1   Overall Description

1.1   Product Perspective

The platform aims at replacing the traditional means of communication between taxi drivers and customers, and it is therefore designed as a stand-alone entity that is able to operate with a minimum set of dependencies towards external systems.

1.1.1   Software Interfaces

- **Google Maps API**: used in both the back-end and the taxi-side applications, this free API is necessary to elaborate routes and display maps. The system interacts with the API through standard GET requests.

  - Mnemonic: map rendering and route elaboration
  - Documentation: https://developers.google.com/maps/

- **GPS**: this well known system is required by the mobile applications to get information about the users' position. The interaction happens through the standard GPS protocol provided by the OS of the users' devices.

  - Mnemonic: user positions gathering
  - Documentation: refer to OS-specific documentation

- **Red Hat Enterprise Linux**: operating system run on the back-end server.

  - Mnemonics: back-end OS
  - Documentation: https://access.redhat.com/documentation/en/red-hat-enterprise-linux/

- **Apache WebServer**: software bundle run on the back-end, which is used to allow interaction between the back-end system and the user-side application. It is used to handle standard HTTP requests from the customer-side web application and to provide CGI services to all the user-side applications.

  - Mnemonics: web-server
  - Documentation: https://httpd.apache.org/docs/2.4/

- **SQLite**: DBMS software component run on the back-end, which is used to store user data in a non-resource-intensive fashion.

  - Mnemonics: DBMS
  - Documentation: https://www.sqlite.org/docs.html

1.1.2 User interface

1.1.3 Device support

The back-end system is designed to be run exclusively on usual x64 server hardware and may therefore require fine-tuning of its environment, but needs to be flexible for hardware scaling (such as replication) in order to ensure its longevity.

The user-side mobile applications, on the other hand, are designed to be run on standard SOC architectures like the vast majority that is found on smartphones and tablets, and take into account the differences in the mobile operating systems that can be run on the SOC.

Finally, the customer-side web application does not have specific device requirements, since it relies on a different level of abstraction (web browser) which is not to be considered as a hardware interface.

1.2 Product functions

The fully developed platform should implement the following high level functions.

- **Customer-side application**:

  [F1] Allow users to request a taxi and receive result of request

  [F2] Allow users to manage their profile

  [F3] Allow users to issue taxi reports

- **Taxi-side application**:

  [F4] Allow users to manage incoming requests and send back confirmations

  [F5] Allow users to issue customer reports

1.3 Actors and related functionalities

The platform may have different functional and non-functional requirements based on the user that is interacting with it. Throughout this document, the set of these different requirements has been referred to with the terminology specified in **??** (e.g. taxi-side application, customer-side application, etc.).

This section is aimed at giving a high level description of the platform, based on the different actors that participate in its use:

[A1] **Guest**: any person who wishes to start a session with the platform through one of the user-side applications. [A1] actors may:

- Register a profile into the platform
- Log into the platform

After completing a login, [A1] actors may become either of the two following types of actor.

[A2] Customer: a person as defined in **??**. Based on what actions [A2] completes, they can be further specialized in:

  [A2.1] Logged-in Customer: an [A1] actor who logged in to the platform with a customer profile. [A2.1] actors may:

   * Issue a taxi request
   * View and edit their profile

  [A2.2] Passenger: an [A2.1] actor who issued a taxi request. [A2.2] actors may:

   * Cancel or edit the issued taxi request
   * Issue a taxi report

[A3] Taxi driver: a person as defined in **??**. [A3] actors may:

  - Handle (accept or refuse) a taxi request issued by an [A2] actor.
  - Edit their availability status
  - Issue a customer report
  - Notify a technical problem

1.3.1 User characteristics

Users do not need specific competences to use the app other than a basic knowledge of web browsing (customers only) and of their mobile operating system (both customers and taxi drivers).

1.4 Constraints

The development of the platform will be limited by the following constraints to which the system must be subject to. These are not listed as part of the specific requirements of the platform because they stem from circumstances beyond the designers' control, and are not directly part of the customer's requests.

[C1] **Privacy policy**: processing of user data, be it personal or related to the use of the platform, is to be carried out according to the national laws and the ways and means of said processing must be clearly defined and available to the users at any time.

[C2] **Taxi licenses**: the use of the taxi-side application and the related access to the platform, is intended exclusively for taxi drivers officially recognized by the town's appropriate taxi organizations.

[C3] **Data integrity and data security**: users' personal data must be stored according to specific standards that guarantee their integrity and security. Further analysis of this constraint will be carried out in the following sections of the document.

[C4] **Legal waiver of responsibility**: due to the nature of the platform's domain, the use of the platform can be associated to car accidents and other related possible injuries; users are therefore required to accept a legal disclaimer, in order to use the platform.

1.5 Domain Assumptions

Some domain assumption have to be introduced in order to further define the scope of the project, and to clearly state under which assumptions the development should proceed in those areas that are not explicitly covered by the initial customer's specification.

The mentioned assumptions are the following:

[AS1] It is assumed that the user-side devices are capable of running the intended user-side applications, according to the descriptions in section 1.1.3. Hardware or software compatibilities will not be discussed further.

[AS2] It is assumed that a public database exists, which can be exploited to perform checks on the official status of taxi drivers in order to ensure the satisfaction of constraint [C2] of section 1.4.

[AS3] It is assumed that the number of users in the first year of operation will not exceed the platform's capacity, and modifications to the platform will not be needed. Later modifications are not discussed in the present document.

[AS4] It is assumed that any pre-existing taxi reservation service shall not be integrated with the platform.