

**Software Engineering 2: “PowerEnJoy”**

**Project Plan (V. 1.0)**

Authors:

Sergio CAPRARA

Soheil GHANBARI

Erica TINTI

Milan, Italy

15/01/2017

**Table of contents**

[1. Introduction 3](#_Toc472332391)

[1.1 Revision History 3](#_Toc472332392)

[1.2 Purpose and Scope 3](#_Toc472332393)

[1.3 Definitions and Abbreviations 3](#_Toc472332394)

[1.4 Reference Documents 4](#_Toc472332395)

[2. Project size, cost and effort estimation 5](#_Toc472332396)

[2.1 Size estimation: function points 5](#_Toc472332397)

[2.2 Cost and effort estimation: COCOMO II 6](#_Toc472332398)

[3. Schedule 8](#_Toc472332399)

[4. Resource allocation 9](#_Toc472332400)

[5. Risk management 10](#_Toc472332401)

[6. Hours of work 11](#_Toc472332402)

# Introduction

## Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Authors** | **Description** |
| 1.0 | 22/01/2017 | S. Caprara, S. Ghanbari, E. Tinti | First release |

## Purpose and Scope

In this document, we are providing details on how the components described in the Design Document will be integrated. To ensure that the interaction between them will give the expected results, we are choosing the method to follow and we are keeping in mind that the Integration Test of a component will be done after having Unit Tested it.

In the following chapters, you will find detailed descriptions of the tests and the name of the tools to be used.

## Definitions and Abbreviations

* **User:** the person registered to the system and allowed to access to its functions.
* **Operator:** a person with technical skills, that fixes car issues.
* **App:** short term used to define a mobile application.
* **Power Plug:** a column with one or more electricity socket where it is possible to charge the car.
* **Safe Area** (or Parking Area): a parking area with parking shared with all the other divers and not especially reserved to PowerEnjoy.
* **Special Parking Area** (or Power Station): a parking area reserved exclusively to PowerEnjoy cars where, for each parking space there is a Power Plug where it is possible to charge a car.
* **Car:** PowerEnjoy car.
* **Reservation:** the relation between a user and a car, that allows the user to start using the car. The reservation guarantees that no one else can reserve and use the reserved car till the end of the rental.
* **DB:** database, the collection of system data.
* **DAO:** Data Access Object.
* **Pojo:** Plain Old Java Object. Object having only getter and setter methods.

## Reference Documents

The documents used as a reference to provide the design document are:

* Assignments AA 2016-2017.pdf
* Project planning example document.pdf
* RASD\_PowerEnjoy\_Caprara\_Ghanbari\_Tinti
* DesignDocument\_PowerEnjoy\_Caprara\_Ghanbari\_Tinti
* TestPlan\_PowerEnjoy\_Caprara\_Ghanbari\_Tinti\_v1.0.pdf

# Project size, cost, and effort estimation

This section of the document provides

## Size estimation: function points

The size of the project we are working on, will be estimated using the Function Point Analysis approach. This technique is based on five major components:

* External Input, all operation that takes data in from the external environment
* External Output, all operation that sends data out
* External Inquiry, all operation involving both input and output
* Internal Logic Files, data used and managed by our application
* External Interface Files, data used by our system but generated by other applications

The tables used for the estimation are provided here.

|  |  |  |  |
| --- | --- | --- | --- |
| **File Type Referenced** | **Data elements** | | |
| **1-4** | **5-15** | **> 15** |
| **0-1** | Low | Low | Avg |
| **2** | Low | Avg | High |
| **3 or more** | Avg | High | High |

**Table 1: External Input**

|  |  |  |  |
| --- | --- | --- | --- |
| **File Type Referenced** | **Data elements** | | |
| **1-5** | **6-19** | **> 19** |
| **0-1** | Low | Low | Avg |
| **2-3** | Low | Avg | High |
| **4 or more** | Avg | High | High |

**Table 2: External Output and External Inquiries**

|  |  |  |  |
| --- | --- | --- | --- |
| **Record Element Type** | **Data Elements** | | |
| **1-19** | **20-50** | **> 50** |
| **1** | Low | Low | Avg |
| **2-5** | Low | Avg | High |
| **6 or more** | Avg | High | High |

**Table 3: Internal Logic Files and External Interface Files**

|  |  |  |  |
| --- | --- | --- | --- |
| **Type of Component** | **Complexity of Components** | | |
| **Low** | **Average** | **High** |
| **External Inputs** | 3 | 4 | 6 |
| **External Outputs** | 4 | 5 | 7 |
| **External Inquiries** | 3 | 4 | 6 |
| **Internal Logic Files** | 7 | 10 | 15 |
| **External Logic Files** | 5 | 7 | 10 |

**Table 4: Unadjusted Function Points**

## External Inputs

**Common functions**

The Login and Logout (for both users and operators) are Low complexity functions, because they involve just one file containing the information, so they get 3 FPs each.

**User functions**

User Registration is a function that can be classified as having an average complexity, because it deals with different data and because more than one file is referenced. This leads to 4 FPs.

The Profile Update function involves many fields that need to be updated on the DB. Moreover, if the payment information changes, the system has to check if the new credit card is valid. For these reasons the operation has a medium complexity, corresponding to 4 FPs. (on the Excel 2 inputs taking to 8 FPs)

The Cancel Reservation operation has a low complexity because it references two type of files but has few data. So, it has 3FPs.

**Operator functions**

The Maintenance Request is a simple operation, involving only one component. For this reason, it has low complexity and gets 3 FPs.

The End of Maintenance operation involves few components and has a low complexity, leading to 3 FPs.

## External Outputs

Registration Confirmation

Reservation Expiration

The Money Charge Information is an operation sent by the system to the user and requires the interaction with many different components and has many interactions. For this reason, the complexity is high and the assigned FPs are 7.

## External Inquiries

**User functions**

Car Lookup

Car Information

Car Reservation

Car Unlock

Parking Areas Request low 3 FPs

Special Parking Areas Request low 3 FPs

**Operator functions**

Car List Retrieve low 3 FPs

Car Details medium 4FPs

## Internal Logic Files

Car Info medium 10 FPs

User medium 10 FPs

Parking Area low 7 FPs

Special Parking Area low 7 FPs

Operator low 7 FPs

## External Interface Files

Licence Validation

GPS Access

## Cost and effort estimation: COCOMO II

The cost and effort estimation is made using the COCOMO II method. For this one we provide the table containing figures used in the evaluation. It is based on the following scale drivers:

* Precedentedness (PREC): the value depends on the experience the team has on projects similar to the current one.
* Development flexibility (FLEX): represents the possibility to make changes to the project based on the strictness of the external requirements.
* Risk resolution (RESL): consists in the capacity to solve problems and is strictly related to the risk analysis provided in this document.
* Team cohesion (TEAM): its value represents the capacity of the team members to work together and cooperate
* Process maturity (PMAT): states the level of maturity reached in the process.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Scale Factors** | **Very Low** | **Low** | **Nominal** | **High** | **Very High** | **Extra High** |
| **PREC** | thoroughly  unprecedented | largely  unprecedented | somewhat  unprecedented | generally  familiar | largely  familiar | thoroughly familiar |
| **SFj** | 6.20 | 4.96 | 3.72 | 2.48 | 1.24 | 0.00 |
| **FLEX** | rigorous | occasional  relaxation | some  relaxation | general  conformity | some  conformity | general goals |
| **SFj** | 5.07 | 4.05 | 3.04 | 2.03 | 1.01 | 0.00 |
| **RESL** | little (20%) | some (40%) | often (60%) | generally (75%) | mostly (90%) | full (100%) |
| **SFj** | 7.07 | 5.65 | 4.24 | 2.83 | 1.41 | 0.00 |
| **TEAM** | very difficult  interactions | some  difficult  interactions | basically  cooperative  interactions | largely  cooperative | highly  cooperative | seamless  interactions |
| **SFj** | 5.48 | 4.38 | 3.29 | 2.19 | 1.10 | 0.00 |
| **PMAT** | Level 1  Lower | Level 1  Upper | Level 2 | Level 3 | Level 4 | Level 5 |
| **SFj** | 7.80 | 6.24 | 4.68 | 3.12 | 1.56 | 0.00 |

The higher level of components of our system that need to be integrated are:

* User Application
* Operator Application
* Controller
* Model
* Database

The listed components are divided into lower level components, that will need integration testing.

# Schedule

The following paragraphs contain the detail of the Test Cases defined in the previous chapter.

# Resource allocation

For supporting and automating Integration Tests we will use two testing tools: JUnit and Arquillian.

# Risk management

Integration tests should also verify the responses of the system in specific cases, such as

# Hours of work

To make this document we have spent:

* Sergio Caprara, 14 hours
* Soheil Ghanbari, 8 hours
* Erica Tinti, 14 hours