

PowerEnJoy

Design Document

A.Y 2016/2017

Francesco Tinarelli(matr:806146)

Marco Wenzel(matr:878021)

Versione1.1

**Index**

1) Introduction------------------------------------------------------------------4

1.1) Purpose---------------------------------------------------------------4

1.2) Scope------------------------------------------------------------------4

1.3) List of definitions and abbreviations---------------------------4

1.4) Reference documents----------------------------------------------5

1.5) Overview-------------------------------------------------------------5

2) Integration strategy--------------------------------------------------------6

2.1) Entry criteria-------------------------------------------------------6

2.2) Elements to be integrated----------------------------------------6

2.3) Integration test strategy-----------------------------------------8

2.4) Sequence of component/functions integration --------------9

2.4.1) Database--------------------------------------------------9

2.4.2) Central system------------------------------------------9

2.4.3) Screen----------------------------------------------------11

2.4.4) Web application---------------------------------------11

2.4.5) Mobile application------------------------------------11

3) Individual steps and test description ---------------------------------11

3.3) Integration test case I1-----------------------------------------11

3.2) Integration test case I2------------------------------------------11

3.3) Screen---------------------------------------------------------------20

3.4) Web application--------------------------------------------------20

3.5) Mobile application-----------------------------------------------21

3.6) Testing procedure for reaching the goals--------------------21

4) Individual steps and test description---------------------------------23

5) Program stubs and test data required--------------------------------24

6) Appendix--------------------------------------------------------------------24

6.1) Hour of work------------------------------------------------------24

1. Introduction

1.1 Purpose

The Introduction Test Planning Document [ITPD] describes the test planning of the integration of the components. These components are described in details in the Design Document [DD] in paragraph 2.2.

The purpose of this document is to test how these components interface each other for improving this interaction.

1.2 Scope

The scope is the same described in the Design Document [DD] in chapter 1.2.

1.3 List of definitions and abbreviations

Definitions

**Reservation:** it’s the ability to reserve an available car for up to 1 hour, provided

only for registered user. For reserve a car the user must choose the available car from the map and click on it. For use this service, user must be logged on web/mobile app.

**Reservation Timer:** this timer count how much time remain for open the car.

**Unlock Car Button:** it’s a button on the mobile app that allow the system to check the user position and the reservation for unlock the car.

**Safe Area:** every zone where it’s possible park the car safely.

**Power Grid Area:** special safe area where it’s possible park the car safely and user takes care of plugging the car into power grid if he wants.

**Plugging Time:** the time that the user has to plug the car before the system create the bill.

**Bill:** Amount of cost that the user must pay for the ride, discount and/or fee included.

**System:** it’s the new system that we want implement. The system interacts with user for providing services, External Agency for payment, cars for receive ride data. A piece of system software is installed into each car, so in this way the system and the cars can communicate and send data.

**External Agency:** It’s an agency that provides all type of payments service, when user must pay a fee or pays a ride. Interact with system for receive the amount to subtract on user bank account.

**User:** he is a client of the service. His personal data is already memorized into the database system (already registered on system), the personal data inserted: Name, Surname, E-mail, Birth Date, Driving License number, Credit card number.

**Discount:** it’s a reward for good behavior.

**Ride:** service provided by the system that allow to engine ignites and start to charging the user for a given amount of money per minute.

**Fee:** amount of money that the user pays in addiction, fee can be assigned through missed reservation or bad behaviors.

**Car:** an automobile, in each car we have a screen and a GPS for provide services. Car send to the system the data of the ride.

**Central System:** The main system of our application, it contains all the main actions. It is the controller of our MVC.

Abbreviations

**API:** Application Programming interface.

**Rasd**: Requirements analysis and specification document.

**MVC**: Model view controller.

**DD:** Design document.

**ITPD:** Integration test plan document.

1.4 Reference Documents

For creating this ITPD we used

* Structure of Document: Assignments2016/2017.pdf
* Example: TP.pdf
* Example: Integration Test Plan Example.pdf
* Verification and validation part I.pdf
* Verification and validation part II.pdf

1.5 Overview

In the second chapter the items to be tested are mentioned. A specification for each test case is given in the third chapter. The fourth chapter specifies the procedures for these test cases. In the fifth chapter the reports for all test cases are presented.

1. Integration strategy

2.1 Entry criteria

This are the entry criteria for our test integration

* Requirements analysis and documentation document and designed document are revised and completed in all their parts
* The code of the application is complete and it is working
* The code is fully tested
* The product satisfies the requirements and the chosen design

2.2 Elements to be integrated

The following diagram represents the central system of our application. It is the same diagram that is in the DD but we have color the components for better understanding the integration strategy of our testing.

We divide this diagram in 5 part:

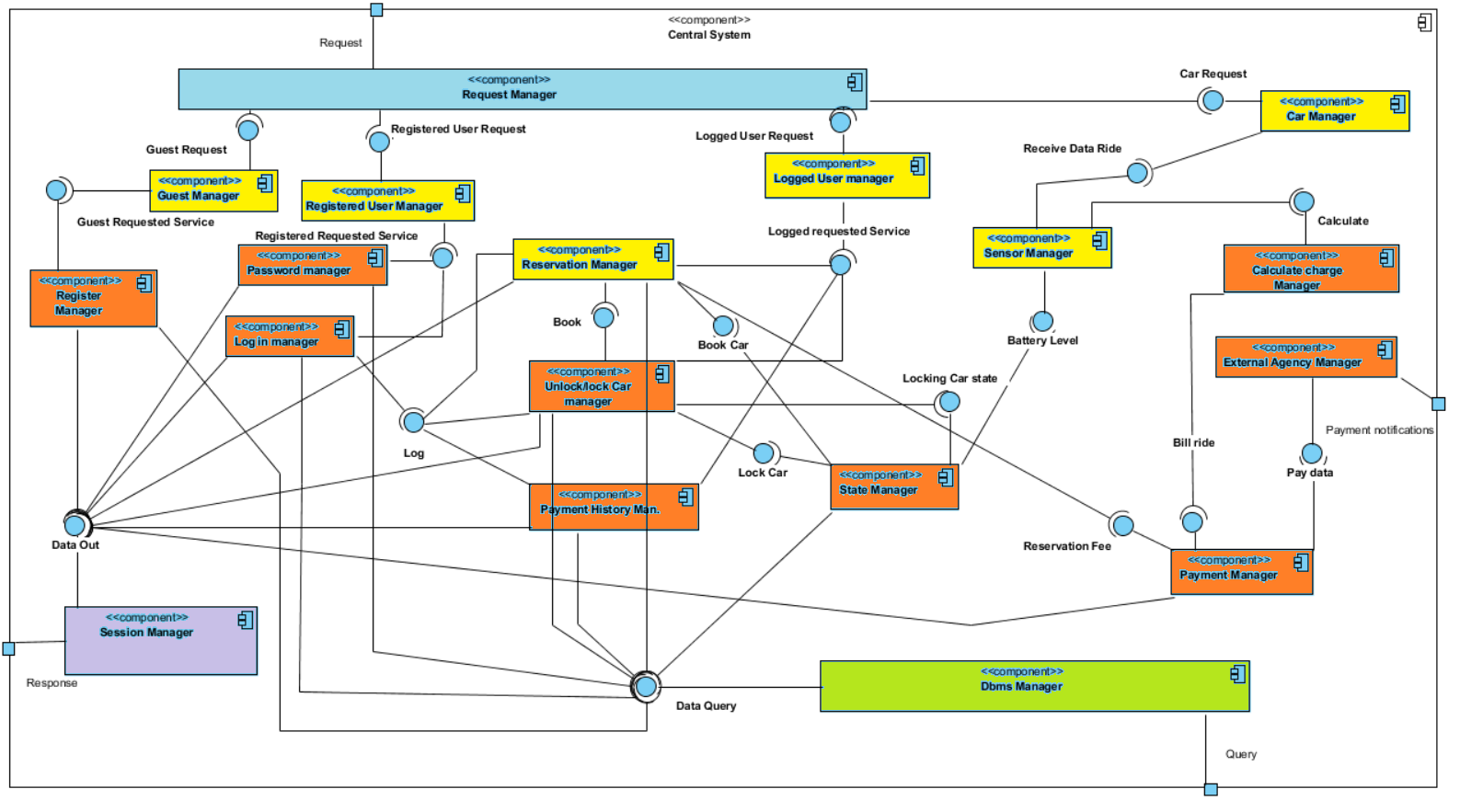
**Light blue:** for request manager, that aims to receive request from clients.

**Yellow:** for all component manager that aims to sort the various type of the requests. We treat this part all-in-one because all components have the same input requirements and have the same outputs.

**Orange:** for all component that provide a service and check the requirements for that service.

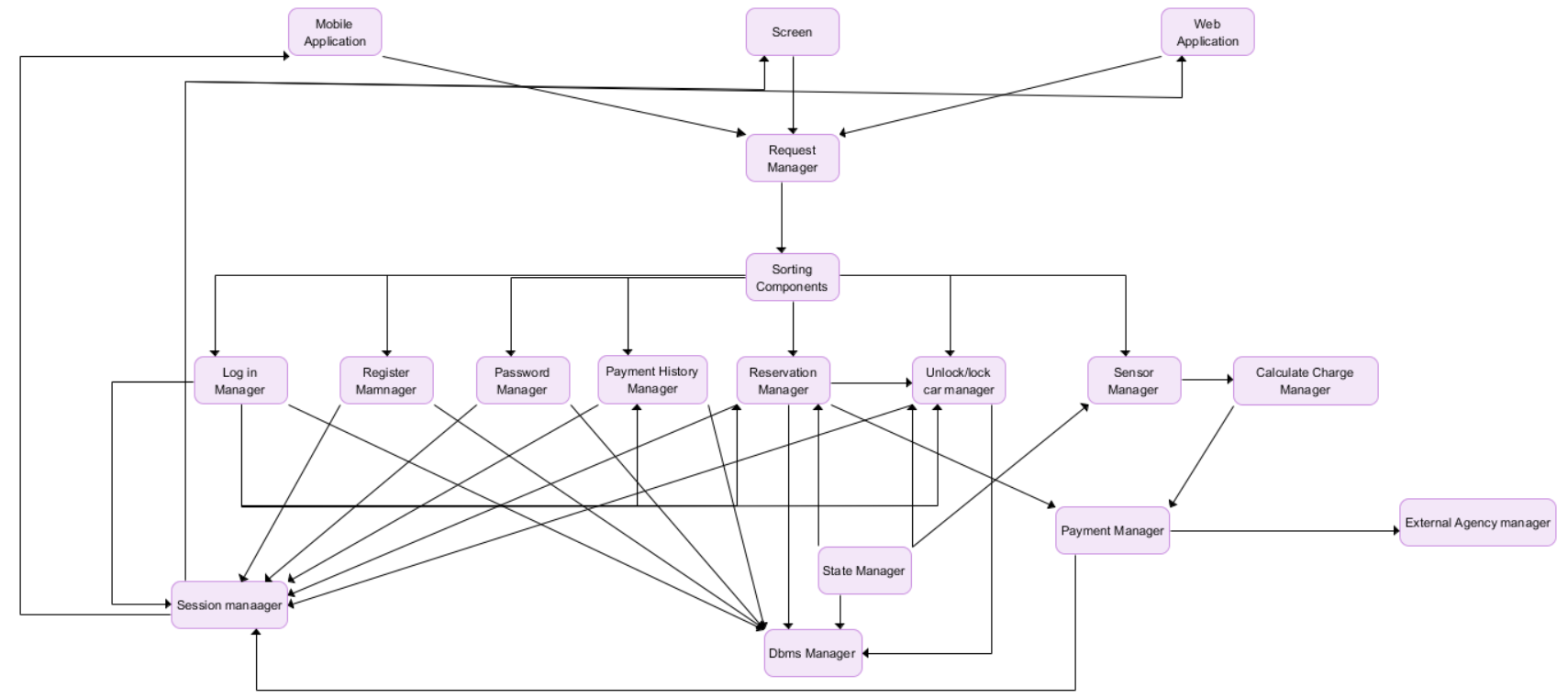
**Green:** dbms manager that aims to create the query for the database.

**Purple:** Session manager that aims to manager the response to send to clients.



In test process the Web Server tier will be together with the web application because we have thought integration test process part don’t need to test the dynamically loading of data, moreover we want to test all staff concerning the requests (creation, providing and response request and requirements checking).

The integration test will be focus on each of this component:



2.3 Integration test strategy

We decide to do a bottom-up strategy for out test integration. This strategy permit us to avoid some architectural mistakes that can we do making a top-down strategy. This is our strategy:

* We will start to test the database and its interaction with the Dbms manager.
* Test the interaction of the components of the central system
* Test the interaction of the different views of our system (screen, mobile app and web app) with the request manager of the central system.

2.4 Sequence of component/functions integration

This are the components to be integrated:

|  |  |
| --- | --- |
| ID | Components to Integrate |
| I1 | Database 🡨🡪 Central System |
| I2 | Central System🡨🡪 Central System |
| I3 | Central System 🡨🡪 Screen |
| I4 | Central System 🡨🡪 Web Application |
| I5 | Central System 🡨🡪 Mobile Application |

I2 means testing procedure into Central System, as we already said we want focalize the test procedure on the Central System part, integration among all the Central System components.

The 🡪 indicates that the left side component returns the results into the right-side components… e.g.: A🡪B so B call the A and the results of A appears in B.

Now we enter in details:

**2.4.1. Database**

|  |  |
| --- | --- |
| ID= I1 | Components |
| I1C1 | Database 🡪Dbms Manager |

**2.4.2. Central System**

|  |  |
| --- | --- |
| ID= I2 | Components |
| I2C1 | Log in Manager 🡪 Dbms Manager |
| I2C2 | Register Manager 🡪 Dbms Manager |
| I2C3 | Password Manager 🡪 Dbms Manager |
| I2C4 | Payment History Man.🡪 Dbms Manager |
| I2C5 | Reservation Manager 🡪 Dbms Manager |
| I2C6 | Unlock/Lock Manager 🡪 Dbms Manager |
| I2C7 | State Manager 🡪 Dbms Manager |
| I2C8 | Log in manager 🡪 Session Manager |
| I2C9 | Register Manager 🡪 Session Manager |
| I2C10 | Password Manager 🡪 Session Manager |
| I2C11 | Payment History Man.🡪 Session Manager |
| I2C12 | Reservation Manager 🡪 Session Manager |
| I2C13 | Unlock/Lock Manager 🡪 Session Manager |
| I2C14 | Payment Manager 🡪 Session Manager |
| I2C15 | Log in Manager 🡪 Payment History Man. |
| I2C16 | Log in Manager 🡪 Reservation Manager |
| I2C17 | Log in Manager 🡪 Unlock/Lock Manager |
| I2C18 | State Manager 🡪 Reservation Manager |
| I2C19 | State Manager 🡪 Unlock/Lock Manager |
| I2C20 | State Manager 🡪 Sensor Manager |
| I2C21 | Payment Manager 🡪 Reservation Man. |
| I2C22 | Payment Manager 🡪 Calculate Charge M. |
| I2C23 | External Agency Man 🡪 Payment Man. |
| I2C24 | Sensor Manger 🡪 Calculate Charge Man. |
| I2C25 | Reservation Manager 🡪 Unlock/Lock Man. |
| I2C26 | Sorting Components 🡪 Log in Manager |
| I2C27 | Sorting Components 🡪 Register Manager |
| I2C28 | Sorting Components 🡪 Password Man. |
| I2C29 | Sorting Components 🡪 Reservation Man. |
| I2C30 | Sorting Components 🡪 Unlock/Lock Man. |
| I2C31 | Sorting Components 🡪 Sensor Manager |
| I2C32 | Request Manager 🡪 Sorting Components |

**2.4.3. Screen**

|  |  |
| --- | --- |
| ID= I3 | Components |
| I3C1 | Screen 🡪 Request Manager |

**2.4.4. Web Application**

|  |  |
| --- | --- |
| ID = I4 | Components |
| I4C1 | Web Application 🡪 Request Manager |

**2.4.5. Mobile Application**

|  |  |
| --- | --- |
| ID = I5 | Components |
| I5C1 | Mobile Application 🡪 Request Manager |

1. Individual steps and test description

3.1. Integration test case I1

|  |  |
| --- | --- |
| Test Case Identifier | I1C1 |
| Test Case Description | Perform a query into the database |
| Test Item(s) | Dbms Manager 🡪 Database |
| Input Specification | A query to perform |
| Output Specification | Check that the query is performed and the response data are syntactically valid |
| Environmental Needs | None |

3.2. Integration test case I2

|  |  |
| --- | --- |
| Test Case Identifier | I2C1 |
| Test Case Description | Create a query into the database |
| Test Item(s) | Log in Manager 🡪 Dbms Manager |
| Input Specification | Log in Data to check |
| Output Specification | Check that the correct methods are called to create the query in dbms manager |
| Environmental Needs | None |

|  |  |
| --- | --- |
| Test Case Identifier | I2C2 |
| Test Case Description | Create a Log in query into the database |
| Test Item(s) | Log in Manager 🡪 Dbms Manager |
| Input Specification | Log in Data to check |
| Output Specification | Check that the correct methods are called to create the query in dbms manager |
| Environmental Needs | None |
|  |  |
| Test Case Identifier | I2C3 |
| Test Case Description | Create a Password Recovery query into the database |
| Test Item(s) | Password Manager 🡪 Dbms Manager |
| Input Specification | Password Data to check |
| Output Specification | Check that the correct methods are called to create the query in dbms manager |
| Environmental Needs | None |

|  |  |
| --- | --- |
| Test Case Identifier | I2C4 |
| Test Case Description | Create a Payment History query into the database |
| Test Item(s) | Payment History Man. 🡪 Dbms Manager |
| Input Specification | Payment History Data to check |
| Output Specification | Check that the correct methods are called to create the query in Dbms manager |
| Environmental Needs | None |

|  |  |
| --- | --- |
| Test Case Identifier | I2C5 |
| Test Case Description | Create a Reservation query into the database |
| Test Item(s) | Reservation Manager 🡪 Dbms Manager |
| Input Specification | Reservation Data to check |
| Output Specification | Check that the correct methods are called to create the query in Dbms manager |
| Environmental Needs | None |

|  |  |
| --- | --- |
| Test Case Identifier | I2C6 |
| Test Case Description | Create a Unlock/Lock query into the database |
| Test Item(s) | Unlock/Lock Car Man. 🡪 Dbms Manager |
| Input Specification | Unlock/Lock Data to check |
| Output Specification | Check that the correct methods are called to create the query in Dbms manager |
| Environmental Needs | None |

|  |  |
| --- | --- |
| Test Case Identifier | I2C7 |
| Test Case Description | Create a State query into the database |
| Test Item(s) | State Manager 🡪 Dbms Manager |
| Input Specification | State Data to check |
| Output Specification | Check that the correct methods are called to create the query in Dbms manager |
| Environmental Needs | None |

|  |  |
| --- | --- |
| Test Case Identifier | I2C8 |
| Test Case Description | Manage a log in session |
| Test Item(s) | Log in Manager 🡪 Session Manager |
| Input Specification | Login Data response to manage |
| Output Specification | Check that the correct methods are called to manage the query response in session manager |
| Environmental Needs | Dbms Manager |

|  |  |
| --- | --- |
| Test Case Identifier | I2C9 |
| Test Case Description | Manage a register session |
| Test Item(s) | Registration Manager 🡪 Session Manager |
| Input Specification | Registration Data to manage |
| Output Specification | Check that the correct methods are called to manage the query response in session manager |
| Environmental Needs | Dbms Manager |

|  |  |
| --- | --- |
| Test Case Identifier | I2C10 |
| Test Case Description | Manage a Password recovery session |
| Test Item(s) | Password Manager 🡪 Session Manager |
| Input Specification | Password recovery Data to manage |
| Output Specification | Check that the correct methods are called to manage the query response in session manager |
| Environmental Needs | Dbms Manager |

|  |  |
| --- | --- |
| Test Case Identifier | I2C11 |
| Test Case Description | Manage a Payment history session |
| Test Item(s) | Payment history Man. 🡪 Session Manager |
| Input Specification | Payment history Data to manage |
| Output Specification | Check that the correct methods are called to manage the query response in session manager |
| Environmental Needs | Dbms Manager |

|  |  |
| --- | --- |
| Test Case Identifier | I2C12 |
| Test Case Description | Manage a Reservation session |
| Test Item(s) | Reservation Manager 🡪 Session Manager |
| Input Specification | reservation Data to manage |
| Output Specification | Check that the correct methods are called to manage the query response in session manager |
| Environmental Needs | Dbms Manager |

|  |  |
| --- | --- |
| Test Case Identifier | I2C13 |
| Test Case Description | Manage a Unlock/Lock car session |
| Test Item(s) | Unlock/Lock car Man. 🡪 Session Manager |
| Input Specification | Unlock/lock car Data to manage |
| Output Specification | Check that the correct methods are called to manage the query response in session manager |
| Environmental Needs | Dbms Manager |

|  |  |
| --- | --- |
| Test Case Identifier | I2C14 |
| Test Case Description | Manage a Payment session |
| Test Item(s) | Payment Manager 🡪 Session Manager |
| Input Specification | Payment Data to manage |
| Output Specification | Check that the correct methods are called to manage the query response in session manager |
| Environmental Needs | Dbms Manager |

|  |  |
| --- | --- |
| Test Case Identifier | I2C15 |
| Test Case Description | Check User Data |
| Test Item(s) | Log in Manager 🡪 Payment History man. |
| Input Specification | Log in Data to check |
| Output Specification | Check that the correct methods are called to provide a first check about user data in payment history manager, if they are correct use these data for the query. |
| Environmental Needs | Log in Manager |

|  |  |
| --- | --- |
| Test Case Identifier | I2C16 |
| Test Case Description | Check User Data |
| Test Item(s) | Log in Manager 🡪 Reservation Manager |
| Input Specification | Log in Data to check |
| Output Specification | Check that the correct methods are called to provide a first check about user data in reservation manager, if they are correct use these data for the query. |
| Environmental Needs | Log in Manager |

|  |  |
| --- | --- |
| Test Case Identifier | I2C17 |
| Test Case Description | Check User Data |
| Test Item(s) | Log in Manager 🡪 Unlock/lock car man. |
| Input Specification | Log in Data to check |
| Output Specification | Check that the correct methods are called to provide a first check about user data in unlock/lock car manager, if they are correct use these data for the query. |
| Environmental Needs | Log in Manager |

|  |  |
| --- | --- |
| Test Case Identifier | I2C18 |
| Test Case Description | Notify or demand about state change |
| Test Item(s) | State Manager. 🡪 reservation Manager |
| Input Specification | Reservation Car Data |
| Output Specification | Check that the correct methods are called to manage the notify or demand about state change in reservation and state manager. |
| Environmental Needs | State Manager |

|  |  |
| --- | --- |
| Test Case Identifier | I2C19 |
| Test Case Description | Notify or demand about state change |
| Test Item(s) | State Manager. 🡪 Unlock/lock Manager |
| Input Specification | Reserved car Data |
| Output Specification | Check that the correct methods are called to manage the notify or demand about state change in unlock/lock and state manager. |
| Environmental Needs | State Manager |

|  |  |
| --- | --- |
| Test Case Identifier | I2C20 |
| Test Case Description | Notify or demand about state change |
| Test Item(s) | State Manager. 🡪 Sensor Manager |
| Input Specification | Sensor car Data |
| Output Specification | Check that the correct methods are called to manage the notify or demand about state change in sensor and state manager. |
| Environmental Needs | State Manager |

|  |  |
| --- | --- |
| Test Case Identifier | I2C21 |
| Test Case Description | Create a bill for reservation fee |
| Test Item(s) | Reservation Manager🡪 Payment Manager |
| Input Specification | Reservation car Data |
| Output Specification | Check that the correct methods are called to create a bill regarding miss reservation in payment manager. |
| Environmental Needs | Reservation Manager |

|  |  |
| --- | --- |
| Test Case Identifier | I2C22 |
| Test Case Description | Create a bill for the ride |
| Test Item(s) | Calculate Charge M.🡪 Payment Manager |
| Input Specification | Ride Data |
| Output Specification | Check that the correct methods are called to create a bill regarding ride data in payment manager. |
| Environmental Needs | Calculate Charge Manager |

|  |  |
| --- | --- |
| Test Case Identifier | I2C23 |
| Test Case Description | Send bill to external agency |
| Test Item(s) | Payment Man 🡪 External Agency man. |
| Input Specification | Reservation fee bill or ride bill |
| Output Specification | Check that the correct methods are called to send the bill. |
| Environmental Needs | Payment Manager |

|  |  |
| --- | --- |
| Test Case Identifier | I2C24 |
| Test Case Description | Send data to check various discounts |
| Test Item(s) | Sensor Manager 🡪 Calculate Charge man. |
| Input Specification | Ride data |
| Output Specification | Check that the correct methods are called to check various discounts. |
| Environmental Needs | Sensor manager |

|  |  |
| --- | --- |
| Test Case Identifier | I2C25 |
| Test Case Description | Checking the reservation distance and validity |
| Test Item(s) | Reservation manager 🡪 Unlock/lock man. |
| Input Specification | Reservation data. |
| Output Specification | Check that the correct methods are called to check the validity of the reservation. |
| Environmental Needs | Reservation Manager |

|  |  |
| --- | --- |
| Test Case Identifier | I2C26 |
| Test Case Description | Sorting various type of incoming request |
| Test Item(s) | Sorting Components 🡪 Log in manager |
| Input Specification | Request |
| Output Specification | Check that the correct methods are called to sorting the incoming request to the right component. |
| Environmental Needs | Sorting Components |

|  |  |
| --- | --- |
| Test Case Identifier | I2C27 |
| Test Case Description | Sorting various type of incoming request |
| Test Item(s) | Sorting Components 🡪 Register manager |
| Input Specification | Request |
| Output Specification | Check that the correct methods are called to sorting the incoming request to the right component. |
| Environmental Needs | Sorting Components |

|  |  |
| --- | --- |
| Test Case Identifier | I2C28 |
| Test Case Description | Sorting various type of incoming request |
| Test Item(s) | Sorting Components 🡪 Password man |
| Input Specification | Request |
| Output Specification | Check that the correct methods are called to sorting the incoming request to the right component. |
| Environmental Needs | Sorting Components |

|  |  |
| --- | --- |
| Test Case Identifier | I2C29 |
| Test Case Description | Sorting various type of incoming request |
| Test Item(s) | Sorting Components 🡪 Reservation man.. |
| Input Specification | Request |
| Output Specification | Check that the correct methods are called to sorting the incoming request to the right component. |
| Environmental Needs | Sorting Components |

|  |  |
| --- | --- |
| Test Case Identifier | I2C30 |
| Test Case Description | Sorting various type of incoming request |
| Test Item(s) | Sorting Components 🡪 Unlock/lock man. |
| Input Specification | Request |
| Output Specification | Check that the correct methods are called to sorting the incoming request to the right component. |
| Environmental Needs | Sorting Components |

|  |  |
| --- | --- |
| Test Case Identifier | I2C31 |
| Test Case Description | Sorting various type of incoming request |
| Test Item(s) | Sorting Components 🡪 Sensor Manager |
| Input Specification | Request |
| Output Specification | Check that the correct methods are called to sorting the incoming request to the right component. |
| Environmental Needs | Sorting Components |

|  |  |
| --- | --- |
| Test Case Identifier | I2C32 |
| Test Case Description | Sorting the incoming request |
| Test Item(s) | Request Manager 🡪 Sorting components. |
| Input Specification | Request |
| Output Specification | Check that the correct methods are called to sort the incoming request in base of user type. |
| Environmental Needs | Request Manager |

|  |  |
| --- | --- |
| Test Case Identifier | I2C33 |
| Test Case Description | Send responses |
| Test Item(s) | Session Manager 🡪 Mobile Application. |
| Input Specification | Response to send |
| Output Specification | Check that the correct methods are called in base of user type. |
| Environmental Needs | Dbms manager |

|  |  |
| --- | --- |
| Test Case Identifier | I2C34 |
| Test Case Description | Send responses |
| Test Item(s) | Session Manager 🡪 Screen |
| Input Specification | Response to send |
| Output Specification | Check that the correct methods are called request in base of user type. |
| Environmental Needs | Dbms manager |

|  |  |
| --- | --- |
| Test Case Identifier | I2C35 |
| Test Case Description | Send responses |
| Test Item(s) | Session Manager 🡪 web application |
| Input Specification | Response to send |
| Output Specification | Check that the correct methods are called in base of user type. |
| Environmental Needs | Dbms manager |

3.3. Screen

|  |  |
| --- | --- |
| Test Case Identifier | I3C1 |
| Test Case Description | Send(screen) and receive (central system) the requests |
| Test Item(s) | Screen 🡪 Central System |
| Input Specification | Command from Screen |
| Output Specification | Check that the correct methods are called and the client receives a notification about response |
| Environmental Needs | Car monitor |

3.4. Web application

|  |  |
| --- | --- |
| Test Case Identifier | I4C1 |
| Test Case Description | Send(screen) and receive (central system) the requests |
| Test Item(s) | Request Manager 🡪 Sorting components. |
| Input Specification | Command from Web application |
| Output Specification | Check that the correct methods are called and the client receives a notification about response |
| Environmental Needs | Pc with internet |

3.5. Mobile application

|  |  |
| --- | --- |
| Test Case Identifier | I5C1 |
| Test Case Description | Send(screen) and receive (central system) the requests |
| Test Item(s) | Request Manager 🡪 Sorting components. |
| Input Specification | Command from mobile application |
| Output Specification | Check that the correct methods are called and the client receives a notification about response |
| Environmental Needs | Mobile phone with Power enjoy application |

3.6. Testing procedure for reaching the goals

|  |  |
| --- | --- |
| Test Procedure ID | TP1 |
| Goals | Allow to register on the system |
| Procedure Steps | I4C1, I2C32, I2C27, I2C2, I1C1, I2C9 |

|  |  |
| --- | --- |
| Test Procedure ID | TP2 |
| Goals | Allow to log in on the system. |
| Procedure Steps | (I4C1 o I5C1), I2C32, I2C26, I2C1 , I1C1, I2C8 |

|  |  |
| --- | --- |
| Test Procedure ID | TP3 |
| Goals | Allow to find the locations of available cars within a certain distance from their current position or from a specified address. |
| Procedure Steps | (I4C1 o I5C1), I2C32, I2C29, I2C16, I2C18, I2C5, I1C1, I2C12 |

|  |  |
| --- | --- |
| Test Procedure ID | TP4 |
| Goals | Allow to reserve a single car for up to 1 hour. |
| Procedure Steps | (I4C1 o I5C1), I2C32, I2C29, I2C21 |

|  |  |
| --- | --- |
| Test Procedure ID | TP5 |
| Goals | Allow to pay a fee if he misses the reservation. |
| Procedure Steps | (I4C1 o I5C1), I2C32, I2C29, I2C21, I2C23, I2C14 |

|  |  |
| --- | --- |
| Test Procedure ID | TP6 |
| Goals | Allow to tell the system she’s nearby. |
| Procedure Steps | I5C1, I2C32, I2C30, I2C25, I2C6, I1C1, I2C13 |

|  |  |
| --- | --- |
| Test Procedure ID | TP7 |
| Goals | Allow to see the current charges. |
| Procedure Steps | I3C1 |

|  |  |
| --- | --- |
| Test Procedure ID | TP8 |
| Goals | Allow to reach grid stations location. |
| Procedure Steps | I3C1 |

|  |  |
| --- | --- |
| Test Procedure ID | TP9 |
| Goals | Allow to receive a discount if the user has at least two passengers. |
| Procedure Steps | I3C1, I2C32, I2C31, I2C24, I2C22 |

|  |  |
| --- | --- |
| Test Procedure ID | TP10 |
| Goals | Allow to receive a discount if the user left the car with no more than 50% battery empty. |
| Procedure Steps | I3C1, I2C32, I2C31, I2C24, I2C22 |

|  |  |
| --- | --- |
| Test Procedure ID | TP11 |
| Goals | Allow to receive a discount if the user left the car in a Power grid station and takes care of plugging the car. |
| Procedure Steps | I3C1, I2C32, I2C31, I2C24, I2C22 |

|  |  |
| --- | --- |
| Test Procedure ID | TP12 |
| Goals | Allow to receive a fee if the user left the car more than 3km from the nearest power station or left the car with more than 80% of the battery empty. |
| Procedure Steps | I3C1, I2C32, I2C31, I2C24, I2C22 |

|  |  |
| --- | --- |
| Test Procedure ID | TP13 |
| Goals | Allow to see an e-mail sent by the system with the bill ride and payment resume, at the end of this one. |
| Procedure Steps | I3C1, I2C32, I2C31, I2C24, I2C22, I2C21 ,I2C23 ,I2C14 |

|  |  |
| --- | --- |
| Test Procedure ID | TP14 |
| Goals | Allow to user to pay immediately after at the end of the ride. |
| Procedure Steps | I3C1, I2C32, I2C31, I2C24, I2C22, I2C21, I2C23, I2C14  I4C1, I2C32, I2C4, I2C1, I2C11 |

1. Tools and test equipment require

In this section, we will suggest possible tools that can be used for testing PowerEnJoy.

**Moquito:** It is used for creates mockups. This mockups perform the unit testing.

**Junit:** It is use for write repeatable testing of the unit or component we want to test.

**Manual testing:** The developers test himself the program using it. He uses a black-box test strategy to ensure the correct behavior of the system.

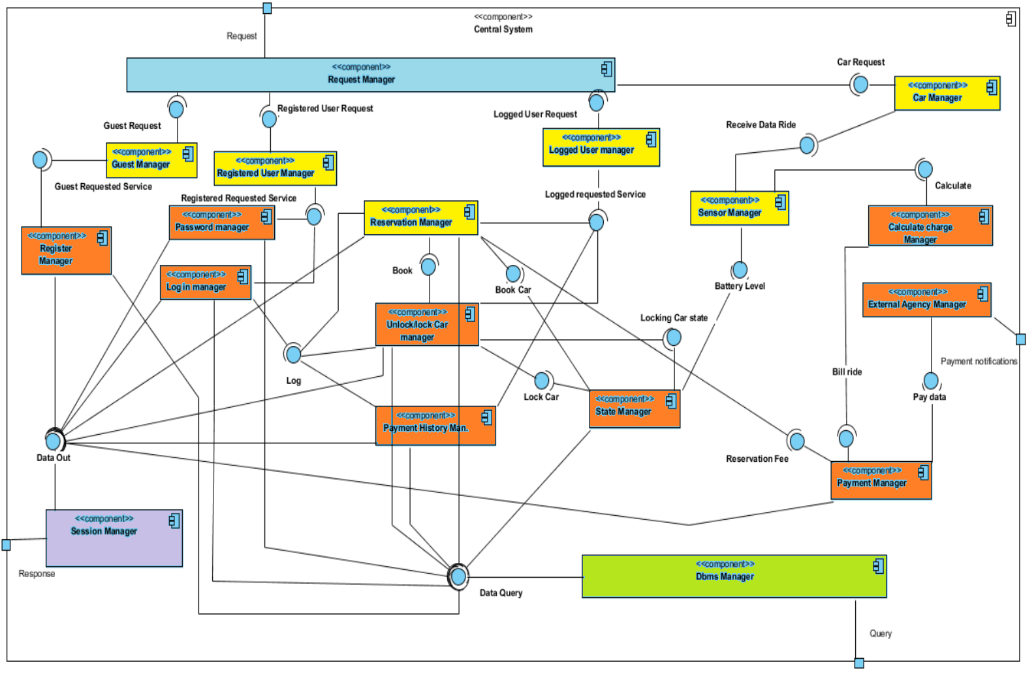
1. Program stub and test data required

In this section, we describe the stub we must use for our test integration.

Cause we decide to do a bottom-up strategy for our test integration, we won`t use the stubs, but instead of this, we use drivers.

The driver is a temporary program used to simulate the main module.

We have 5 drivers, one for color of the following image



First driver is the Green for the Dbms manager, this driver create a query for the Dbms.

Second is the Red, this driver contains the main action of our system, this action will be use with the session manager or the Dbms manager.

Third Yellow, this driver parses the request to the right component one.

Fourth Blue, this driver receives the request from the client and parses it.

Final is Purple, this driver receives the result of the action and sends them to the right client.

1. Appendix

6.1 Hour of work

**Marco Wenzel:** 10 hours.

**Francesco Tinarelli:** 10 hours.