Project Plan Document - v1.0

Gianpaolo Branca Luca Butera Andrea Cini

POLITECNICO
MILANO 1863

Contents

1 Introduction

1.1 Purpose and Scope

1.2 Definitions, Acronyms, Abbreviations

- Data element type (DET): a unique user recognizable, non recursive, field.
- Record element type (RET): Record elements type, a user recognizable subgroup of data elements.
- File types referenced (FRT): files updated or referenced in a transaction.
- MSO: Money saving option.
- SA: Safe Areas.

1.3 Reference documents

2 Project size, cost and effort estimation

In this section we will use well known approaches to project planning to estimate di dimension and the cost in time and money that our project will have. We will use the Function Points approach for the size estimation and than COCOMO for the cost and effort estimation and we will not consider, for the most part, the presentation layer.

2.1 Size estimation: function points

In the following tables the reference tables we are going to use for the size estimation, they classify the complexity of each element counting the numbers of file types referenced, data element types and record element types. Obviously we are going to consider these reference values, but we are also going to estimate the complexity of the components based on the specific knowledge that we have acquired about the domain of our system.

Table 1: Logic Files

	Data Elements		
Record Elements	1-19	20-50	51+
1	Low	Low	Avg
2-5	Low	Avg	High
6+	Avg	High	High

Table 2: My caption

	Data Elements		
File Types	1-5	6-19	20+
0-1	Low	Low	Avg
2-3	Low	Avg	High
4+	Avg	High	High

Table 3: External Input

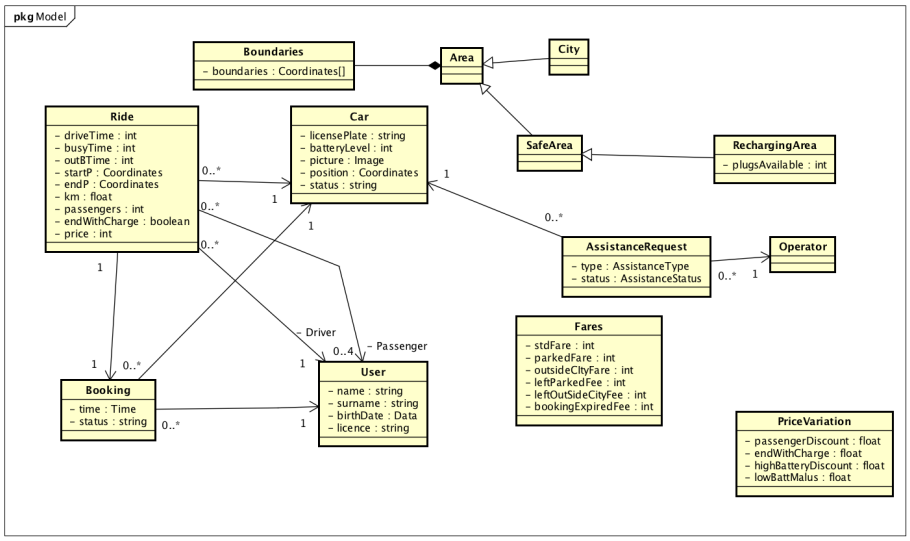
	Data Elements		
File Types	1-4	5-15	16+
0-1	Low	Low	Avg
2-3	Low	Avg	High
4+	Avg	High	High

Table 4: Function Points

	Complexity Weight		
Function Type	Low	Average	High
Internal Logic Files	7	10	15
External Logic Files	5	7	10
External Inputs	3	4	6
External Outputs	4	5	7
External Inquiries	3	4	6

2.1.1 Internal Logic Files (ILFs)

In this section we are going to analyze the complexity of our ILFs, we are going to refer to this simplified version of the model of the internal representation of our data (some attributes of the classes in the pictures are obviously not persistent but representative of the complexity of the informations related with the single entity):



- Users:
  - **Estimated complexity**: Average
    The first type of internal data our system will have to deal with are the user related ones. The handling of the user data will include not so trivial operations: password retrieval, driving license and PayPal account checks and profile personalization functionalities (profile image).
- Rides:
  - **Estimated complexity**: Low
    The rides data handling will be straight forward with the creation of the entity at the start of a ride and the update of the main fields when the ride is ended. The only dynamic aspect of the ride is the "paid" field that has to be updated when the payment is obtained.
- Bookings:
  - **Estimated complexity**: Low
    The information about cars bookings are static and easy to manage.
- Cars:
  - **Estimated complexity**: High
    The status of the cars are is the most critical type of data that our system will have to handle, informations about the cars dynamic and complex to retrieve.
- Assistance requests:
  - **Estimated complexity**: Low

  Easy informations to handle, static once generated(only the status changes only once).
- Fares/PriceVariations:
  - **Estimated complexity**: Low

Static informations that can be updated by the management system, easy to handle and maintain.

- Safe and RechargingArea:
  - **Estimated complexity**: Average

  Static information about the position of safe areas and dynamic updated to the number of plugs available, average complexity (high complexity only for the creation of the objects)

Table 5: ILF

ILF	Complexity	FPs
Users Data	Average	10
Rides Data	Low	7
Bookings Data	Low	7
Cars Data	High	15
Assistance Requests Data	Low	7
System parameters	Low	7
Areas Data	Low	10
Total		63

2.1.2 External Logic Files (ELFs)

The external data sources we rely on are PayPal, for the payments handling and GoogleMaps for the maps and navigation related services.

- PayPal:
  - **Estimated complexity**: Low

  Simple files containing payment informations.
- GoogleMaps:
  - **Estimated complexity**: High

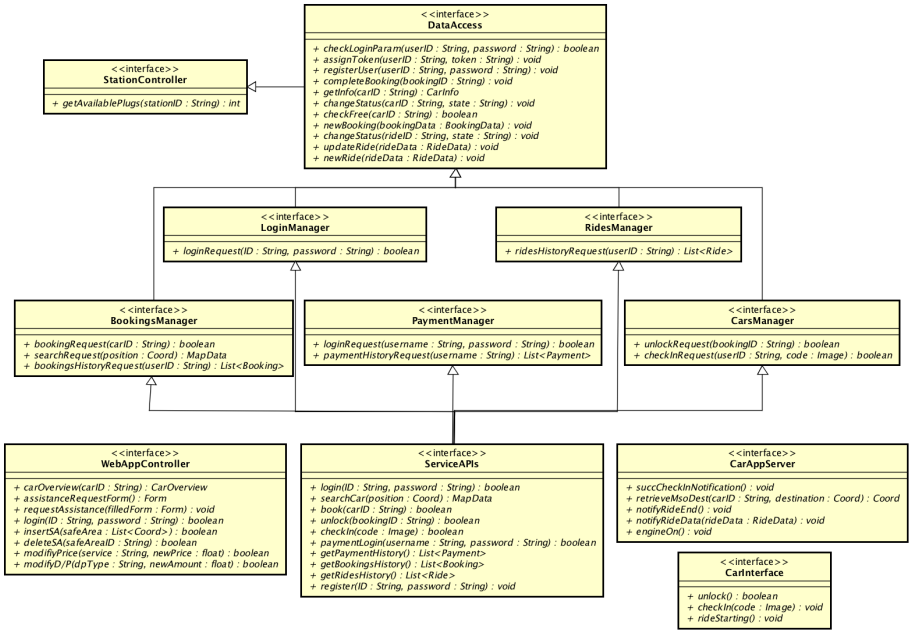
  WE heavily rely on the files produced by the GoogleMaps service, for both the navigation and the maps functionalities, we consider that both of this elements have to be considered to have an high complexity.

Table 6: ELF

ELF	Complexity	FPs
Payment info	(Very)Low	2
Navigation Data	High	10
Maps Data	High	10
Total		22

2.1.3 External Inputs (EIs)

We report here the component interfaces diagrams that will become handy in the next sections.



Our system will have to deal mainly with the following kinds of inputs:

- Login/logout of operators and users: simple operations. Both low complexity.

From the users:

- Login/logout operation: simple operations with simple operations of validations on data. Both operations have a low complexity.
- Registration: operation of an average complexity, it requires only a small amount of checks.
- QR-code check-in: simple operation but that requires some components to be handled. Average complexity.
- car locking/unlocking: simple operations that involve a good number of components. Both average complexity.
- Search for car: complex operation that needs the collaboration of a good number of components. High complexity.
- Book/un-book a car: simple operations but that change the status of car. Both average complexity.
- Payment: simple operation. Low complexity.
- Money saving option: a complex operation that involves multiple components. High complexity.

- Ride data: data that the system receive during the ride. Hard to handle, high complexity.

From the operator:

- Get car overview: complex operation of data retrieval. High complexity.
- Assistance request: not so complex operation that generates a request form to be sent to the legacy system. Average complexity.
- Insertion/delation of safe or recharging areas: they are crucial parameters for the system. Both high complexity.
- Overall view: overall view of all the cars and their position. High complexity.
- Parameters modification: simple modification but they have to be notified to the users. Average complexity.

Table 7: EIS

EIs	Complexity	FPs
Login/Logout	Low	2*3
Registration	Average	4
Qr-code check-in	Average	4
Car locking/unlocking	Average	2*4
Search for a car	High	6
Book/unbook	Average	2*4
Payment	Low	3
MSO	High	6
Ride data	High	6
Overall view	High	6
Car overview	High	6
Assistance request	Average	4
Insert/delete SA	High	2*6
Parameters modification	Average	4
Total		83

2.1.4 External Inquiries (EQs)

The main operations of simple data retrieval are:

- User profile: operation of average complexity.
- System parameters and safe area: both operation of average complexity.
- Payment history: low complexity operation, simple interaction with PayPal.
- Ride history: low complexity operation.

Table 8: EQ

EIs	Complexity	FPs
User Profile	Average	4
Payment history	Low	3
Ride history	Low	3
System parameters and SA	Average	2*4
Total		18

2.1.5 External Outputs (EOs)

The external output that our system produce are the mainly correlated with the notification system that we estimate having a cost of 12 function points. The other output is the assistance request sent to the legacy system, operation that has an average complexity.

Table 9: EOS

Eos	Complexity	FPs
Notification system	—	12
Assistance request	Average	5
Total		17

2.1.6 Overall Estimation

Our analysis has produced the following data:

Table 10: Total function points

Function Type	Value
Internal Logic Files	63
External Logic Files	22
External Inputs	83
External Inquiries	18
External Outputs	17
Total	203

With Java Enterprise edition as development platform and without considering in depth the aspects of the development of the mobile and web applications, we end up with the following results:

Average estimation: SLOC = 46 * 203 = **9338**

High estimation: SLOC = 67 * 203 = **13601**

(Function point language table from *www.qsm.com*)

Table 11: Scale factors values

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
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
RESL	little (20%)	some (40%)	oftes (60%)	generally (75%)	mostly (90%)	full (100%)
Sfj	7.07	5.65	4.24	2.83	1.41	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

2.2 Cost and effort estimation: COCOMO II

In this section we are going to use the COCOMO II model to estimate the cost and effort needed for our project.

Brief description of the Scale Factors and motivation for our choices:

- **Precedentedness**: reflects the experience of the development team in similar projects. Since our team has no experience in developing large web applications in our case will be low.
- **Flex**: reflects the degree of freedom in the development process in relation to constraints in requirements and external interfaces to use. In our case the requirements are quite strict, but we have a good degree of choice from an implementation point and in some functionalities. We think Nominal it's the correct value.
- **Team**: reflects the level of cooperation between the team members. In our case the correct value is high.
- **Risk resolution**: reflects how good the architecture as been defined and if risks has been taken into account. Since we are not expert in designing systems like this one, even if we have a good risk plan, to be conservative, we think that Nominal is the right value.

- **Process maturity**: after problems faced during the the first phases of the project have been overcome, we are q confident of our team capabilities even if we are unexperienced, but it's better to be conservative even in this case. The correct value is Nominal.

Table 12: Scale drivers values

Scale Driver	Factor	Value
Precedentedness (PREC)	Very low	6.20
Development flexibility (FLEX)	Nominal	3.04
Team cohesion (TEAM)	High	2.19
Risk resolution (RESL)	Nominal	4.24
Process maturity (PMAT)	Nominal	4.68
Total		20.35

2.2.1 Scale drivers

2.2.2 Cost drivers

2.2.3 Effort equation

2.2.4 Schedule estimation