

1 Overall Description

1.1 Product Perspective

As stated in ??

The product we will release is composed by four main software applications.

- A **Web Application (User)** addressed to the *users* to use our service. This application must interface mainly with the **Back-end Application** and with Google's Maps API.
- A **Mobile Application (User)** addressed to the *users* to use our service and available for Android, iOS, Windows Mobile and Blackberry. This application must interface mainly with our **Back-end Application** and with Google's Maps API
- A **Mobile Application (Taxi Driver)** addressed to the *taxi drivers* to use our service and available for Android, iOS. This application must interface mainly with our **Back-end Application** and with Google's Maps API
- A **Back-end Application** that will handle all the business logic and that must interface mainly with Google's Maps API and with a MongoDB database .

1.1.1 User Interfaces

1.1.2 Hardware Interfaces

Both the *Mobile Application (User)* and the *Mobile Application (Taxi Driver)* must interface with the GPS module and with the Network module.

1.1.3 Software Interfaces

Web Application (User)

- MyTaxyService API
 - Mnemonic : Back-end
- Google Maps API
 - Mnemonic : Google Maps API
 - Version Number : V3

- Source : <https://developers.google.com/maps/documentation/javascript/>

Mobile Application (User and Taxi Driver)

- MyTaxyService API
 - Mnemonic : Back-end
- Google Maps API
 - Mnemonic : Google Maps API
 - Version Number : V3
 - Source : <https://developers.google.com/maps/>
- Android SDK
 - Mnemonic : Android
 - Version Number : 6.0
 - Source : <http://developer.android.com/sdk/index.html>
- iOS SDK
 - Mnemonic : iOS
 - Version Number : 9.2
 - Source : <https://developer.apple.com/ios/download/>
- Windows Mobile SDK
 - Mnemonic : Windows Mobile
 - Version Number : 6.5
 - Source : <http://www.microsoft.com/en-us/download/details.aspx?id=17284>
- BlackBerry SDK
 - Mnemonic : BlackBerry
 - Version Number : 10
 - Source : <https://developer.blackberry.com/>

Back-end Application

- Node.js API
 - Mnemonic : Node.js
 - Version : 4.2.1
 - Source : <https://nodejs.org/api/>
- MongoDB API
 - Mnemonic : MongoDB
 - Version : 3.0
 - Source : <https://docs.mongodb.org/manual/>
- Google Maps API
 - Mnemonic : Google Maps API
 - Version Number : V3
 - Source : <https://developers.google.com/maps/documentation/javascript/>
- Javascript API
 - Mnemonic : Javascript
 - Source : <https://developer.mozilla.org/en/docs/Web/API>

1.1.4 Communication Interfaces

Web Application (User)

Every application must interface with the Internet network. This interface is handled by the operative systems and not by The applications themselves.

1.1.5 Memory Constrains

Mobile Applications (User and Taxi Driver) and Web Application

The Mobile and Web Applications can not exceed 75MB of RAM usage.

Back-End Application The Back-End application can not exceed 15GB of total RAM usage.

1.1.6 Site Adaptation Requirements

Software Adaptation

Every Mobile Application must be developed according to the platforms design guidelines.

1.2 Product Functions

This section highlights the main product functions sorted by application.

1.2.1 Web Application and Mobile Application (User)

- [F1] The app must allow the user to book a taxi on the spot.
- [F2] The app must allow the user to reserve a taxi ride up to 7 days before the chosen date.
- [F3] The app must allow the user to share his ride with other people whose destination is included in the ride path.

1.2.2 Mobile Application (Taxi Driver)

- [F4] The app must provide position updates to the *back-end application*
- [F5] The app must allow the taxi driver to accept or refuse a ride request.
- [F6] The app must allow the taxi driver to report issues that could delay his arrival (traffic jam, car faults, etc ...)
- [F7] The app must allow the taxi driver to overview a history of his last rides.
- [F8] The app must allow the taxi driver to report a absent user.

1.2.3 Back-End Application

- [F9] The application must provide an interface for the registration process of users and taxi drivers
- [F10] The application must handle the request queue using a fair policy [Rif needed]
- [F11] The application must provide an accurate [rif needed] estimate of the total cost and time of a ride

[F12] The application must provide an interface to allow an operator to perform all basics CRUD operations

1.3 User Characteristics

This section identifies all the different users of the product.

User [rif needed] The user has to be at least 13 years old.
Has to be familiar with standard mobile applications interfaces.

Taxi Driver [rif needed] The driver has to be a regularly licensed taxi driver.

The driver has to be familiar with standard mobile applications interfaces and is required to own a compatible device [rif needed]

Maintenance Guy [rif needed] The maintenance guy must be able to debug and identify issues in a full stack environment. The maintenance guys must be familiar with the application requirements and the overall infrastructure and design.

Support Operator The support operator must be familiar with the overall application design and has to know how to resolve common user's issues.

1.4 Constraints

1.4.1 Regulatory Policies

- **Privacy Policy** Data should be collected and stored following the privacy policy guidelines provided by Mozilla Foundation [rif needed] https://developer.mozilla.org/en-US/Marketplace/Publishing/Policies_and_Guidelines/Privacy_policies
- **Taxi Driver Policy** The client application for taxi drivers must be accessible only by registered and regularly licensed driver.[rif needed]

1.4.2 Hardware Limitations

Mobile Application (User) The Mobile Application (User) must be developed in order to be available for the 80% of the devices currently on the market for each different platform.

Web Application (User) The Web Application must be developed in order to be available for the 100% of devices that support the latest version [rif needed] of most used browser [rif needed]

1.4.3 Interfaces to other applications

Mobile Application (User) The developer of the Mobile Application (User) must follow the design guideline [rif needed] of the platform and has to interface with the respective Developer Console and make sure that the app will be accepted and published.

The application must interface with Google's Maps API [rif needed].

Web Application (User) The application must interface with Google's Maps API [rif needed].

Mobile Application (Taxi Driver) The application must interface with Google's Maps API [rif needed].

Back-end Application The application must interface with Google's Maps API [rif needed].

1.4.4 Parallel Operations

Back-end Application The Back-end Application must handle parallel request from multiple users without generating conflicts or inconsistency.

The system must decline a request from a user if it finds another active request from the same user.

1.4.5 Reliability Requirements

Back-end Application This requirement of reliability and availability refers to the total time the system is available for the applications of the end users.

The system must reach a 99.99% uptime (corresponding to a inactivity time of 50 minutes/year).

1.4.6 Safety and Security

Mobile Application (Taxi Driver) The application must include an automatic night mode and gallery mode in order not to compromise the driver security [rif needed]

1.5 Assumptions And Dependencies

1.5.1 Domain Assumptions

We assume that the system will be deployed in a city of about 1.5M inhabitants with about 4800 regular licensed taxi drivers.

We assume that at launch time the end user mobile application will reach an audience of about 100k users while the web application will reach an audience of about 10k users.

We assume that the city is divided in taxi zones.

1.5.2 Taxi Driver Assumptions

We assume that every taxi driver will have a compatible device (personal or provided by the project client) [if needed] with a working GPS module.

1.5.3 User Assumptions

We assume that every user will have a device with an enabled Internet connection.

We also assume that the location provided by the user (manually or automatically via GPS) is always correct.