| **N** | **SUBSYSTEM** | **COMPONENT** | **INTEGRETES WITH** |
| --- | --- | --- | --- |
| I1 | Data, Application tier | UserLogin | DBMS |
| I2 | Data, Application tier | UserRegistration | DBMS |
| I3 | Data, Application tier | Booking | DBMS |
| I4 | Data, Application tier | Reservation | DBMS |
| I5 | Application tier | UserManager | UserLogin  UserRegistration  DataManager |
| I6 | Application tier | ReservationManager | Booking  Reservation  UnlockCarManager |
| I7 | Application tier | UtilityManager | EmailSender  PaymentManager |
| I8 | Application tier | ApplicationController | UserManager  ReservationManager  UtilityManager |
| I9 | Mobile | UIManager | UIKit |
| I10 | Mobile | UIManager | Android.view |
| I11 | Mobile | GPSManager | CarLocator |
| I12 | Mobile | GPSManager | LocationListener |
| I13 | Mobile | MobileAppController | UIManager  GPSManager  ResourceManager |
| I14 | Web | WebAppController | JavaServerFaces |

| **N** | **SUBSYSTEM** | **INTEGRETES WITH** |
| --- | --- | --- |
| SI1 | APPLICATION TIER | DATA TIER |
| SI2 | MOBILE APPLICATION | APPLICATION TIER |
| SI3 | WEB TIER | APPLICATION TIER |
| SI4 | WEB BROWSER | WEB TIER |

| **Test Case Identifier** | SI1T1 |
| --- | --- |
| **Test Item(s)** | Application tier —> Data Tier |
| **Input Specification** | Typical calls to the methods of the JPA Entities, mapped with tables in the Data tier. |
| **Output Specification** | The Data tier shall respond by doing the correct queries on the test database. It must also react in the right way both if the requests are made correctly and if they come from unauthorized sources that are trying to access the data. |
| **Environmental Needs** | Complete implementation of the Java Entity Beans, Java Persistence API, Test Database, driver that calls the Java Entity Beans. |
| **Test Description** | The response will be compared with the expected output of the queries. |
| **Testing Method** | Automated with JUnit. |

| **Test Case Identifier** | SI2T1 |
| --- | --- |
| **Test Item(s)** | Mobile Application —> Application Tier |
| **Input Specification** | Typical API calls to the Application tier (REST API). |
| **Output Specification** | The Application tier shall respond accordingly to the API specification. Also, it must react correctly if the requests are malformed or maliciously crafted. |
| **Environmental Needs** | Complete implementation of the Application tier; REST API client (driver) that mocks the actual mobile client. |
| **Test Description** | The clients should make typical API calls to the application  tier; the responses are then evaluated and checked  against the expected output. The driver of this test  is a standard REST API client that runs on Java. |
| **Testing Method** | Automated with JUnit. |

| **Test Case Identifier** | SI2T2 |
| --- | --- |
| **Test Item(s)** | Mobile Application —> Application Tier |
| **Input Specification** | Multiple concurrent requests to the REST API of the application tier. |
| **Output Specification** | The business tier must answer the requests in a reasonable  time with the applied load. |
| **Environmental Needs** | Tomcat Server, fully developed application tier, Apache JMeter. |
| **Test Description** | This test case assesses whether the business tier fulfills  the performance. |
| **Testing Method** | Automated with Apache JMeter. |

| **Test Case Identifier** | SI3T1 |
| --- | --- |
| **Test Item(s)** | Web Tier —> Application Tier |
| **Input Specification** | Requests for services offered by the application tier, also  invalid ones. |
| **Output Specification** | The web tier must call the proper REST APIs or report  an error. |
| **Environmental Needs** | Tomcat Server, Web tier |
| **Test Description** | This test has to ensure the right translation from  HTTPS requests into REST APIs calls, reporting errors  when needed. |
| **Testing Method** | Automated with JUnit. |

| **Test Case Identifier** | SI3T2 |
| --- | --- |
| **Test Item(s)** | Web Tier —> Application Tier |
| **Input Specification** | Multiple concurrent API calls to the Application tier. |
| **Output Specification** | Web requests should be served without problems when a reasonable load is applied on the Application tier. |
| **Environmental Needs** | Tomcat Server, Web tier, Apache JMeter. |
| **Test Description** | This test case assesses whether the business tier fulfills  the performance |
| **Testing Method** | Automated with Apache JMeter. |

| **Test Case Identifier** | SI4T1 |
| --- | --- |
| **Test Item(s)** | web browser —> Web tier |
| **Input Specification** | Typical and well-formed HTTPS requests from client browser; incomplete, malformed and maliciously crafted requests. |
| **Output Specification** | The web tier shall display the requested pages if the requests are valid; if the requests are invalid it shall display a generic error message. |
| **Environmental Needs** | Tomcat Server, fully developed web tier, HTTP client (driver). |
| **Test Description** | This test should emulate HTTP requests from typical users of the service and also incorrect requests. |
| **Testing Method** | Automated with JUnit. |

| **Test Case Identifier** | SI4T2 |
| --- | --- |
| **Test Item(s)** | web browser —> Web tier |
| **Input Specification** | Multiple concurrent requests to the web server. |
| **Output Specification** | Web pages should be served without problems when a reasonable load is applied on the web server. |
| **Environmental Needs** | Tomcat Server, fully developed web tier, Apache JMeter. |
| **Test Description** | This test case assesses whether the web tier fulfills the performance |
| **Testing Method** | Automated with Apache JMeter. |

| **Test Case Identifier** | I1T1 |
| --- | --- |
| **Test Item(s)** | UserLogin —> DBMS |
| **Input Specification** | Typical queries on database tables |
| **Output Specification** | The queries return the correct results. |
| **Environmental Needs** | Tomcat Server, Test Database, driver for the JavaEntity Beans. |
| **Test Description** | The purpose of these tests is to check that the correct methods of the Entity Beans are called, and that they execute the correct queries to the DBMS. |
| **Testing Method** | Automated with JUnit. |

| **Test Case Identifier** | I2T1 |
| --- | --- |
| **Test Item(s)** | UserRegistration—> DBMS |
| **Input Specification** | Typical queries on database tables |
| **Output Specification** | The queries return the correct results. |
| **Environmental Needs** | Tomcat Server, Test Database, driver for the JavaEntity Beans. |
| **Test Description** | The purpose of these tests is to check that the correct methods of the Entity Beans are called, and that they execute the correct queries to the DBMS. |
| **Testing Method** | Automated with JUnit. |

| **Test Case Identifier** | I3T1 |
| --- | --- |
| **Test Item(s)** | Booking—> DBMS |
| **Input Specification** | Typical queries on database tables |
| **Output Specification** | The queries return the correct results. |
| **Environmental Needs** | Tomcat Server, Test Database, driver for the JavaEntity Beans. |
| **Test Description** | The purpose of these tests is to check that the correct methods of the Entity Beans are called, and that they execute the correct queries to the DBMS. |
| **Testing Method** | Automated with JUnit. |

| **Test Case Identifier** | I4T1 |
| --- | --- |
| **Test Item(s)** | Reservation —> DBMS |
| **Input Specification** | Typical queries on database tables |
| **Output Specification** | The queries return the correct results. |
| **Environmental Needs** | Tomcat Server, Test Database, driver for the JavaEntity Beans. |
| **Test Description** | The purpose of these tests is to check that the correct methods of the Entity Beans are called, and that they execute the correct queries to the DBMS. |
| **Testing Method** | Automated with JUnit. |

| **Test Case Identifier** | I5T1 |
| --- | --- |
| **Test Item(s)** | UserManager —> UserLogin, UserRegistration, DataManager |
| **Input Specification** | Methods call from UserManager to UserLogin, UserRegistration, DataManager to manage and update the information of the User. |
| **Output Specification** | The User information must be correct and up-to-date. |
| **Environmental Needs** | Tomcat Server. |
| **Test Description** | Verify that the information is correctly updated and that it refers  to the correct user. Control that the user’s information is persistently updated. |
| **Testing Method** | Automated with JUnit. |

| **Test Case Identifier** | I6T1 |
| --- | --- |
| **Test Item(s)** | ReservationManager —> Booking, Reservation, UnlockingCar |
| **Input Specification** | Methods call from ReservationManager to Booking, Reservation, UnlockingCar to update Reservation and Bookings’ status and to find an available car in a specified parking, and to unlock a car. |
| **Output Specification** | The reservation must be correctly updated without duplicating elements and the correct first available Car must be returned and removed from the ones available. The Booking status must be updated and the information of the user must be correctly checked when he try to unlock a car. |
| **Environmental Needs** | Tomcat Server |
| **Test Description** | The test aims to verify that the ReservationManager requests  are correctly satisfied by Booking, Reservation, UnlockingCar. |
| **Testing Method** | Automated with JUnit. |

| **Test Case Identifier** | I7T1 |
| --- | --- |
| **Test Item(s)** | UtilityManager —> EmailSender, PaymentMethod |
| **Input Specification** | Methods call from UtilityManager to the EmailSender in order to guarantee a right email authentication process. and also Method call from UtilityManager to the PaymentMethod in order to guarantee a right payment. |
| **Output Specification** | The email authentication process and payment process must be correctly handled. |
| **Environmental Needs** | Tomcat Server, e-mail sender and receiver and payment tools. |
| **Test Description** | Assure that a user can properly verify his/her email address in order to start using the system functionalities. and Assure that a payment is properly and automatically done when a user finish to use the car. |
| **Testing Method** | Automated with JUnit |

| **Test Case Identifier** | I8T1 |
| --- | --- |
| **Test Item(s)** | ApplicationController —> UserManager, ReservationManager,  UtilityManager |
| **Input Specification** | Request from ApplicationController to UserManager, ReservationManager, UtilityManager for the  functionalities offered. |
| **Output Specification** | the concurrency between the request must be properly  managed and the ApplicationController has to be able to provide the right functionality carrying out the proper request. |
| **Environmental Needs** | Tomcat Server |
| **Test Description** | Multiple requests for the UserManager, ReservationManager, UtilityManager SessionBeans have to be simultaneously carried out, in order to ensure that the users have no concurrency trouble. |
| **Testing Method** | Automated with JUnit |

| **Test Case Identifier** | I9T1 |
| --- | --- |
| **Test Item(s)** | UIManager —> UIKit |
| **Input Specification** | Methods call from UIManager to the UI elements, to display output data and change their status. |
| **Output Specification** | The view shall change accordingly and display the  output data. |
| **Environmental Needs** | Xcode, iOS Simulator. |
| **Test Description** | Verify that the bindings of the view items are correctly set in the controller and that the view actually changes and responds to method calls. Check that the output is displayed correctly. |
| **Testing Method** | Automated (iOS testing suite), manual testing on  physical devices. |

| **Test Case Identifier** | I9T2 |
| --- | --- |
| **Test Item(s)** | UIManager —> UIKit |
| **Input Specification** | Perform (or simulate) gestures on the UI elements. |
| **Output Specification** | The controller shall receive the actions and log them. |
| **Environmental Needs** | Xcode, iOS Simulator. |
| **Test Description** | Check that the gestures perform the correct actions on the controller. |
| **Testing Method** | Automated (iOS testing suite), manual testing on physical devices. |

| **Test Case Identifier** | I10T1 |
| --- | --- |
| **Test Item(s)** | UIManager —> android.view |
| **Input Specification** | Methods call from UIManager to the UI elements, to display output data and change their status. |
| **Output Specification** | The view shall change accordingly and display the output data. |
| **Environmental Needs** | Android Emulator. |
| **Test Description** | Verify that the bindings of the view items are correctly set in the controller and that the view actually changes and responds to method calls. Check that the output is displayed correctly. |
| **Testing Method** | Automated (Android testing suite), manual testing on  physical devices. |

| **Test Case Identifier** | I10T2 |
| --- | --- |
| **Test Item(s)** | UIManager —> android.view |
| **Input Specification** | Perform (or simulate) gestures on the UI elements. |
| **Output Specification** | The controller shall receive the actions and log them. |
| **Environmental Needs** | Android Emulator. |
| **Test Description** | Check that the gestures perform the correct actions on the controller. |
| **Testing Method** | Automated (Android testing suite), manual testing on physical devices. |

| **Test Case Identifier** | I11T1 |
| --- | --- |
| **Test Item(s)** | GPSManager —> CarLocation |
| **Input Specification** | Calls to the CarLocation framework methods to get location data of the car. |
| **Output Specification** | Car location data or a meaningful error status shall be returned. |
| **Environmental Needs** | Xcode, iOS Simulator. |
| **Test Description** | The purpose of the test is to check that our controller  (GPSManager) can correctly get the position from the  corresponding iOS API. Error statuses shall also be  checked. |
| **Testing Method** | Automated (iOS testing suite). |

| **Test Case Identifier** | I12T1 |
| --- | --- |
| **Test Item(s)** | GPSManager —> LocationListener |
| **Input Specification** | Calls to the Android Location framework methods to get location data of the user. |
| **Output Specification** | User location data shall be returned, or a meaningful error status. |
| **Environmental Needs** | Android Emulator. |
| **Test Description** | The purpose of the test is to check that our controller (GPSManager) can correctly get the position from the  corresponding Android API. Error statuses shall also be checked. |
| **Testing Method** | Automated (Android testing suite). |

| **Test Case Identifier** | I13T1 |
| --- | --- |
| **Test Item(s)** | MobileApplicationController —> UIManager, GPSManager, ResourceLoader |
| **Input Specification** | Calls to GPSManager methods to get the user’s location.  Load application resources (images, sounds, data) from ResourceManager. |
| **Output Specification** | The location data shall be returned from GPSManager  in a suitable format, or an exception shall be raised if the location data is not available. ResourceManager should provide the required resources without errors. |
| **Environmental Needs** | Xcode, iOS Simulator, Android Emulator. |
| **Test Description** | GPSManager should be able to return the correct  GPS data in a universal and consistent format independently  from the architecture (iOS or Android). ResourceLoader is responsible for the retrieval of the resources stored into the application bundle. This test aims to assessing that all the resources can be accessed without errors by the mobile application. |
| **Testing Method** | Automated (Android and iOS testing suites). |

| **Test Case Identifier** | I14T1 |
| --- | --- |
| **Test Item(s)** | WebApplicationController —> JavaServerFaces |
| **Input Specification** | WebController is given the typical output to be displayed  on the web page. |
| **Output Specification** | JavaServerFaces shall display the required output in  a correct way. |
| **Environmental Needs** | Tomcat Server, Stub of the Application Tier of the third part system to provide the output data. |
| **Test Description** | The purpose of this test case is to check if JSF can communicate correctly with the WebApplicationController bean. |
| **Testing Method** | Automated with JUnit. |