, elaborating requirements. Deciding on the tools at the beginning of the project and making sure all needed licenses are available.
Medium	Bad communication between team members	Regular weekly meetings, PL and TL coordinating the team.
Medium	Lack of motivation for working on a project	Constant communication between team members and solving problems together.
Low	Lack of knowledge in technologies	Choosing technologies that most team members are familiar with, and dividing the work to team members considering their knowledge.
Low	Losing work because of SVN error	Making regular backups
Low	Team member leaves the project	Making sure that PL and TL are always informed about project status. Making sure that always at least two team members work together on important parts of the project.
Medium	Final product doesn't meet the requirements	Regular contact with project supervisor, daily supervising of work of every team member

8. Communication

8.1 Meetings

8.1.1 Skype meetings

Every Thursday at 19.00 h the team discusses the project progress and important issues on a regular weekly meeting. Except the regular meeting, more meetings can be arranged during the week if the need for that arises. The time of irregular meetings is usually determined with Doodle.

Every meeting is entrusted to one team member who leads the meeting. This team member should be prepared for the meeting. He should decide on the topics that will be discussed and write the minutes of meeting document after the meeting is finished. The person leading the meeting is very important since the meetings are usually Skype voice calls which would be very hard to coordinate without the leader.

8.1.2 Meetings with project supervisor

Meetings with project supervisor are usually arranged via e-mail. They are arranged when it's necessary to discuss some important issues with the supervisor and they take place on Italian side. The whole team discusses the topic of the meeting and questions to be asked on Skype meetings. After the meeting with the supervisor Italian members of the team inform the rest of the team about it.

It is also considered to make those meetings regular once a week. Team members from Croatia should also be able to participate in those meetings using Skype voice call.

8.1.3   Meetings in person

Both the Italian and the Croatian side can sometimes have meetings in person in order to discuss the project. Meetings in person are not so often as other types of meetings.

8.2   Google group

Asynchronous communication is maintained via Google group. When the issue appears, the team member opens a new topic on the Google group.

8.3   Person to person communication

Person to person communication is very often and usually not planned. To communicate that way team members use instant messengers like Skype, Google talk and MSN messenger. Team members from the same country also use mobile phones for communication.

9.   Configuration management

SVN will be used for sharing and managing code. There will be a person in charge for administrating SVN repository. SVN administrator is responsible for making regular backups in case of SVN error. Since project is feature-based, decision has been made that every feature will be developed and tested in separate branch and integrated into trunk afterwards. Every member is supposed to follow SVN and code policies.

For sharing documents team has decided to use Dropbox where all formal and informal documents will be stored and Google Drive in case two or more people work on same document at same time.

The team will use Trello collaboration tool for project management. Trello will be used to help team members organize their work and monitor the status of the project.

For the purpose of this project the team will use:
- Windows Azure cloud service
- Trello collaboration tool

10. Project plan

10.1 Time schedule

Id	Milestone Description	Responsible Dept./Initials	Finished week				Metr.	Rem.
			Plan	Forecast		Actual		
				Week	+/-			
M001	Dividing the tasks	LZ, LD	W43	W43	0	W43	0	
M002	Gathering requirements	LZ,LA	W44	W44	0	W44	0	
M003	System architecture	Whole team	W45	W45	0	W45	0	
M004	Developed feature: A taxi notifies the server of its location continuously	IP, JJ, KZ, FK, LZ	W46	W46	0	W45	+1	
M005	Developed feature: A taxi can get its zone information from the server	IP, JJ, KZ, FK, LZ	W47	W47	0	W47	0	
M006	Developed feature: A taxi can change its status	FK, LZ	W48	W48	0	W48	0	
M007	Developed feature: A customer can place an order for a taxi	LD, MC	W48	W48	0	W49	-1	
M008	Developed feature: A taxi can receive a customer order	Whole team	W49	W49	0			
M009	Developed feature: After the taxi is selected, customer gets taxi info	IP, JJ, KZ	W50	W50	0			
M010	Optional functionality	Whole team	W51	W51	0			
M013	Final product	Whole team	W03	W03	0			

Comment: *Finished week: Plan* = original plan; *Actual* = week when it was actually available; *Forecast Week* indicates the estimated finished week; *Forecast +/-* indicates changes in the estimation compared to the previous project report; *Metr* is the difference between Plan and Actual, that is if Plan is w604 and Actual is w606, then Metr will be -2. To be filled in when the milestone is reported as completed; *Rem* is a remark index number.

10.1.1 Remarks

Remark Id	Description

10.2 Activity plan

	w43	w44	w45	w46	w47	w48	w49	w50	w51	W52	w01	w02	w03
Dividing the tasks													
Gathering requirements													
System architecture													
Developing Feature 1													
Developing Feature 2													
Developing Feature 3													
Developing Feature 4													
Developing Feature 5													
Developing Feature 6													
Adding Optional													
Unit Testing													
Acceptance Testing													
Documentation													
Delivering final product													

ID	Predecessor	Activity	Days	Md ays	Rem.
1		Dividing the tasks	1	1	
2		Gathering requirements	5	10	
3		System architecture	6	18	
4	1,2,3	Developing Feature 1	7	42	
5	1,2,3	Developing Feature 2	7	42	
6	1,2,3	Developing Feature 3	4	24	
7	1,2,3	Developing Feature 4	10	60	
8	1,2,3	Developing Feature 5	7	42	
9	1,2,3	Developing Feature 6	7	42	
10	9	Adding optional functionality	14	112	
11		Unit Testing	14	28	
12		Acceptance Testing	14	28	
13		Documentation	25	60	
14		Delivering final product	7	21	

Total planned calendar days for the project to be finished and total planned man-days required to finish the project

Planned effort (days)	Planned effort (man-days)
128	530

10.2.1 Remarks

Remark Id	Description
1	One working day is considered to be 4 – 5 h of work