**I**ntegration **T**est **P**lan **D**ocument



Version 1.0

Luca Santini 808710

Riccardo Remigio 874939

1. Introduction
   1. Purpose and Scope
   2. List of definitions and abbreviations
   3. List of reference documents
2. Integration strategy
   1. Entry criteria

Before starting the integration testing phase:

-the RASD and DD have to be completed and released in order to have a clear and complete picture of the structure of the software architecture and requirements to follow

-the mainly components of the software have to be developed. During the development of the components we wrote the unit tests to ensure that the single components work correctly. In this way we are sure that if there is an error in the testing execution, it is not related to the single components but to the interaction between them.

- due to the lower complexity of the software, we decided to completely develop all the components. In this way we don’t have to use stubs or drivers, therefore we are more sure that the functionalities used in the integration are the really one.

Furthermore we save the time of writing them.

* 1. Elements to be integrated

In this chapter we list all the components that have to be integrated in the testing phase, following the architecture written in the design document.

Based on the component view of the design document, it’s possible to identify the interaction between the components of the application server, which is the main high level component of our architecture.

These components are:

-Client request manager

-Reserve manager

-Profile manager

-Payment handler

-Data manager

-Car handler

-Position handler

We will test also the integration between the components above and the external interfaces given by outer services.

* 1. Integration testing strategy

We decided to implement the testing phase through the bottom-up method, because we have all the implemented code available

* 1. Sequence of component/Function integration
     1. Software integration sequence

To decide the order of the components to be tested first, we rely on the component view of the design document, respecting the bottom-up method.

In this way we follow the possible interaction between the components as shown in the diagram quoted above.

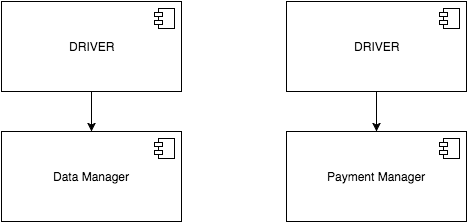
First of all we integrate the components with elementary functions, that are the components most used by the others and which have the need of less dependency.

Following the bottom-up method we used a driver for each component that we haven’t integrated yet and which is needed for testing of the actual under testing component

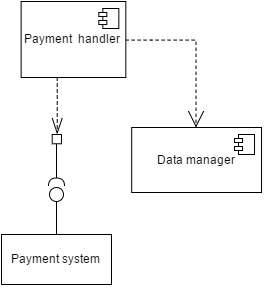
These components are:

-Data manager, which permits the access to data to the other components.

-Payment handler, which interfaces with the external payment system.



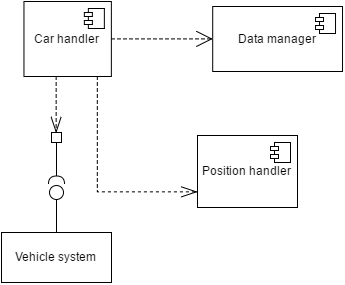
-Position handler, which gives the access to the services given by a third party software for the position and the visualization of the maps.



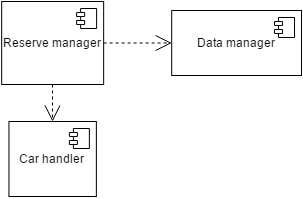
After the integration of the basic components, we can continue integrating the rest of the components of the application server, till arriving at the integration of the whole system.

So for each component we follow the lines drawn in the component diagram (which represent the possible calls to the methods of the component).

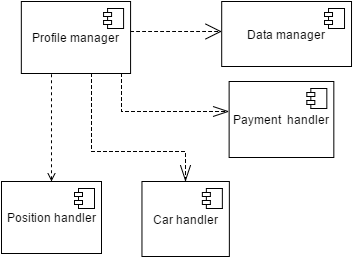
-Car handler:



-Reserve manager:



-Profile manager:



The last one is the client request manager, because it is called only by the browser of the client through an interface.

