Software Requirements Specification

for

TCAB

**Version 1.1 approved**

**Prepared by TCAB Crew**

**TCAB**

**Nov. 21, 2018**

**Table of Contents**

[**Introduction**](#_8scvcbacznc0) **3**

[**Purpose**](#_3znysh7) **3**

[**Document Conventions**](#_8btk91rumhtd) **3**

[**Intended Audience and Reading Suggestions**](#_7scta6ijbwcn) **3**

[**Product Scope**](#_jn3w6imb89kb) **3**

[**References**](#_1e647gbu8bsq) **4**

[**Overall Description**](#_qpivtvlq4zie) **4**

[**Product Perspective**](#_2s8eyo1) **4**

[**Product Functions**](#_1hlh3rpuzfdt) **4**

[**User Classes and Characteristics**](#_r5umq0urs9ee) **6**

[**Operating Environment**](#_xfd3qvuboap6) **7**

[**Design and Implementation Constraints**](#_p3zvd4i8fh8p) **7**

[**User Documentation**](#_fii2xpwu17oe) **7**

[**Assumptions and Dependencies**](#_k74cmlhjpt6r) **8**

[**External Interface Requirements**](#_43g9uc8vow2g) **8**

[***User Interfaces***](#_2jxsxqh) ***8***

[***Hardware Interfaces***](#_ngmby7kctyf4) ***8***

[**Software Interfaces**](#_z3o49hopw5oo) **9**

[**Communications Interfaces**](#_9yx062wrbmdi) **9**

[**Other Nonfunctional Requirements**](#_3whwml4) **9**

[**Performance Requirements**](#_2bn6wsx) **9**

[**Security Requirements**](#_rdtsl3zdf8wj) **10**

[**Business Rules**](#_cb06hpc2y9am) **10**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Initial Revision | Nov 1, 2018 | NA | 1.0 |
| Second Revision | Nov 22, 2018 | Improved structure.  Document conventions.  More requirements. | 1.1 |

# Introduction

## Purpose

The purpose of this product is to build a platform where all the targeted riders have access to the list of available and nearby TCABs, book a fare in a minimalistic way. The scope, in this case, is the mobile app and the API design where all operations are being performed. TCABs are found for the nearest (parameter of 2km) rider available through an algorithm. Previous works of this magnitude include Uber, Grab, Lyft, Careem. The app aims at providing an online taxi service that is similar to the above platforms in technical basis while the differences in the way TCAB drivers and riders are connected varies due to the cultural, security, safety, pricing differences to the above platforms.

## Document Conventions

*There are no major conventions of format followed, but the clarification of these terms below:*

TCAB: taxi car

Driver: TCAB driver

Rider: App user

Fare: a typical TCAB fare with attributes.

App: The Taxi-finding application

## Intended Audience and Reading Suggestions

This paper is intended to give a general idea of everything that is connected. The reader is expected to have a minimum understanding on how REST APIs work. The user must also identify terms such as *Database, API, User, Oauth, Authentication, Authorization, CRUD* operations. Hence, mostly this report is aiming at technicalities and laying out the framework for the operation of the app. Some parts of the report are directed towards marketers, testers, and project managers.

## Product Scope

The vast number of taxis in the KRG roaming in entropy creates a market niche where an organizing mechanism will maximize profit for individual TCAB drivers and bettering the service in terms of security and swiftness for riders. The aim of TCAB is to offer TCAB drivers and their riders a medium of engaging. TCAB is a mere middleman platform. TCAB will also organize the taxi body in the city of operation, giving the rider a sense of security when booking a TCAB. While the huge number of taxis in a region like KRI is disproportionate to the population. Thus making taxi driving a minimal income job due to its reduced demand in the market. For the TCAB (as part of the public transportation system) to survive and progress into more organized and standardized service, it is essential that private companies like TCAB reorganize the market and provide a standard taxi service that in quality will eventually reduce the number of taxis, provides better service for cheaper with more accountability and raise the TCAB drivers’ monthly income by attracting more riders to the company’s platform.

## References

*Grab Business Model* [*https://time-to-first-byte.ghost.io/grabs-business-model-2012-2018/*](https://time-to-first-byte.ghost.io/grabs-business-model-2012-2018/)

*Uber Business Model* [*https://jungleworks.com/uber-business-model-revenue-insights/*](https://jungleworks.com/uber-business-model-revenue-insights/)

*Careem Business Model* [*https://rctom.hbs.org/submission/careem-why-is-a-local-startup-winning-the-competition-with-a-global-giant-like-uber/*](https://rctom.hbs.org/submission/careem-why-is-a-local-startup-winning-the-competition-with-a-global-giant-like-uber/)

API Standards <https://cloud.google.com/apis/design/>

CRUD <https://docs.microsoft.com/en-us/iis-administration/api/crud>

# Overall Description

## Product Perspective

The TCAB application intended herein is built from scratch and is in the way a part of a larger system. The application is in its entirety is a new technology introduced to Kurdistan, since there’s no precedence for such a feat. However, the idea of the app is similar to many other existing online taxi services around the world such as Uber, Grab, Careem.

## Product Features and Functions

This project includes two types of applications for two different users: TCAB drivers (B2B), and TCAB riders (B2C)

**Features**

* Real-time GPS tracking algorithm with LQE & Gyro APIs
* Real-time ETA calculation algorithm for price prediction
* Clear and intuitive UX with organized-feel transitions, and Animations
* Clean UI
* All products will be available for iOS and Android
* All products will be available in Kurdish (Sorani, Kurmanji-Latini), Arabic, and English
* Animated splash screen
* App slide menu - user-friendly way to access all features of the rider’s app.

**Functions**

1. Account verification - riders will need to verify their mobile number via OTP received via SMS
2. Login and Register
3. The user will have the option to log-in via email and phone.
4. Book a Trip - rider will enter the location, drop off location using Google Places API, or manually drag and pin them on the app.
5. Rider will also use real-time ETA for the nearest driver available for each type of service
6. Rides:
   1. has an option to press Ride now
   2. Schedule a fare for pick up later by selective future date and time.
7. E-Wallet, *this is not required for the beta phase, but must be implemented within the first year of launch*
   1. balance will be shown on the top along with his/her avatar image and an option to log out.
   2. invoicing and payment gateway integration of Paypal, debit/credit card or up to 3 mobile money providers.
8. Booking a Fare:
   1. The user can cancel the fare in X minutes of the window.
   2. Cancelation is free before the driver arrives at the pickup location. Otherwise, the user has to be penalized a fee.
   3. The user account shall be disabled until the outstanding balance is cleared.
   4. Review and rating - users can provide feedback, all feedbacks are listed
9. Driver Information:
   1. Name
   2. Picture
   3. Vehicle plate, name, and model
   4. Rider will see driver’s details on screen and see he is coming to pick him up and can also track driver on the map and can also call or SMS the driver.
10. Booking history:
    1. Sorted by most recent
    2. Metadata about fare:
       1. Date and time
       2. Driver information
       3. Pick up and destinations location.

**The TCAB rider App (B2C)**

1. The app should include a login page which also includes a sign-up option.
   1. The login should only be made through a phone number and email.
   2. There is an option to use the app using a one-time password, provided via SMS.
2. The home page of the app shows a map relative to the rider’s geolocation. The geolocation is derived from rider’s mobile device - GPS module.
3. The map will show all TCAB drivers’ heartbeat within a parameter of 2 KM.
4. A header fixed on the map will trigger the booking process:
   1. Pick up Location:
      1. Pin your location: mark their location on the map, using their mobile devices’ GPS.
   2. Destination:
      1. Explore: A list of nearby places to explore (cafes, restaurants, hotels, malls, etc…)
      2. Pin your destination: mark their location on the map, using their mobile devices’ GPS.
      3. Type: explicitly type the destination. The keyword is used to determine similar destinations.
   3. Ride option
      1. Private taxi
      2. Family taxi
      3. Carpooling
      4. Delegate: to book fares outside of normal working hours 5:00 AM-12:00 AM.
5. A request to the API is made with details of the destination. Then a notification will be sent to all the TCABs in the parameter of 2Km and one driver confirms the request. The API returns fare details to the user.
6. The user is shown the estimated fare details
   1. Exact fare price
   2. Shortest route possible polygon is shown on the map
   3. Estimated fare time
   4. Driver information\*
   5. A button at the bottom will let the user confirm the fare if they chose to take it.
7. When a fare is confirmed and the TCAB reaches the designated place, a cancellation of the fare will cost the rider an amount of money (preferably between 1500-2000 IQD), and will immediately hold the rider's account to prevent the abusive behavior from customers. A second chance will be given to an account only when the riders pay off his debt.
8. It should be clearly mentioned through a big caveat that any such behaviors mentioned above will duly be met with suspending the account until the due debts will be paid off.
9. The main method of payment is through cash.
10. The rider will be prompted to review the driver.

**The TCAB Driver App (B2B)**

The following list is the differences and additions to the TCAB rider App (B2C):

1. The map will not show other TCAB drivers to any individual user of the app.
2. Fare popup:
   1. Location (name) and destination (name)
   2. Two buttons (large)
      1. Accept (green)
      2. Decline (red)
3. A declined offer will be dismissed from the view for 2 minutes. This is to account for the scenario where their driver might miss click the offer. Also, there is a chance that the pinged TCAB driver is the only one around in that parameter, so the rider must get a taxi, hence the rider can ping the TCAB driver after 2 minutes.
4. When the driver arrives at the rider's location, the driver’s app should send a push notification to the rider saying that the ride is ready. The notification should be sent when the driver is in 100 meters range.
   1. Direction API, to determine which route the driver should take to figure out the quickest one.
5. The app should provide directions and route to the pick-up location of the rider and then the direction and route from the rider's location to the destination set by the customer.
6. Once the driver arrives at the fare’s destination - within a parameter of 20 meters, he can mark the fare as complete.
7. After leaving feedback view, the driver can once again broadcast their vehicle heartbeat on the pool of available drivers.
8. For a driver to qualify for being available for booking, the mobile app must send a heartbeat to the network periodically (5 seconds). If the app stops sending a heartbeat for a period longer than 15 minutes the driver is considered not available for riders and is out of work time.
9. Drivers’ account shall be disabled when the former fails to pay their commission share. There is a grace period.

## User Classes and Characteristics

1. Supervisors: Tweaks, maintains, and tests the app.
2. Analyst: Have permission to check historical records for record keeping, monitoring, and finances. These data are used toward generating including but not limited to daily, weekly, monthly, quarterly, and annual reports.
3. Drivers: Use the app to detect riders and fares in their respective area. They use it to find locations of riders, obtain directions from it, and use it as a communication channel with the operators.
4. Riders: Use the app to locate TCABs for their need of a ride to a designated destination. They use it to give about feedback drivers.
5. Operators: People who are working, in shifts, on the dispatching panel to oversee and monitor the drivers’ activities, and provide help to them in cases of functional/technical failure.

## Operating Environment

The app hardware requirement must have a GPS module for determining longitude and latitude of any instance of the app. That is a user operating the app.

The app software environment must be either iOS or Android.

Minimum Software versions required:

iOS 8.0 ([source](https://www.statista.com/statistics/565270/apple-devices-ios-version-share-worldwide/))

Android 4.4 Kitkat ([source](https://www.statista.com/statistics/271774/share-of-android-platforms-on-mobile-devices-with-android-os/))

## Design and Implementation Constraints

Communications protocols: HTTP 1.1 and HTTP/2

Languages communications for the app interface: Kurdish, Arabic, and English.

## User Documentation

When we have the app ready and is in development stage we will make tutorial videos explaining the instructions in the required languages (English, Kurdish, and Arabic) for riders and TCAB drivers in how to use the app properly. There will also be a user manual for both B2C and B2B interfaces when the app is first downloaded on devices.

## Assumptions and Dependencies

One major dependency is Google Maps and Google Maps API for almost anything related to showing locations, finding places and determining routes. If for any reason Google servers are down, the app is essentially useless.

Fast and scalable servers to host and deploy the API on AWS E2.

One consideration is a backup map provider: MapBox with similar API to be deployed in such technical failure.

# External Interface Requirements

## Hardware Interfaces

TCAB is intended for mobile use. Hence, the final product shall be a mobile app targeting Android and iOS users. There are two main stacks: front-end and back-end. Front-end refers to the mobile app. Back-end refers to the REST API accessed through HTTPS protocols. All operations follow CRUD design. Data formats transmitted are in JSON.

## Software Interfaces

Linux

Nodejs runtime.

MongoDB.

HTTP

Google API.

Phone SMS API

## Communications Interfaces

API is deployed on an HTTP server. Hence, the only protocol to access the API is HTTPS.

Endpoints require an authentication. That is either a client ID or an RS256 signed JWT.

Load Balancing through distributed database architecture for easy scalability and to reduce infrastructure costs

# Other Nonfunctional Requirements

## Performance Requirements

The Pool API must be fast in determining which vehicles) are closer to the user.

## Security Requirements

The API is stateless. Authentication is processed by several mediums. First, a password-less login using SMS authentication. The user provides a valid phone number and receives a one-time code in lieu of a password.

**Appendix B: Analysis Models**

A statistical analysis must be implemented through Google Analytics. Respective actions, dimensions, and views must be created to reflect desired reports.