myTaxyService

Project Plan Document

Version 1.0

Matteo Farè, Federico Feresini, Thierry Ebali Essomba

February 1, 2016

Contents

Chapter 1

Introduction

This document aims to state the Project plan for MyTaxiService system.

The document will be composed of these sections:

- Estimation of effort and cost, this will be done by the applying Functional points method and COCOMO to identify the tasks for the project and their schedule.
- The identification of the tasks needed for this application.
- Resource allocation where the tasks will be assigned to all the project members.
- Risk analysis, their relevance and their recovery actions.

Chapter 2

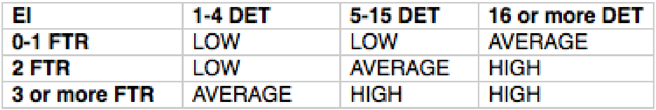
Effort and Cost Estimation

2.1 Functional Points

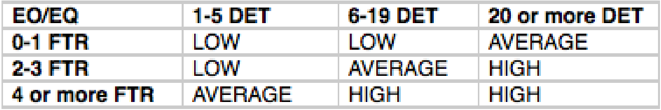
Functional points analysis considers as a good size indicator the number of different data structures that are used. The following table contains all the useful elements to perform the Function Point estimation approach

- External Inputs (EI), elementary input activities.
- External outputs (EO), elementary output activities.
- External inquiries (EQ), elementary interrogation activities.
- External Logic Files (ELF), external files to the application. Internal Logic Files (ILF), internal files to the application.

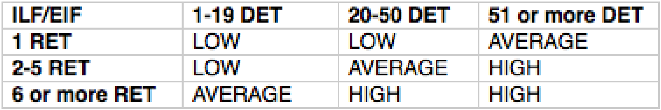
Each element is classified as low, average, high complex based on complexity table. These tables use different elements to achieve the classification: Data element type (DET), a user-recognizable field. File type referenced (FTR), an internal logic file which is read or held by functions, or an external file read by functions (EI/EQ/EO). Record element type (RET), a subset of user-recognizable inside a ILF or EIF.



EI	**1-4 DET**	**5-15 DET**	**16 or more DET**
0-1 FTR	LOW	LOW	AVERAGE
2 FTR	LOW	AVERAGE	HIGH
3 or more FTR	AVERAGE	HIGH	HIGH

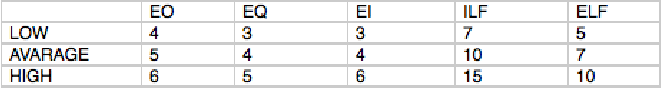


EO/EQ	**1-5 DET**	**6-19 DET**	**20 or more DET**
0-1 FTR	LOW	LOW	AVERAGE
2-3 FTR	LOW	AVERAGE	HIGH
4 or more FTR	AVERAGE	HIGH	HIGH



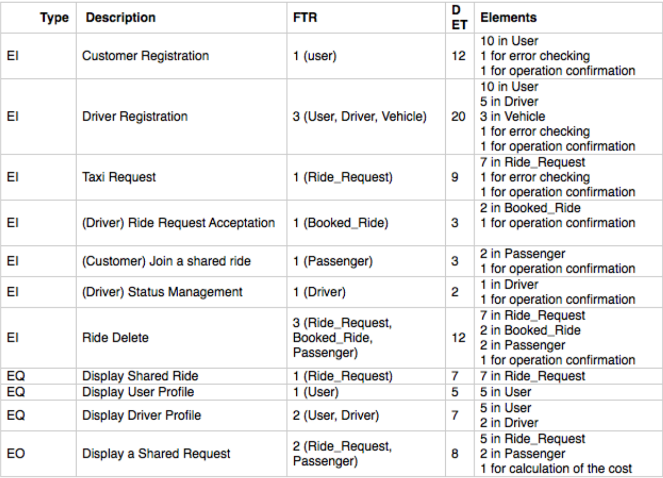
ILF/EIF	**1-19 DET**	**20-50 DET**	**51 or more DET**
1 RET	LOW	LOW	AVERAGE
2-5 RET	LOW	AVERAGE	HIGH
6 or more RET	AVERAGE	HIGH	HIGH

Based on the complexity of each elements, we can calculate the total of Unadjusted Function Points (UFPs), which give an indication about the functional size of the system. The calculation of the UFP is based on the following table:



	EO	EQ	EI	ILF	ELF
LOW	4	3	3	7	5
AVARAGE	5	4	4	10	7
HIGH	6	5	6	15	10

The following tables contain the elements we need to compute the value of the Functional Point Count (FPC).



Type	Description	FTR	DET	Elements
EI	Customer Registration	1 (user)	12	10 in User 1 for error checking 1 for operation confirmation
EI	Driver Registration	3 (User, Driver, Vehicle)	20	10 in User 5 in Driver 3 in Vehicle 1 for error checking 1 for operation confirmation
EI	Taxi Request	1 (Ride_Request)	9	7 in Ride_Request 1 for error checking 1 for operation confirmation
EI	(Driver) Ride Request Acceptation	1 (Booked_Ride)	3	2 in Booked_Ride 1 for operation confirmation
EI	(Customer) Join a shared ride	1 (Passenger)	3	2 in Passenger 1 for operation confirmation
EI	(Driver) Status Management	1 (Driver)	2	1 in Driver 1 for operation confirmation
EI	Ride Delete	3 (Ride_Request, Booked_Ride, Passenger)	12	7 in Ride_Request 2 in Booked_Ride 2 in Passenger 1 for operation confirmation
EQ	Display Shared Ride	1 (Ride_Request)	7	7 in Ride_Request
EQ	Display User Profile	1 (User)	5	5 in User
EQ	Display Driver Profile	2 (User, Driver)	7	5 in User 2 in Driver
EO	Display a Shared Request	2 (Ride_Request, Passenger)	8	5 in Ride_Request 2 in Passenger 1 for calculation of the cost

USER:(FirstName, Surname, Address, Telephone, BornDate, Email, Username, Password, CreditCardIDNumber, Type)

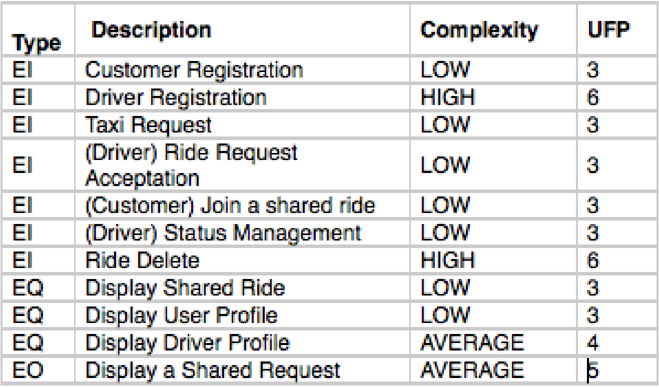
DRIVER: (Username, License number, Vehicle, IBAN, status)

VEHICLE: (Vehicle Registration Plate, Vehicle Model, MaximumNumOfPassengers)

RIDE_REQUEST(NumRide,StartingPosition, Destination, Date, Time, Customer, Sharing)

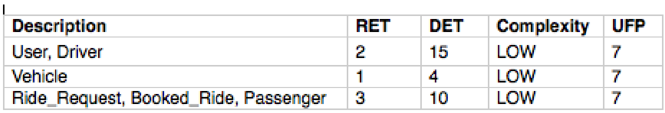
BOOKED_RIDE(Ride, Driver)

PASSENGER(Ride, UserUsername)



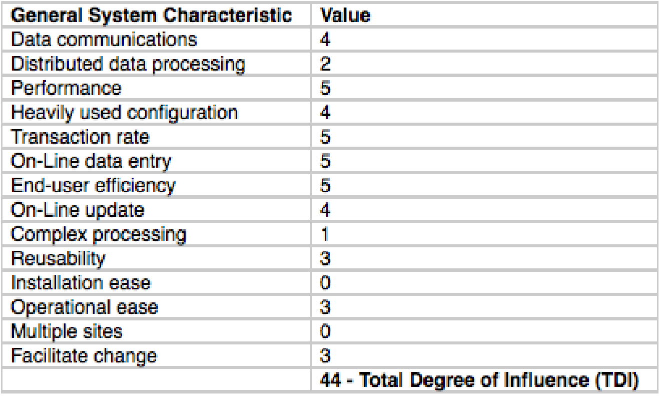
Type	Description	Complexity	UFP
EI	Customer Registration	LOW	3
EI	Driver Registration	HIGH	6
EI	Taxi Request	LOW	3
EI	(Driver) Ride Request Acceptation	LOW	3
EI	(Customer) Join a shared ride	LOW	3
EI	(Driver) Status Management	LOW	3
EI	Ride Delete	HIGH	6
EQ	Display Shared Ride	LOW	3
EQ	Display User Profile	LOW	3
EQ	Display Driver Profile	AVERAGE	4
EO	Display a Shared Request	AVERAGE	5

The following table contains RETs and DETs contained by the ILFs of the application



Description	RET	DET	Complexity	UFP
User, Driver	2	15	LOW	7
Vehicle	1	4	LOW	7
Ride_Request, Booked_Ride, Passenger	3	10	LOW	7

The last value we need to obtain the Final Function Point Count (FPC) is the the Value Adjustment Factor (VAF). Previously we have to assign a value to 14 general system characteristics based on whether is has no influence to strong influence on a scale of 0 to 5.

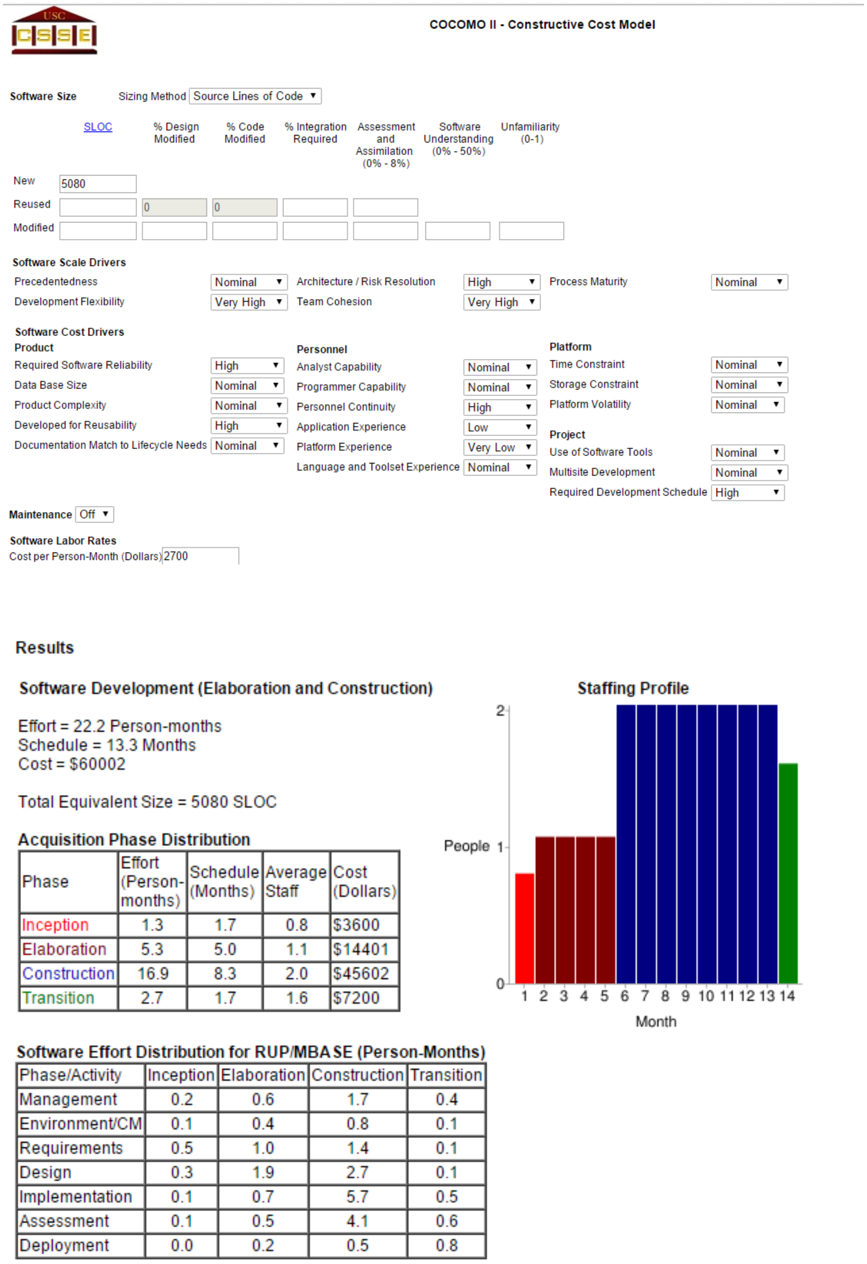


General System Characteristic	Value
Data communications	4
Distributed data processing	2
Performance	5
Heavily used configuration	4
Transaction rate	5
On-Line data entry	5
End-user efficiency	5
On-Line update	4
Complex processing	1
Reusability	3
Installation ease	0
Operational ease	3
Multiple sites	0
Facilitate change	3
	44 - Total Degree of Influence (TDI)

2.2 COCOMO

Here is reported the estimation done with the help of:

http://csse.usc.edu/tools/COCOMOII.php



COCOMO II - Constructive Cost Model

Software Size Sizing Method Source Lines of Code ▾

	SLOC	% Design Modified	% Code Modified	% Integration Required	Assessment and Assimilation (0% - 8%)	Software Understanding (0% - 50%)	Unfamiliarity (0-1)
New	5080						
Reused		0	0				
Modified							

Software Scale Drivers

Precedentedness	Nominal ▾	Architecture / Risk Resolution	High ▾	Process Maturity	Nominal ▾
Development Flexibility	Very High ▾	Team Cohesion	Very High ▾		

Software Cost Drivers

Product

Required Software Reliability	High ▾
Data Base Size	Nominal ▾
Product Complexity	Nominal ▾
Developed for Reusability	High ▾
Documentation Match to Lifecycle Needs	Nominal ▾

Personnel

Analyst Capability	Nominal ▾
Programmer Capability	Nominal ▾
Personnel Continuity	High ▾
Application Experience	Low ▾
Platform Experience	Very Low ▾
Language and Toolset Experience	Nominal ▾

Platform

Time Constraint	Nominal ▾
Storage Constraint	Nominal ▾
Platform Volatility	Nominal ▾

Project

Use of Software Tools	Nominal ▾
Multisite Development	Nominal ▾
Required Development Schedule	High ▾

Maintenance Off ▾

Software Labor Rates

Cost per Person-Month (Dollars) 2700

Results

Software Development (Elaboration and Construction)

Effort = 22.2 Person-months
Schedule = 13.3 Months
Cost = $60002

Total Equivalent Size = 5080 SLOC

Acquisition Phase Distribution

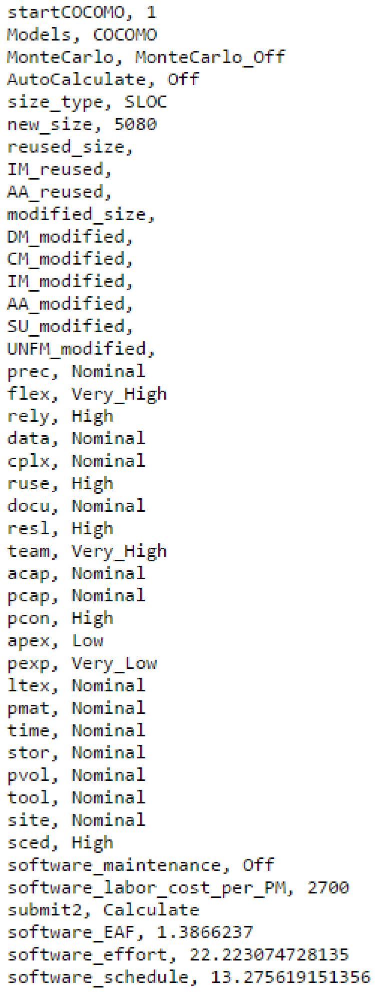
Phase	Effort (Person-months)	Schedule (Months)	Average Staff	Cost (Dollars)
Inception	1.3	1.7	0.8	$3600
Elaboration	5.3	5.0	1.1	$14401
Construction	16.9	8.3	2.0	$45602
Transition	2.7	1.7	1.6	$7200

Staffing Profile

Software Effort Distribution for RUP/MBASE (Person-Months)

Phase/Activity	Inception	Elaboration	Construction	Transition
Management	0.2	0.6	1.7	0.4
Environment/CM	0.1	0.4	0.8	0.1
Requirements	0.5	1.0	1.4	0.1
Design	0.3	1.9	2.7	0.1
Implementation	0.1	0.7	5.7	0.5
Assessment	0.1	0.5	4.1	0.6
Deployment	0.0	0.2	0.5	0.8

Here it is the output file generated by the application:



```
startCOCOMO, 1
Models, COCOMO
MonteCarlo, MonteCarlo_Off
AutoCalculate, Off
size_type, SLOC
new_size, 5080
reused_size,
IM_reused,
AA_reused,
modified_size,
DM_modified,
CM_modified,
IM_modified,
AA_modified,
SU_modified,
UNFM_modified,
prec, Nominal
flex, Very_High
rely, High
data, Nominal
cplx, Nominal
ruse, High
docu, Nominal
resl, High
team, Very_High
acap, Nominal
pcap, Nominal
pcon, High
apex, Low
pexp, Very_Low
ltex, Nominal
pmat, Nominal
time, Nominal
stor, Nominal
pvol, Nominal
tool, Nominal
site, Nominal
sced, High
software_maintenance, Off
software_labor_cost_per_PM, 2700
submit2, Calculate
software_EAF, 1.3866237
software_effort, 22.223074728135
software_schedule, 13.275619151356
```

Chapter 3

Task Identification

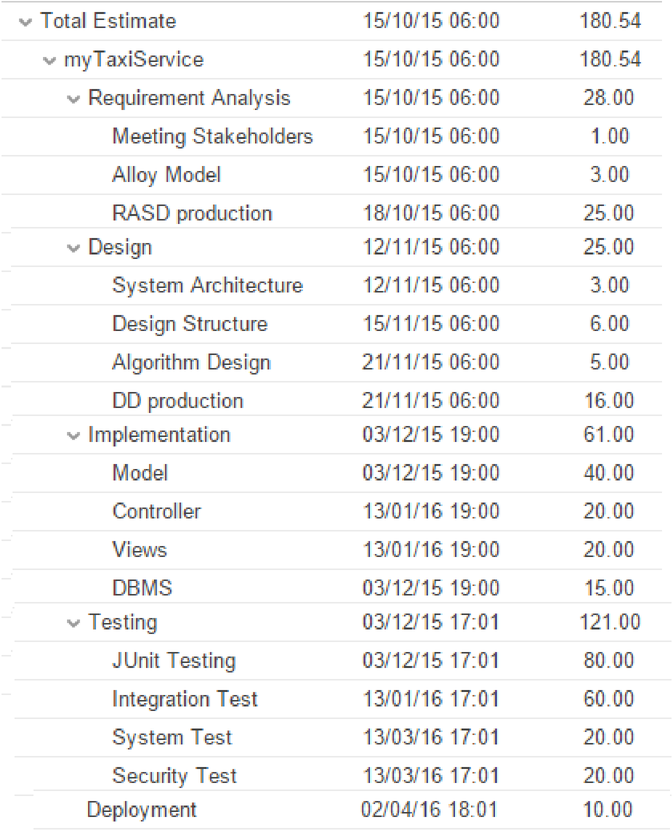
3.1 Project Tasks

Here are listed the main task of the project. For each tasks, it's possible to define a list of subtasks.

1. Requirement Analysis
   - Meeting stakeholders
   - Alloy Model
   - RASD production
2. Design
   - System architecture
   - Design structure
   - Algorithm design
   - Design Document production
3. Implementation
   - DBMS
   - Model
   - Controller
   - Views
4. Testing
   - JUnit testing
   - Integration Tests
   - System Tests
   - Security Test
5. Deployment

3.2 Task Schedule

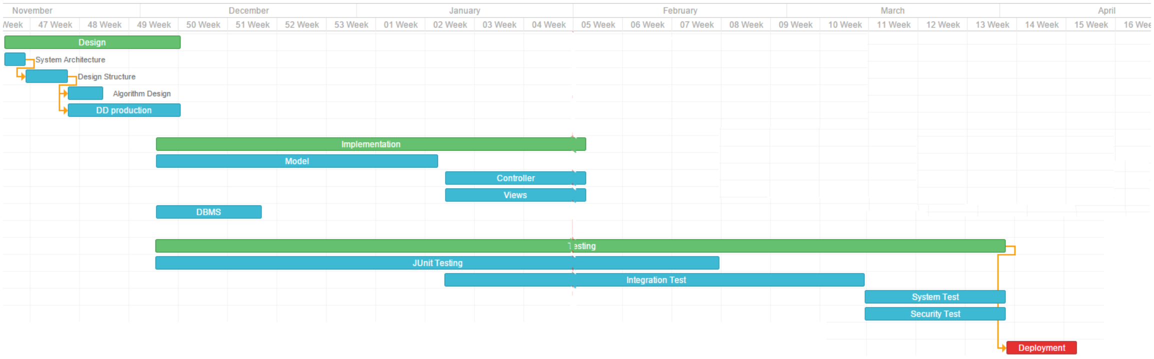
This image shows the schedule of the tasks previously described.



Total Estimate	15/10/15 06:00	180.54
myTaxiService	15/10/15 06:00	180.54
Requirement Analysis	15/10/15 06:00	28.00
Meeting Stakeholders	15/10/15 06:00	1.00
Alloy Model	15/10/15 06:00	3.00
RASD production	18/10/15 06:00	25.00
Design	12/11/15 06:00	25.00
System Architecture	12/11/15 06:00	3.00
Design Structure	15/11/15 06:00	6.00
Algorithm Design	21/11/15 06:00	5.00
DD production	21/11/15 06:00	16.00
Implementation	03/12/15 19:00	61.00
Model	03/12/15 19:00	40.00
Controller	13/01/16 19:00	20.00
Views	13/01/16 19:00	20.00
DBMS	03/12/15 19:00	15.00
Testing	03/12/15 17:01	121.00
JUnit Testing	03/12/15 17:01	80.00
Integration Test	13/01/16 17:01	60.00
System Test	13/03/16 17:01	20.00
Security Test	13/03/16 17:01	20.00
Deployment	02/04/16 18:01	10.00

3.2.1 Gantt

This Gantt chart is intended to give a better overview of the tasks and their related time.



Chapter 4

Resource Allocation

The team working on the MyTaxiService system is composed by three members:

- Matteo Farè
- Thierry Ebali Essomba
- Federico Feresini

Team members can be considered as resources of the project. In this chapter each task previously described is assigned to one or more members of the group.

4.1 Requirement Analysis

Task	**Resource**
Meeting Stakeholders	All team members
Alloy Model	Federico Feresini
RASD Production	All team members

4.2 Design

Task	**Resource**
System architecture	Thierry Ebali
Design structure	Matteo Farè
Algorithm Design	Federico Feresini
Design Document production	All team members

4.3 Implementation

Task	Resource
DBMS	Thierry Ebali
Model	Matteo Farè, Federico Feresini
Controller	Matteo Farè, Thierry Ebali
Views	Federico Feresini

4.4 Testing

Task	Resource
JUnit Testing	Matteo Farè
Integration Test	Thierry Ebali, Federico Feresini
System Tests	Federico Feresini
Security Tests	Thierry Ebali

4.5 Deployment

Task	Resource
Deployment	All team member

Chapter 5

Risk Evaluation

5.1 List of Risks

Risk	**Probability**	**Impact**
Inaccurate user estimation	Moderate	Marginal
Missing of functions	Low	Critical
Cost forecast are inaccurate	Moderate	Critical
Inaccurate change priorities	Low	Critical
Stakeholders conflicts	Low	Marginal
Architecture lacks feasibility	Low	Catastrophic
Architecture is not fit for purpose	Moderate	Critical
Design lacks feasibility	Low	Catastrophic
Failure to integrate components	High	Marginal
Requirements are low quality	Moderate	Critical
User interface is low quality	Moderate	Marginal

5.2 Recovery Actions

Here for each risk a possible recovery action to apply is provided.

- Inaccurate user estimation: try to make the system able to expand in case of an unexpected number of users.

- Missing of functions: immediately add these functions and integrate them to the rest of the system.

- Cost forecast are inaccurate: ask stakeholders for a more high budget, or save money on a part of the system in order to have more budget where it's needed.

- Inaccurate change priorities: reassign the jobs to each member of the group.

- Stakeholders conflicts: try to reach a common point between all different ideas.
- Architecture lacks feasibility: design a new proper architecture.
- Architecture is not fit for purpose: try to adapt the system for the chosen architecture. If this is impossible, change some part of the architecture.
- Design lacks feasibility: change some of the components of the project.
- Failure to integrate components: modify the way these components interacts.
- Requirements are low quality: meet stakeholders to have more information of what is needed.
- User interface is low quality: modify some features in order to have a more user-friendly interface.

Chapter 6

Appendix

6.1 Tools Used

- TeXstudio: to write and redact the whole document.
- Gantt Pro: to create the Gantt Chart.
- Gimp: to edit some images.

6.2 Hours of Work

In order to redact this document each member of the group has spent 7 hours.