# 2 Project size, cost and effort estimation

This section of the document provides an estimation about the size, the cost and the effort necessary for this project. On order to make a better estimation, we consider only the business logic code excluding the user interface code.

For the estimation of the project size, we use the Function Points approach that take in consideration all the features that the system provides to the user. This estimation is also related to the number of lines of code to be written in Java, useful also for effort estimation with COCOMO approach.

COCOMO approach relies on an equation based on different variables that has as basis the number of lines of code estimated in the Function Points section.

## 2.1 Size estimation: Function Points

Function points gives an estimation on the project size taking as input some symbols that represent the complexity of the functionality to rate. These symbols are Record Element Type (RET), Data Element Type (DET) and File Type Referenced (FTR) that contribute to the calculus of the complexity of each functionality of the system. The attribution of these symbols is made through the count of the types of records/data visible to the user, managed by each function. The complexity is assigned according to the tables below that take as entry these three variables.

For Internal Logic Files (ILF) and External Interface Files (EIF)

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Data Element Type (DET)** | | |
| **Record Element Type (RET)** | **1-19** | **20-50** | **51+** |
| **1** | Low | Low | Avg |
| **2-5** | Low | Avg | High |
| **6+** | Avg | High | High |

For External Output (EO) and External Inquiry (EQ)

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Data Element Type (DET)** | | |
| **File Type Referenced (FTR)** | **1-5** | **6-19** | **20+** |
| **1** | Low | Low | Avg |
| **2-3** | Low | Avg | High |
| **4+** | Avg | High | High |

For External Input (EI)

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Data Element Type (DET)** | | |
| **File Type Referenced (FTR)** | **1-4** | **5-15** | **16+** |
| **1** | Low | Low | Avg |
| **2-3** | Low | Avg | High |
| **4+** | Avg | High | High |

Unadjusted Function Points (UFP) Complexity Weights

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Complexity Weight** | | |
| **File Type** | **Low** | **Average** | **High** |
| **ILF** | 7 | 10 | 15 |
| **ELF** | 5 | 7 | 10 |
| **EI** | 3 | 4 | 6 |
| **EO** | 4 | 5 | 7 |
| **EQ** | 3 | 4 | 6 |

### 2.1.1 Internal Logic Files (ILF)

According to the IFPUG, “An ILF is a user-identifiable group of logically related data or control information maintained within the boundary of the application. The primary intent of an ILF is to hold data maintained through one or more elementary processes of the application being counted”. Internal logic files are identifiable groups of logically related data that resides entirely within the application boundary and is maintained through External Inputs. They store any kind of data that will be used in a second moment for any purpose. All the data stored in the database are actually ILFs.

Users’ data are stored as Internal Logic Files in the main database including username, password, email, birth date, driving license number and payment method (which are the most relevant data needed for suitability about driving and consequently for reserving a car). Also name and surname are included in this table for communications in case of necessity. This logic file has 1 RET and 8 DET.

Administrators and employees’ table that is composed by username, password and position. This logical file has 1 RET and 3 DETs.

Cars’ data are stored in a separate table on the database and they include plate number as unique identifier, position (stored as a couple of two floating point numbers), car state (already defined in the RASD) and all the sensors’ information (number of passengers, battery charge, plug status and doors status, necessary for the car lock/unlock function). These data are useful for the business logic to track the cars, to maintain them and also to compute a suitable parking destination (based on the destination inserted by the user) in case the user sets the money saving option. This logic file has 1 RET and 7 DET.

Also parking information are stored in the main database in a separate table: it contains Parking ID that is the unique identifier, parking type (it is a flag that distinguishes if a parking is a normal safe area or a special parking area), position, number of free slots, number of free plugs and number of available cars. These data are useful for better distribution of cars for the money saving option and for the reservation system that picks the best parking where users can take the car. This logic file has 1 RET and 6 DET.

Reservations are stored in a dedicated table on the main database. This table links the cars with the users and the attributes that compose it are reservation date, return date (this field is updated at the moment of the lock of the car or at the end of the pit stop timer, inserting the current timestamp), user, payment flag (that indicates if the reservation has been payed already) and money saving option flag. These are the main attributes used for the calculation of the drive price and the billing but there are also, start position, end position and duration of the drive for statistical purpose (used mainly to map the mostly used parking). This logic file has 3 RET and 21 DET.

Finally, the system keeps track of the user apps currently connected with a list of objects representing the clients on which it is working. This list is stored in the main memory to allow fast read and update of the list by the main system and it is useful for the observer design pattern to inform users about car states. This logic file has 1 RET and 1 DET.

According to the tables listed above the complexity of these ILFs is listed here:

|  |  |  |
| --- | --- | --- |
| **ILF** | **Complexity** | **FPs** |
| User data | Low | 7 |
| Admin data | Low | 7 |
| Car data | Low | 7 |
| Parking data | Low | 7 |
| Reservation | Avg | 10 |
| Active clients | Low | 7 |
| **Total** | | 45 |

### 2.1.2 External Interface Files (EIF)

According to the IFPUG, "An external interface file (EIF) is a user identifiable group of logically related data or control information referenced by the application, but maintained within the boundary of another application. The primary intent of an EIF is to hold data referenced through one or more elementary processes within the boundary of the application counted. This means an EIF counted for an application must be in an ILF in another application".

External Interface Files represent the data that the system will use/reference, but data that are not maintained by the system. This means that an EIF counted for an application must be in an ILF in another application.

Since they represent data handled by the main system but produced by another component (that are the user app and the car computer), all the information about GPS mapping on both user and car mobile devices and sensor detection on the car are the only EIFs of the main system.

The interactions that involve the EIFs are:

* Visualization of nearest parking (2 RETs that are user table and parking table, 2 DETs that are position attributes on both tables);
* Generation of shortest path to the selected parking for pickup (same as the previous interaction);
* Computation of the destination parking for money saving option (4 RETs that are all 4 the main tables in the database, 6 DETs that are user and money saving option in reservation, user position, car position and parking position);
* Retrieval of the car in case of parking outside of a safe area (caused by end of pit stop timer or car failure) (1 RET that is the car table and 1 DET that is the car position);
* Computation of extra fees or discounts based on the sensors’ data (4 RETs that are all 4 the main tables in the database and 6 DETs that are, car position and sensors’ data, parking position, parking plug slots and reservation money saving option flag);
* Calculation of the final amount to pay intended as the basic tariff (2 RET that are user and reservation tables, and 6 DETs that are user’s payment method, reservation user, reservation date, return date and the two flags).

The table below represents the FPs assigned for these EIFs:

|  |  |  |
| --- | --- | --- |
| **EIF** | **Complexity** | **FPs** |
| Nearest parking | Low | 5 |
| Navigator to the selected parking | Low | 5 |
| Money saving option | High | 10 |
| Car retrieval | Low | 5 |
| Extra fees and discounts | High | 10 |
| Price calculation | Low | 5 |
| **Total** | | 40 |

### 2.1.3 External Inputs (EI)

According to the IFPUG, External Inputs “are elementary processes that process data or control information that comes from outside the application boundary. The primary intent of an EI is to maintain one or more ILFs and/or to alter the behavior of the system”. All user inputs and all file feeds are external input for the main system.

Since there are mainly 3 types of user and one of them can send two different types of input, we clamp all the input types in 4 categories:

1. User through user app:
   1. Login/registration: they involve 1 FTR each and respectively 8 and 2 DETs (in this final case, the DETs are username and password);
   2. User info visualization: it involves the complete user table (1 FTR, 8 DETs);
   3. Car finding: it involves user, car and parking tables modifying or maintaining user position, car position, ID and state, parking position, number of available cars (3 FTRs, 6 DETs);
   4. Car reservation: it involves all the tables and especially the attributes username and position of the user table, car position and car state, parking position and number of available cars and all the attributes of the reservation table, since it is a new tuple insertion (4 FTRs, 11 DETs);
   5. Money saving option activation (MSO activation): it involves user, parking and reservation tables, on user position, parking position and reservation money saving option flag attributes (3 FTRs, 3 DETs);
   6. Car lock/unlock: they involve only the car table and modifies the car state attribute (1 FTR, 1 DET each);
2. User through car computer:
   1. Car pit stop/full stop: they involve respectively the car table and both car and reservation tables. It handles the attributes car state and reservation return date (1 FTR and 1 DET, 2 FTRs and 2 DETs);
3. Administrator through admin app:
   1. Removal of a user: it involves the user and the reservation tables and all their attributes but also the user app list(3 FTRs and 14 DETs);
   2. Add/remove a parking: it involves the parking table with all his attributes (1 FTR, 6 DETs each);
   3. Add/remove car: it involves the car and parking tables with all car attributes and parking’s number of available cars and free slots (2 FTR, 8 DETs each);
   4. Car maintenance: it involves the car and the parking tables modifying the car state, the parking number of free slots and number of available cars (2 FTRs, 3 DETs);
4. Employee through employee app:
   1. Car retrieval: it involves the admin and car table, using the admin position and the car position (2 FTRs, 2 DETs).

The final calculation is listed below:

|  |  |  |
| --- | --- | --- |
| **EI** | **Complexity** | **FPs** |
| Login/Registration | Low | 3x2 |
| User info | Low | 3 |
| Car finding | Avg | 4 |
| Car reservation | High | 6 |
| MSO activation | Low | 3 |
| Lock/unlock | Low | 3x2 |
| Pit stop/full stop | Low | 3x2 |
| Remove user | Avg | 4 |
| Add/remove parking | Low | 3 |
| Add/remove car | Avg | 4 |
| Car maintenance | Low | 3 |
| Car retrieval | Low | 3 |
| **Total** | | 51 |

### 2.1.4 External Inquiries (EQ)

According to the IFPUG, “An external inquiry (EQ) is an elementary process that sends data or control information outside the application boundary. The primary intent of an external inquiry is to present information to a user through the retrieval of data or control information from an ILF of EIF. The processing logic contains no mathematical formulas or calculations, and creates no derived data. No ILF is maintained during the processing, nor is the behavior of the system altered”. They represent all the data queries made by a user, that can be a common user or a privileged user like admins or employees.

There are 3 interactions that can be seen as External Inquiries:

* User information visualization explained before;
* Car info visualization by admin for maintenance;
* Car position visualization by employee for retrieval.

They’re very simple so they have low complexity and they reference 1 RTF (respectively user table and car table for the last two) and all the needed attributes of each table.

|  |  |  |
| --- | --- | --- |
| **EQ** | **Complexity** | **FPs** |
| User information | Low | 3 |
| Car information | Low | 3 |
| Car position | Low | 3 |
| **Total** | | 9 |

### 3.1.5 External Outputs (EO)

According to the IFPUG, “An external output (EO) is an elementary process that sends data or control information outside the application boundary. The primary intent of an external output is to present information to a user through processing logic other than, or in addition to, the retrieval of data or control information. The processing logic must contain at least one mathematical formula or calculation, create derived data maintain one or more ILFs or alter the behavior of the system”.

These are all very simple actions because they need to notify the user that something has happened so in the final table they will have all low complexity:

* Send the verification email;
* Notify the user that the registration was successful or not;
* Notify the user that the reservation was successful or not;
* Notify the user that the payment was successful or not;
* Notify the admin that the car add was successful or not;
* Notify the admin that the parking add was successful or not;
* Notify the employee that there’s a car to retrieve;

|  |  |  |
| --- | --- | --- |
| **EO** | **Complexity** | **FPs** |
| Verification email | Low | 4 |
| Registration notification | Low | 4 |
| Reservation notification | Low | 4 |
| Payment notification | Low | 4 |
| Car add notification | Low | 4 |
| Payment add notification | Low | 4 |
| Retrieval notification | Low | 4 |
| **Total** | | 28 |

### 2.1.6 Overall Estimation

This is the resulting table that summarize the computation of the FP:

|  |  |
| --- | --- |
| **Function Type** | **Value** |
| Internal Logic Files | 45 |
| External Interface Files | 40 |
| External Inputs | 51 |
| External Inquiries | 9 |
| External Outputs | 28 |
| **Total** | 173 |

Considering the fact that the system will be implemented in Java Enterprise Edition (J2E or JEE) without considering the implementation of the different applications for both common users and privileged users (that can be implemented only as presentation layer with no business logic) the lines of code will be between BL and BU where: