POLITECNICO
MILANO 1863

PROJECT PLAN

V 0.0 - Software Engineering 2 - AA 2016/2017

Botta Daniel
806545

Bruschi Agnese
810762

Cano Adriana
812233

Summary

1 Introduction

The Project Plan aim is to create a rigorous plan for the progress of the process. It will consider the team composition, the various tasks each member will have to complete and the overall estimation of the time and resources needed to realize the project.

We reckoned that our estimation made is an estimation of an **early design**. As a matter of fact, at this point we imagine we are in the early stages of coding. Thus, we will first calculate the size of our project with Function Points, then we will use them to calculate effort and costs.

Lastly, we will present our project schedule, allocation of resources and risk management (which also is taken into account in the schedule planning).

1.1 Revision History

Version	Date	Authors	Summary
1.0	22/01/2017	Botta Daniel Enrico, Bruschi Agnese and Cano Adriana	Initial Release

1.2 Purpose and scope

This document represents the **Project Plan Document** for the **PowerEnJoy** system. The scope is to do a detailed analysis of the schedule of the work done so far, an estimation of the work needed for the project to be completed efficiently and effectively, allocation and maintenance of resources as well as to consider all the possible risks that the project might face, and how to proceed in case of that happening. Using different strategies, we calculate a numerical complexity of the work considering every component and their impact on it.

1.3 Definitions, Acronyms and Abbreviations

- **RASD:** Requirement Analysis and Specification Document
- **DD:** Design Document
- **ITPN:** Integration Test Plan Document
- **PM:** Project Management

- **PP:** Project Planning
- **FP:** Function Points
- **ILF:** Internal Logical File
- **EIF:** External Interface File
- **EI:** External Input
- **EO:** External Output
- **EQ:** External Inquiry
- **UFP:** Unadjusted Function Point

- **LOC:** Lines Of Code

- **FO:** Field Operator

1.4 Referenced Documents

Name	Version	Description
Assignments AA 2016 -2017.pdf	-	The headlines of the ITPD were descripted in this document.
RASD	2.0	Requirement Analysis and Specification Document
DD	1.2	Design Document
ITPN	1.1	Integration Test Plan Document
Function Point complexity evaluation tables	-	
COCOMO II Model definition manual	-	
QSM for AVC values	-	http://www.qsm.com/resources/function-point-languages-table

2 Project Size Estimation: Function Points

To estimate the size of our project, we will apply the **Function Points approach**. It assumes that the size of a project is dependent from the functionalities that the project itself has to offer.

A FP is a quantity which is calculated from a combination of **program characteristics** (data structures, inputs, outputs, inquiries and external interfaces); each one of them is associated with a **weight**, and the UFP is obtained by summing all the partial values, (number of elements of a type multiplied by the type weight).

The function types that allows us to estimate the size of our project, that form its external representation, are listed below.

To characterize the entities that will be considered (as simple, medium or complex) we will refer to the following weight table:

Function Types	Weights		
	Simple	Medium	Complex
n. inputs	3	4	6
n. outputs	4	5	7
n. inquiries	3	4	6
n. ILF	7	10	15
n. EIF	5	7	10

2.1 Internal Logic Files

ILFs are sets of data that are used and managed by the same application. PowerEnJoy needs to store information that is vital for the application to function.

The tables composing our system are named: **Cars**, **Users**, **Field Operators**, **Credentials** and **Reservations**.

All the information is stored in a string format, reducing thus the space needed to store items and it can be easily managed.

First of all, we can start with information about the **cars** offered to the clients. The system needs to know at any time things such as: car location, locking status of the car (locked, unlocked or temporarily locked), general state of the car (reserved, in use, on hold, under maintenance) and other information during the use of the car such as the number of people on it [see Management of Persistent Data, chapter 4.3.1.3 *Car*, DD].

For the **clients**, a profile is required to be created in order for them to be able to use the services of PowerEnJoy. In order to keep the integrity of the application, we need to store the information inserted by them upon registration: name, surname, driving license and payment method, eventual additional information such phone number and address. Also, information related to the client's use of the services is stored: the most recent searches made, concluded reservations. For each user we will also need to know in case of payment issues the profile is blocked/pending and therefor the services to be temporarily prohibited to such user [see Management of Persistent Data, chapter 4.3.1.1 *Customer*, DD].

Another actor in our system that needs to have data stored is the Field Operator. The FO is in possession of an account. The information stored comprehends: e-mail address, password, driving license and phone number.

Credentials: in this table, for each user and for each field operator registered in the system their e-mail address and password are stored, so that the system can have a quick look-up for the process of logging in or logging out.

Reservations are stored in a different table, where their status is updated during the ongoing process. Also, other information stored includes the user who is associated to the reservation (User_ID), the calculated fee at the end of the reservation (it is zero at starting time) and the payment procedure.

ILF	Complexity	FPs
Car	Complex	15
User	Medium	10
Field Operator	Medium	10
Credentials	Simple	7
Reservation	Complex	15

Users and Field Operators were considered Medium complexity fields due to their connection to the payment information, and Reservation because of its multiple links to other entities (user, car).

$$\mathbf{UFP} = \sum \mathbf{ILF} * \mathbf{FPs} = 57$$

2.2 External Logic Files

ELF are data sets used by the application, but created and stored by other systems.

Concerning PowerEnJoy, all the information that the application gets from the car software is generated outside our application. (The number of people is counted by sensors, as it is the battery charge of the car. The minutes the car has been used, which will be inserted in the equation to establish the charge, are calculated by the car system clock, and the position is calculated thanks to a GPS.)

To allow that, the car system is required to have a minimum amount of memory, where the data will be stored. (2 Gb).

– Number of people in the car

– Current battery charge of the car

– Usage time (to calculate the fee)

– Car position

– Car reservation code inserted on the keypad

– Doors locked/unlocked

– If the parked car is charging or not

ELF	Complexity	FPs
Nb of people in the car	Simple	5
Current battery charge	Simple	5
Usage time	Simple	5
Car position	Simple	5
Car reservation code	Simple	5
Doors locked	Simple	5

Moreover, the system communicates with Google Map. When the user inserts its position or an address and selects the range that the system should consider when calculating which cars to show him/her, the system itself will singlehandedly calculate through a simple algorithm which cars to show, and pass to the map their positions. In this way the system will present the user the map of the chosen range with the available cars. Thus we don't need to take any information from the outside concerning this particular feature.

UFP = ∑ (ELF*FPs) = 30

2.3 External Inputs

External Inputs elaborate data coming from different sources. We have divided ours in those coming from the user and those coming from the field operator.

The user:

- ✓ Registration: the guest fills the reservation form and then the data is elaborated, making sure that the right information is inserted in each field. For this reason this is an operation of medium complexity, hence 4 FPs.
- ✓ Login/logout: simple operations that involve only the Profile Manager: 3 FPs each.
- ✓ Information update: the user can change his/her profile data, and since the Profile Manager may check the correctness of the credit card date and check for database info, it's considered a medium complexity operation, hence 4FPs.
- ✓ Search for a car: this operation requires a thorough search through the database to find the appropriate cars, so we give it 4 FPs.

- ✓ **Reserve a car**: this operation is quite complex, seeing that it involves Reservation Manager, Car Manager and Payment Manager, hence 6 FPs.
- ✓ **Cancel a reservation**: this operation involves only the Reservation Manager and the cancelation of an entity stored in the database, so it has been considered simple, and has been given 3 FPs.

Registration, Information update, search for a car are operations that require a search in the database, to save or search through a relevant amount of data. Thus, this is why they were considered to be of Medium complexity.

USER	Complexity	FPs
Registration:	Medium	4
Login/logout:	Simple	2x3=6
Information update:	Medium	4
Search for a car:	Medium	4
Reserve a car:	Complex	6
Cancel a reservation:	Simple	3

Field operator:

- ✓ **Registration**: this is a fairly simple operation that involves only the Operator Manager, but it does some checks for the integrity of the fields completed, hence it is a medium complexity operation that gets 4 FPs.
- ✓ **Login/logout**: simple operations, completely identical to the user login and logout.
- ✓ **Update car information**: this a fairly complex operation that involves Operator Manager and Car Manager, which updates the status of the car throughout the entire system, making said car available to offer services once again, hence 6 FPs.

FO	Complexity	FPs
Registration	Medium	4
Login/Logout	Simple	3
Update car info	Complex	6

The overall count of FPs for EI:

UFP = ∑ (ELF*FPs) = 40

2.4 External Inquiries

External Inquiries are all the actions in which input and output are considered, but not modified. In our case we have two sources from which the inquiries can come from: the users and the FOs. All the inquiries are seen as fairly simple operations of information retrieval.

User:

- ✓ View of information: this is a simple operation, in which the client can see all the data stored in the profile.
- ✓ View of reservations: the user can see his/her past reservations, and the current one (if it exists).
- ✓ View of recent Searches: the user is shown the recent searches as hint for his/her new operations.

They are all simple operations since they involve only the Profile Manager, so their FP value is 3.

USER	Complexity	FPs
View of reservations	Simple	3
View of recent Searches:	Simple	3
View of information:	Simple	3

Field Operator:

- ✓ View of cars with problems: the field operator can access to the list of cars that require his/her intervention.

FO	Complexity	FPs
View cars	Simple	3

The overall complexity is:

UFP = ∑ (ELF*FPs) = 12

2.5 External Outputs

- ✓ Result of a search
- ✓ Notify a user of a successful reservation
- ✓ Notify a user of successful payment of a reservation
- ✓ Notify a user of the commitment of the changes requested

- ✓ Notify the user of the successful registration
- ✓ Notify the user that his/her reservation has been cancelled
- ✓ Notify a FO that the car status was updated

EO	Complexity	FPs
Successful registration notification (Mail with password)	Simple	3
Successful reservation notification (Mail with reservation code)	Simple	3
Notify a user of successful payment of a reservation (Mail with a receipt of the payment)	Medium	4
Profile Blocked Notification (Mail)	Medium	4
Profile update notification	Simple	3
Successful reservation cancellation notification	Simple	3

The overall complexity is:

UFP = ∑ (ELF*FPs) = 20

2.6 Overall estimation

The overall estimation of the size of the PowerEnJoy project takes into account all the UFPs considered until now.

FPtot = ∑ UFP = 57 + 30 + 40 + 12 + 20 **= 159**

This estimation will be used to calculate the cost and effort estimation. To do that, the total number of FPs is used to calculate the number of lines that should be necessary to write the program defined by the FPs.

LOC = AVC * FPtot

AVC is a language dependent factor, and for JavaScript the average AVC is 47.

We use JavaScript AVC in the LOC esteem:

LOC = ∑ (AVC * FPtot) = 47 * 159 **= 7 473 LOC**

3 Cost and effort estimation: COCOMO II

To estimate the cost and the effort needed for the development of the project we opted for the **Algorithmic cost modelling.**

Effort = A x SizeB x M

- **A** is a constant which depends on the organization
- **B** is the size,
- **M** is a multiplier which represents product, process and people attributes. The most commonly used product attribute for cost estimation is **code size**.

3.1 Scale Drivers

To estimate the costs, the first step is to define some key values, which we present here:

- **Precedentedness (PREC):** presents how much a product is similar to previous ones.
- **Flexibility(FLEX):** defines how much the project is free from any kind of specific constraints to conform with, i.e. pre-established requirements and external interface specs.
- **Architecture / Risk Resolution(RESL):** defines if the risk management plan is complete and accurate, and there are clear definitions of budget and schedule and architecture.
- **Team Cohesion(TEAM):** defines how well the stakeholders and member of the team are able to work together and agree on the views and decisions.
- **Process Maturity(PMAT):** refers to a well-known method for assessing the maturity of a software organization, CMM, now evolved into CMMI.

Scale factors can be *very low*, *low*, *nominal*, *high*, *very high*, *extra high*. Based on their given attribute, a weight is assigned to them, in order to be in the equation.

This is the table of attributes related to the PowerEnJoy system that our documents propose:

PREC	High	2.48
FLEX	Nominal	3.72
RESL	High	2.83
TEAM	Very High	1.10
PMAT	High (SW - CMM Level 3)	3.12

- **PREC** is **high**: precedentedness is high (high = generally familiar). This is because a similar software (EnJoy), which our system enhances and emulates, already existed.
- **FLEX** is **Nominal**: (nominal = some relaxations). Our system has to be integrated with preexistent systems, but overall there is freedom of choice in the architecture structure design.
- **RESL** is **high**: we reckon that our risk management plan covers most of the problems that may occur.
- **TEAM** is **very high**, there is a high level of cohesion within the group, and every decision has been taken by mutual agreement.

- **PMAT** is high (SW-CMM Level 3): We as a group have level have developed its own standard software process through greater attention to documentation, standardization, and integration.

Therefore, the total sum is:

$$\textbf{UFP} = \sum_{j=1}^{5} \mathrm{SFj} = \textbf{13,25}$$

From the values we can derive the parameter E:

$$\textbf{E} = \mathrm{B}+ 0.01 * \sum_{j=1}^{5} SFj = 0.91 + 0.01*13{,}25 = \textbf{1.05}$$

3.2 Cost Drivers

Product factors

- Required Software Reliability (RELY): This is the measure of the extent to which the software must perform its intended function over a period of time. Seeing as it's a web based app the risk is low.
- Data Base Size (DATA): This measure attempts to capture the effect that large data requirements have on product development. The rating is determined by calculating (Database Size)/LOC. Seeing as we do not have a physical database size we are going to put an average value (nominal).
- Product Complexity (CPLX): We have selected high, according to the COCOMO II rating scale.
- Developed for Reusability (RUSE): This cost driver accounts for the additional effort needed to construct components intended for reuse on the current or future projects. Nominal is adequate since we want generic design of software, but only to a certain extent
- Documentation Match to Life-Cycle Needs (DOCU): In COCOMO II, the rating scale for the DOCU cost driver is evaluated in terms of the suitability of the project's documentation to its life-cycle needs. Low is the most appropriate value since every case of the product life-cycle is already foreseen in the documentation.

RELY Descriptors:	slight inconven-ience	low, easily recoverable losses	moderate, easily recoverable losses	high financial loss	risk to human life	
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	0.82	0.92	1.00	1.10	1.26	n/a

RELY	Low	0,92
DATA	Nominal	1
CPLX	Low	0.92
RUSE	Nominal	1
DOCU	Very Low	0.82

Platform factors

- Execution Time Constraint (TIME) This is a measure of the execution time constraint imposed upon a software system. Since the complexity is high, it is appropriate to set the time constraint to high, too.
- Main Storage Constraint (STOR): This rating represents the degree of main storage constraint imposed on a software system or subsystem. Storage is not a problem for PowerEnJoy, since most devices nowadays have terabytes of memory available, and it is an app designed not to occupy too much memory on the customer's phone.
- Platform Volatility (PVOL): "Platform" is used here to mean the complex of hardware and software (OS, DBMS, etc.) the software product calls on to perform its tasks. This rating ranges from low, where there is a major change every 12 months, to very high, where there is a major change every two weeks. Major changes aren't expected.

TIME	High	1.10
STOR	Low	0.92
PVOL	Very Low	0.82

Personnel Factors

- Analyst Capability (ACAP): Analysts are personnel that work on requirements, high level design and detailed design. Our requirement analysis and design of the system has been conducted thoroughly. For this reason, it's set to nominal.
- Programmer Capability (PCAP): Evaluation should be based on the capability of the programmers as a team rather than as individuals. Team synergy exists, therefore value is set to nominal.
- Personnel Continuity (PCON): The rating scale for PCON is in terms of the project's annual personnel turnover. Considering no members of the group has the intention of leaving, this parameter is very low.
- Applications Experience (APEX): This rating is dependent on the level of applications experience of the project team developing the software system or subsystem. The ratings are defined in terms of the project team's equivalent level of experience with this type of application. We are competent in what we do, so nominal is chosen.
- Platform Experience (PLEX): Considering we have never faced a system similar to this kind the value is set to low.
- Language and Tool Experience (LTEX): This is a measure of the level of programming language and software tool experience of the project team developing the software system or subsystem. The tools that we should utilize are quite user-friendly. No complications are in sight therefore we have set the value to nominal.

ACAP	Nominal	1
PCAP	Nominal	1
PCON	Low	0.82
APEX	Nominal	1
PLEX	Low	0.92

LTEX	Nominal	1

Project Factors

- Use of Software Tools (TOOL): The tool rating ranges from simple edit and code (= very low), to integrated lifecycle management tools (= very high). The tools that we should utilize are quite user-friendly and are in fact simple to edit and code.
- Multisite Development (SITE): Determining cost drivers rating involves the assessment and averaging of two factors: *site collocation* (from fully collocated to international distribution) and *communication support* (from surface mail and some phone access to full interactive multimedia). The parameter is set to nominal.

TOOL	Nominal	1
SITE	Nominal	1

General Factor

- **Required Development Schedule (SCED):** This rating measures the schedule constraint imposed on the project team developing the software.

SCED	Nominal	1

Finally, with these gathered value we can find the effort multiplier:

EAF = $\sum_{j=1}^{n} \mathrm{EMj}$ **= 0.467**

3.3 Effort equation

Now we can evaluate the effort needed, and thus, the estimated time.

For the SIZE, as we already stated, we have used the QSM table to get an estimate FP for JavaScript of 47.

The equation to estimate effort is the following:

$$PM = A \times SIZE\ E \times EAF$$

As a brief recap, it is already stated in the document that:

- A = 2.94,
- E = 1.05,
- SIZE = 47,
- EAF = 0.467

$$PM = 2.94 * (7\ 473 * 1{,}05) * 0.467 = 11{,}34\ PM$$

Through the found effort value, it's possible to calculate the value of the **duration (in months) of the project**, using an exponent:

$$F = 0.28 + 0.2 * (E - B) = 0.28 + 0.2 * (1.05 - 0.91) = 0.308$$

$$\text{Duration} = 3{,}67 * PMF = 3{,}67 * (11{,}34 * 0{,}308) = 7{,}75 \text{ months}$$

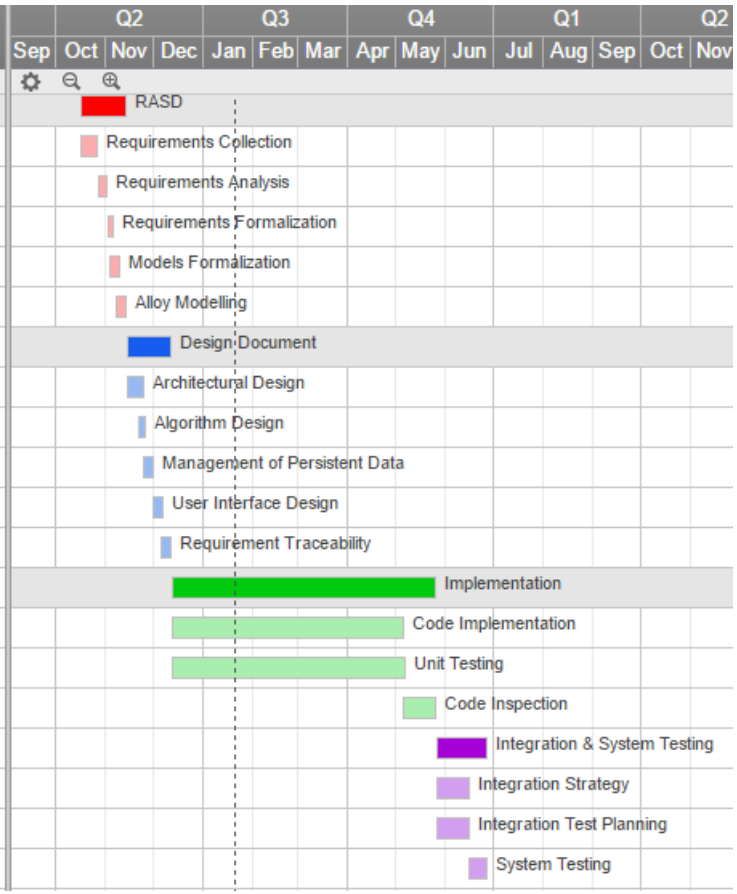
The approximate duration of the project is 8 months, with 3 people working on it.

4 Schedule

In this chapter the tasks and the schedule for completing the project are identified. As it happened, the project started on October 16, 2016 with the introduction of what it consists of and what was required to fulfil. The project is divided in five assignments, each one of them with a scope, specific fields included and a fixed deadline.

– The first assignment was RASD (Requirement Analysis and Specification Document), submission deadline: 15/01/2016.
– The second assignment was DD (Design Document), submission deadline: 11/12/2016.
– The third assignment was ITPD (Integration Test Plan Document), submission deadline: 22/01/2017.
– The fourth assignment was PM (Project Management), submission deadline: 05/01/2017.

	Task	Start Date	End Date	Duration(Days)	Assigned To
1	**RASD**	16/10/16	13/11/16	28	
2	Requirements Collection	16/10/16	26/10/16	10	Botta, Bruschi, Cano
3	Requirements Analysis	27/10/16	01/11/16	5	Botta, Bruschi, Cano
4	Requirements Formalization	02/11/16	05/11/16	3	Botta, Bruschi, Cano
5	Models Formalization	03/11/16	09/11/16	6	Botta, Bruschi, Cano
6	Alloy Modelling	07/11/16	13/11/16	6	Botta, Bruschi, Cano
7	**Design Document**	14/11/16	11/12/16	27	
8	Architectural Design	14/11/16	24/11/16	10	Botta, Bruschi, Cano
9	Algorithm Design	21/11/16	25/11/16	4	Botta, Bruschi, Cano
10	Management of Persistent Data	24/11/16	30/11/16	6	Botta, Bruschi, Cano
11	User Interface Design	30/11/16	06/12/16	6	Botta, Bruschi, Cano
12	Requirement Traceability	05/12/16	11/12/16	6	Botta, Bruschi, Cano
13	**Implementation**	12/12/16	25/05/17	164	
14	Code Implementation	12/12/16	05/05/17	144	Botta, Bruschi, Cano
15	Unit Testing	12/12/16	06/05/17	145	Botta, Bruschi, Cano
16	Code Inspection	05/05/17	25/05/17	20	Botta, Bruschi, Cano
17	**Integration & System Testing**	26/05/17	26/06/17	31	
18	Integration Strategy	26/05/17	15/06/17	20	Botta, Bruschi, Cano
19	Integration Test Planning	26/05/17	15/06/17	20	Botta, Bruschi, Cano
20	System Testing	15/06/17	26/06/17	11	Botta, Bruschi, Cano



5 Resource allocation

In this chapter we are going to provide a general overview of how all the tasks introduced in the schedule were divided to three team members working on this project.

We will start with a disclaimer: as you can see in the schedule code implementation is included in the schedule even though no implementation took place (for the reason that it was not required). The reason this was added was to keep the work as realistic as possible hence an estimated time is included.

The work division was done in full agreement and understanding among the team members. In the following, we will show the work done by each member for each document presented and the time required to do that.

RASD

For this assignment the work starting by doing three team meetings to decide on the strategies that would later be implemented and the decision regarding the work division were taken. Topics that were discussed in team with the respective hours that it took to finalize:

- Project analysis and discussion 5h
- Domain Properties 2h
- Assumptions 5h
- Requirements 5h

The individual work involved was done as follows:

- Adriana Cano:
  - Alloy 30h
  - UML 4h
- Agnese Bruschi:
  - Layout 8h
  - UML 4h
  - User Interface 3h
  - Sequence diagrams 15h
- Daniel Enrico Botta:
  - Scenario 5h
  - Use Cases and description 15h

DD

For this assignment we followed the same strategy as for RASD, starting by some team meetings deciding on the general lines for how to proceed with the fulfilment of the requirements asked for the assignment. The things discussed during the team meetings are:

- Architecture Design: Components 3h
- Database representation: 1h 30m

- Adriana Cano:
  - Architectural Design: general description, Components View, Components Interface 15h
  - Requirement Traceability: 1h
  - BCE: Search-Reserve diagram 1h
  - SD: Search 1h
  - Layout of the document 7h

- Agnese Bruschi:
  - Database representation 12h
  - Mockups 5h
  - UX diagrams 3h
  - BCE: Log-in diagram 2h
  - SD: Log-in 1h
  - Layout of the document 2h

- Daniel Botta:
  - Algorithms
  - Runtime View diagram
  - Deployment View diagram
  - Partook in the decisions regarding the architecture of the system

ITPN

Once again the same strategy was followed to fulfil the requests for the assignment. In the team meetings, the things that were discussed:

- Integration Testing strategy, Entry criteria, Elements to be tested 3h
- Work division 30 m

For each member the work load is:

- Adriana Cano:
  - Individual steps and test description 12 h

- Agnese Bruschi:
  - Layout of the document 1h
  - Integration Strategy 10h
  - Low level Integration Test 3h

- Daniel Botta:
  - Tool and Equipment Required 3h
  - Program Stubs and Test Data Required 3h

PP

The Project Plan uses information and analysis that the team already discussed and planned thoroughly. This is the division of work that has resulted from meetings:

- Adriana Cano:
  - Resource Allocation 1h
  - Project Size Estimation 3h
  - Risk Management 2h
  - Layout 1h

- Agnese Bruschi:
  - Introduction 1h
  - Project Size Allocation 6h
  - Cost and Effort Estimation 1 h
  - Layout and overall analysis 3h

- Daniel Botta:
  - Schedule 3h
  - Cost and effort estimation 7h
  - Resource Allocation 1h
  - Introduction 30 min

In the diagram presented in the following page a representation of the schedule can be seen, followed by the team members that completed the assignments. For each assignment it shows the sections covered, who did the work and the time it took for each of them to be done.
Moreover, the team reckoned that during the progress of the project, some decisions that were taken in the earlier stages had to be revised and changed, or evolved. Thus it was necessary to modify partially the previous documents, which were updated, and the new versions were uploaded.
The team estimated an average of 4 hours for the review.

	Task Name	Duration(Hours)	Assigned To
1	**RASD**		
2	Project Analysis and Discussion	5h	Botta, Bruschi, Cano
3	Domain Properties	2h	Botta, Bruschi, Cano
4	Assumptions	5h	Botta, Bruschi, Cano
5	Requirements	5h	Botta, Bruschi, Cano
6	Alloy Modelling	30h	Cano
7	Uml	4h	Cano
8	Layout	8h	Bruschi
9	Uml	4h	Bruschi
10	User Interface	3h	Bruschi
11	Sequence Diagrams	15h	Bruschi
12	Scenarios	5h	Botta
13	Use Cases and description	15h	Botta
14	**Design Document**		
15	Architecture Design : Components	3h	Botta, Bruschi, Cano
16	Database representation	1h 30m	Botta, Bruschi, Cano
17	Architectural Design	15h	Cano
18	Requirement Traceability	1h	Cano
19	BCE: Search-Reserve diagram	1h	Cano
20	SD: Search	1h	Cano
21	Document Layout	7h	Cano
22	Database Representation	12h	Bruschi
23	Mockups	5h	Bruschi
24	UX Diagrams	3h	Bruschi
25	BCE: Log-in diagram	2h	Bruschi
26	SD: Log-in	1h	Bruschi
27	Document Layout	2h	Bruschi
28	Algorithms	6h	Botta
29	Architecture of the System	5h	Botta
30	Runtime View diagram	2h	Botta
31	Deployment View diagram	2h	Botta
32	**ITPN**		
33	Integ. Testing Discussion	1h	Botta, Bruschi, Cano
34	Entry Criteria Discussion	1h	Botta, Bruschi, Cano
35	Elements to be Tested Discussion	1h	Botta, Bruschi, Cano
36	Individual Steps & Test Desctiption	12h	Cano
37	Document Layout	1h	Bruschi
38	Integration Strategy	10h	Bruschi
39	Low Level Integration Test	3h	Bruschi
40	Tool and Equipment Required	6h	Botta
41	Program Stubs & Test Data Required	6h	Botta
42	**Project Planning**		
43	Introduction	1h	Botta, Bruschi
44	Project Size Estimation (FP) :	8h	Bruschi, Cano
45	COCOMO II : Cost & Effort Estimation	7h	Botta
46	Schedule	3h	Botta
47	Resource Allocation	2h	Botta, Cano
48	Risk Management	2h	Cano
49	Document Layout	3h	Bruschi

6 Risk management

In this section we will introduce all the risks that the development of the project might have to face at some point. To make it more clear, we have individuated TOT types of risks for the project. For each one of them we will go into detail describing the possible things happening and a way to solve the problem in case it happens.

Business Risks:

In this category we consider the risks related to the *business logic* behind that project. Such risks come from the outside, such as the Government. PowerEnJoy is a service that will require the use of a sum of parking spaces across the city, permission is required from the Government to do such things. At some point we could face the decision of the Government on revoking such right. A solution to such thing would be a well-defined contract at starting time, assuring the right for the use of the parking spaces.

Project Risks:

This category of risks is related to the internal construction and management of the project. Such risks can be: organization of the work that needs to be done, unexpected complications such as illness, unavailability of the team members or withdrawal of one or more team members, changes made to the requirements by stakeholders.

Organization of the work: a solution to this problem would be a detailed schedule produced from the moment all the work that needs to be done is defined. Each member is assigned work that fits him/her best and the workload is known from the start.

Unexpected situations: in case of illnesses or unavailability of the team members, the work is temporarily divided among the available members that are most informed of the work that is not being done. In the other case, where a team member or more withdraw from the project, it is a peculiar issue that every project has to deal with. A possible solution would a well-defined contract that prohibits the members to leave the project without a leave notice. In case a leave notice is given, there is enough time to add a new member to the team and train him/her during the period of the notice so that he/she can be prepared when starting to work on the project. Such thing would lead to an increase on the total budget of the problem, for the training period, but it would assure the completeness of the project on time.

Changes to the requirements: this is an issue that might rise if things are not discussed in detail on the first phases of the project. The first step to avoid this would be to exhaust all the possible scenarios before going on with anything on the project, leaving little space to future changes. Another thing that could be done for this risk is in the development of the project: it should be done in such way that modifications are easy to be made to it.

Technical risks:

In this category we include the risks that are of the technical nature. The main issue would be *data loss*. To avoid that, daily backup of the data would be fairly sufficient. Another technical risk in our case would be *loss of communication with the car*. To avoid such risk, seeing that the cars and the system installed in them is offered by a third party, we would need to make sure at choosing time of such third party that their system is reliable.