The following table associate the requirements defined in the RASD document into the design component delineated in this document.

In the left column are listed all the requirements related to the goals, while in the right column there are the main components, as described in the component diagram and in the architectural style section, that allows that particular system functionality.

Note that components that are obviously required for almost all the functionalities, like the *Web Server* or the *Application Server*, sometimes will be omitted in order to preserve readability. The *Database Manager* won’t appear in this table for the very same reason.

|  |  |
| --- | --- |
| ***Goals / Requirements*** | **Design Components** |
| **GOAL G1** |  |
| [R1] Customers should be able to access the service through both the web and the mobile application, even at the same time. | Web Server  Application Server  (Three tier architecture) |
| [R2] Customers must be able to register to the taxi service from the mobile or web homepage. | Web Server  Application Server  (Three tier architecture) |
| [R3] Only registered customers can access MyTaxiService’s services. | Web Server  Application Server  Users Manager  (Three tier architecture) |
| [R4] The system should allow the log out functionality. | Web Server  Application Server  Users Manager  (Three tier architecture) |
| *Notes* | The servers specified in the architecture description and the internal component “Users Manager” verify and eventually allow the costumer to access the system functionalities. |
|  | |
| **GOAL G2** |  |
| [R1] Only registered customers can request a taxi ride. | Web Server  Application Server  Users Manager  (Three tier architecture) |
| [R2] Customers must insert a valid origin location in order to request a ride. | Rides Manager – Checks the syntactic correctness of the inputs |
| [R3] The system will not allow more than a request if the previous one (either request or reservation) has not been accomplished yet. | Rides Manager |
|  | |
| **GOAL G3** |  |
| [R1] The system should allow taxi reservations for a specific path communicated by the customer. | Rides Manager |
| [R2] The system must not allow overlaps between reservations (or requests) made by the same customer. | Rides Manager |
| [R3] The system allows reservations only 2 hours before the time and date specified by the customer. | Rides Manager |
| [R4] The system will assign a taxi driver for the reserved ride 10 minutes before the time and date specified by the customer. | Rides Manager |
|  | |
| **GOAL G4** |  |
| [R1] Taxi drivers should be able to communicate their current availability state to the system. | Message Broker – communicate the change of status |
| [R2] If available, taxi drivers should be able to receive incoming requests. | Ride Manager – provides the requests  Queue Manager – select the taxi driver  Message Broker – send the requests |
| [R3] After receiving an incoming request, the taxi driver should be able to either confirm or not his intention to take charge of the request. | Message Broker – receive and |
| [R4] Taxi drivers must be able to log in the mobile application with preassigned credential and be identified as drivers. | Application Server  Users Manager |
| [R5] At the end of their workshift, taxi drivers must be able to log out of the mobile application in order to communicate to the system that they are no longer active. | Application Server  Users Manager |
|  | |
| **GOAL G5** |  |
| [R1] The system should always search an available taxi giving maximum priority to the taxi zone related to the request and lower priority to the immediate near zones. Any other taxi zone should be ignored. | Queue Manager – uses the breadth first algorithm to find the zones and queues |
| [R2] If no taxis are available in the zones specified in the previous requirement, the system should put the request on hold and periodically check again the taxi availability. | Ride Manager  Queue Manager  Message Broker – tells the customer that no taxis are available |
|  | |
| **GOAL G6** |  |
| [R1] The system should send updates through email and/or in-app notification, as specified by the customer. | Message Broker |
| [R2] Absence of taxis available, reservations overlaps, taxi average waiting time and taxi assigned to customers are events that must be notified to the customer. | Ride Manager – provide the events  Message Broker – sends the events |
|  | |
| **GOAL G7** |  |
| [R1] Customers must leave a valid phone number in order to complete the registration phase. | Users Manager – check the consistency of the fields |
| [R2] Taxi drivers must be able to access to the customer’s phone number when the system has paired them. | Message Broker – sends the request and all the related information |
| [R3] Customers must receive the taxi drivers’ contact number after the system has paired them. | Message Broker – sends the updates of a request and the related data |
| [R4] Customers must receive the taxi code in order to be able to recognize its driver. | Message Broker – sends the updates of a request and the related data |
|  | |
| **GOAL G8** |  |
| [R1] Customers can cancel a request or reservation only if it has not been assigned to a taxi driver yet. | Rides Manager |
| [R2] Customers must be able to visualize the list of all their requests and reservations. | RidesManager  Database Manager |
|  | |
| **GOAL G9** |  |
| [R1] Administrators must be able to create a taxi driver’s account. | Application Server  Users Manager |
| [R2] Administrators must be able to delete a taxi driver’s account. | Application Server  Users Manager |
| [R3] Administrators must be able to change the status of taxi driver. | Application Server  Rides Manager  Users Manager |
| [R4] Administrators must be able to change the status of a ride. | Application Server  Rides Manager |
| *Notes* | The application server provide Admins the direct access to the Users Manager. |
|  | |
| **GOAL G10** |  |
| [R1] Customers and taxi drivers must be able to visualize, both in the mobile and web application, a support phone number which they can call to obtain assistance. | Web Server  Application Server |

Notice also that the component design has partially satisfied some of the *Non Functional Requirements*: for example, the ones about the reactivity of the application are now simply achievable through our light-oriented system.