



...from thoughts to finish

ISO 9001 and 27001 Accredited

---

# Functional Specification for Taxi App Project

---

DEVELOPED BY ZAFAR AYUB

## Table of Contents

1	Overview .....	3
2	Backend Development .....	3
2.1	Core DB .....	3
2.2	Admin Panel .....	4
2.3	Reports .....	4
3	API's Development .....	5
3.1	Default API's .....	5
3.2	Develop API's .....	5
4	Use Case Descriptions .....	6
4.1	Diagram .....	6
4.2	Authentication .....	7
4.3	Profile Access .....	8
4.4	Commission Module .....	9
4.5	Google Searching .....	10
4.6	Picking up .....	11
4.7	Travelling .....	11
4.8	E-Commerce .....	12
4.9	System Updates .....	12

## 1 Overview

Taxi app functional specification document will describe linear activity between all stake holders (customer, driver, and taxi app company) furthermore identify main development and configuration domain separately.

All these segments will continue update in whole development cycle separately where common system flow is already design. This document will contain following values gradually like;

- Use case descriptions
- Class activity descriptions (will continue)
- Sequence activity descriptions (will continue)
- State chart descriptions (Optional)
- Activity descriptions (will continue)
- Component descriptions (Optional)

### Note:

FS document will continually update as per necessity and discussion

## 2 Backend Development

Backend of Taxi app is core of whole system and its also controlling unit of portal and both apps (customer / driver); here bifurcation of backend is:

### 2.1 Core DB

Database will hold all tables' information where these tables belonged with system as well users (customer / driver both);

AA-001. Creation of main DB

AA-002. Add table's (user, group, login information, map, tracking history, financial history, remakes, commission, bank detail, etc.)

AA-003. Specify primary key and indexing

AA-004. Make relation between all of them

### Core DB (Process Code: AA)

Code	Descriptions	DBA	QA
AA-001		Date	Date
AA-002			
AA-003			
AA-004			

## 2.2 Admin Panel

Back office control system where whole DB as well portal & both apps will manage (mean's enable / disable) in here; admin panel consist following modules:

- AB-001. User / group manager (create / edit / disable / delete)
- AB-002. Add user / group rights functionality
- AB-003. Add (driver) group member's list functionality
- AB-004. Add commission functionality
- AB-005. Add bank detail functionality
- AB-006. Add reports functionality

### Admin panel (Process Code: AB)

Code	Descriptions	Developer	QA
AB-001			
AB-002			
AB-003			
AB-004			
AB-005			
AB-006			

## 2.3 Reports

Develop systems and user (customer / driver) reports with respect to requirements, here add some common reports:

- AC-001. Add user / group detail information reports
- AC-002. User login report (successful / unsuccessful)
- AC-003. Customer / driver traveling history reports
- AC-004. Total no. of invoicing reports with respect to customer / driver
- AC-005. Commissioning reports
- AC-006. Driver team members report
- AC-007. Bank transaction reports (customer / driver / Taxi app system)

### Reports (Process Code: AC)

Code	Descriptions	Developer	QA
AC-001			
AC-002			
AC-003			
AC-004			
AC-005			
AC-006			
AC-007			

### 3 API's Development

There are two type of API's one from SDK (Google and iOS) and others will develop by developer:

#### 3.1 Default API's

Default API's are come from Google and iOS SDK with following characteristics, where it's will customize as per Taxi app system:

- AD-001. Versioning API's
- AD-002. Push notification API's (for alert)
- AD-003. Google map API's
- AD-004. Login session API's
- AD-005. Remakes & reviews API's
- AD-006. Certificate API's

##### Default API's (Process Code: AD)

Code	Descriptions	Developer	QA
AC-001			
AC-002			
AC-003			
AC-004			
AC-005			
AC-006			

#### 3.2 Develop API's

These API's will develop from edge to finish with associating respective tables:

- AE-001. Authentication API's
- AE-002. Admin panel API's
- AE-003. Bank transaction API's
- AE-004. User profile API's (customer / driver)
- AE-005. Commission module API's
- AE-006. Google route tracking API's
- AE-007. Fare calculating API's

##### Develop API's (Process Code: AE)

Code	Descriptions	Developer	QA
AE-001			
AE-002			
AE-003			
AE-004			
AE-005			
AE-006			

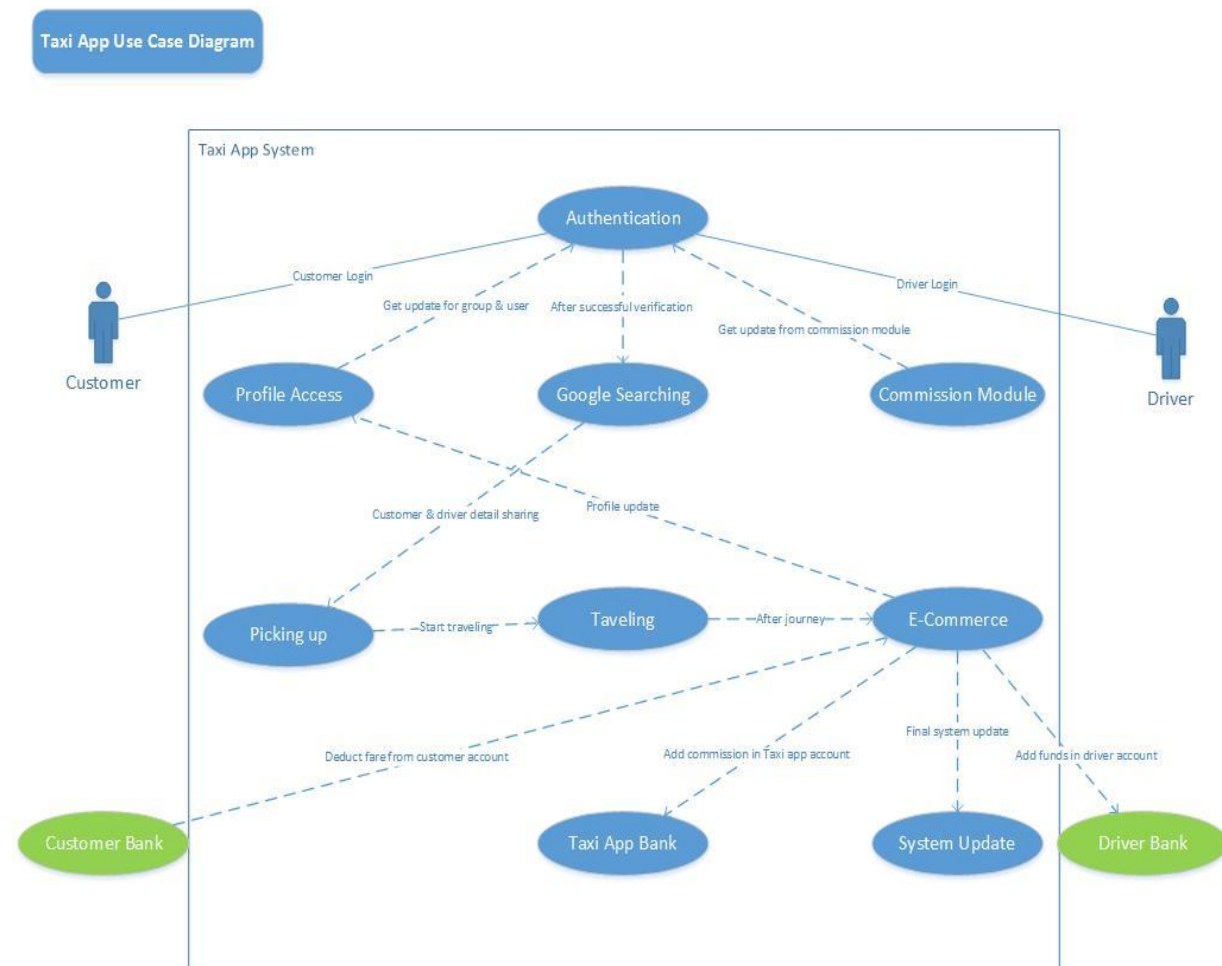
## 4 Use Case Descriptions

In Taxi App system use case illustrates a unit of functionality provided by the system (interflow of system) by this development team will;

- Visualize the primary functional requirements of system
- Identify relationship between all stakeholders (customer, driver and taxi app system)
- As well identify relationship among different use cases

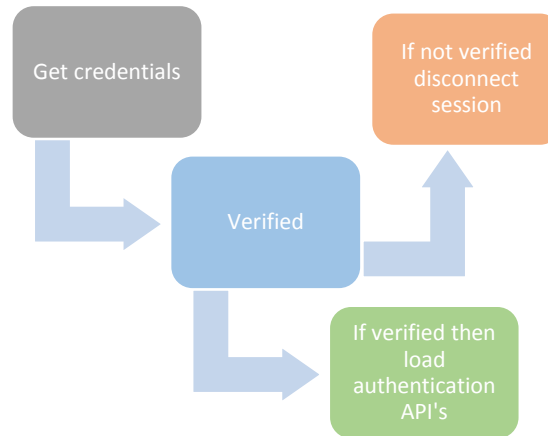
### 4.1 Diagram

Understanding use case diagram is describe in here:



## 4.2 Authentication

Authentication segment contain users (customer, driver, taxi app system and admin) sing-in and login processes where some of login credentials are associate after this user will authorize for system login / access as per his privilege.



- AF-001. Accounts type (customer / driver / taxi app system / admin)
- AF-002. Account status (enable / disable)
- AF-003. Account expiry (associate with billing table)
- AF-004. Account region (will maintain region wise login)
- AF-005. Account schedule (login time will mention)
- AF-006. Associate all values with authentication table
- AF-007. Associate authentication API's and associate with table
- AF-008. Get update from profile module
- AF-009. Get update from commission module

### Authentication Process (Process Code: AF)

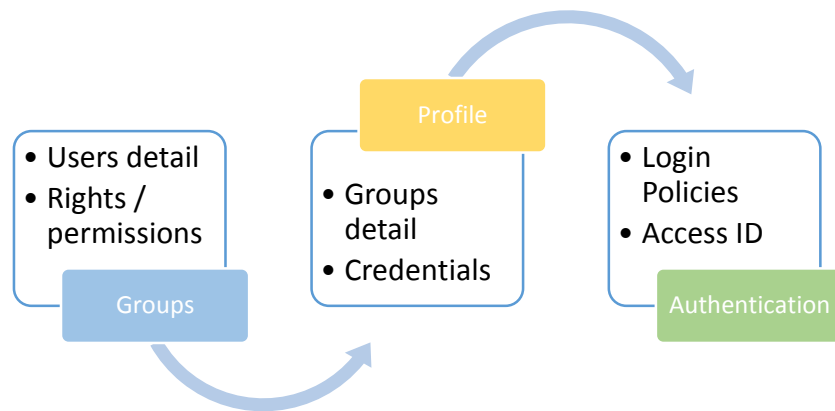
Code	Descriptions	Developer	QA
AF-001			
AF-002			
AF-003			
AF-004			
AF-005			
AF-006			
AF-007			
AF-008			
AF-009			

### 4.3 Profile Access

System profile will contain three type groups:

- Customer (for customer and friends)
- Driver (where he'll add his reference drivers and reference customer)
- System (contain system user for standard work, report user and admin user)

All of three type of users acquire basic rights from their specific groups which will embed in authentication process:



- AG-001. Add groups type (customer / driver / system)
- AG-002. Assign group attribute (user / system)
- AG-003. Add user (account) in group functionality
- AG-004. Add remakes functionality at user (account)
- AG-005. Add user (account) mode functionality (online / offline / busy / etc.)
- AG-006. Associate user to group as member functionality
- AG-007. Associate all functionality with their respective user / group tables

#### Profile Access Process (Process Code: AG)

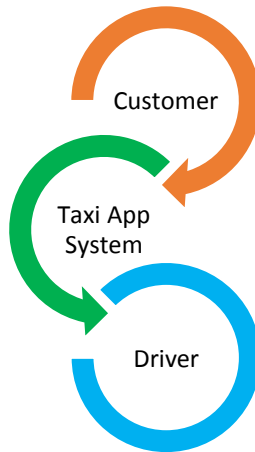
Code	Descriptions	Developer	QA
AG-001			
AG-002			
AG-003			
AG-004			
AG-005			
AG-006			
AG-007			



#### 4.4 Commission Module

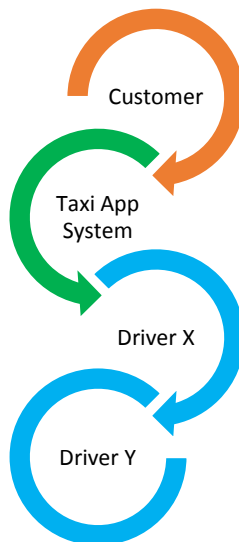
Commissioning module will extract values form commissioning table which may vary (input from admin panel), commissioning module is describe as:

##### Direct Commission



- Customer may search driver availability by Taxi App system
- Driver will track and get reach the picking location
- After completion of journey when customer pay his fare automatically some % commission will deduct and transfer Taxi App system account and remaining will forward to driver account

##### In Direct Commission



- Customer may search driver availability by Taxi App system
- Driver X identify that he isn't able to picking but forward his request to his team member Driver Y
- Driver Y getting picking customer and make his journey
- After completion of journey when customer pay his fare automatically some % commission will deduct for Taxi App system as well some % of commission for Driver X
- And remaining amount will forward to Driver Y account

- AH-001. Develop formula for direct commission module  
 AH-002. Develop formula for in direct commission module  
 AH-003. Associate these values with commission table  
 AH-004. Associate commission API's with system  
 AH-005. Update function for user (customer / driver) authentication

#### Commission Module (Process Code: AH)

Code	Descriptions	Developer	QA
AH-001			
AH-002			
AH-003			
AH-004			
AH-005			

## 4.5 Google Searching

In following functionally user (customer) will get search for nearest driver and driver will get notification from system (with customer picking position); driver accept notification then customer will get driver credentials and possible picking up time.

- AI-001. Add Google map coordinates in back end  
 AI-002. Associate these coordinate with related table "maps"  
 AI-003. Get develop Google map calling API's from table  
 AI-004. Develop map searching function for users (customer / driver)  
 AI-005. Develop route calculation formulae (with respect to distance / with respect time)  
 AI-006. Associate these API's with user / group table  
 AI-007. Associate these API's with "history" table  
 AI-008. Attached driver alert API's (notification)

#### Google Searching (Process Code: AI)

Code	Descriptions	Developer	QA
AI-001			
AI-002			
AI-003			
AI-004			
AI-005			
AI-006			

AI-007			
AI-008			

#### 4.6 Picking up

Driver get picking coordinate (from Google map) and get reached position, where automatically generate notification from system to customer that driver at reached. Customer will enter his / her destination where system will forward this to driver second calculate fare (possible) as per formula AD-005.

- AJ-001. Develop customer location tracking function (for driver) to forward for picking him / her
- AJ-002. Add destination module (associate with related table)
- AJ-003. Associate destination module to with AD-005 for calculating fare (possible) and reaching time
- AJ-004. Generate notification for driver (to destination value)
- AJ-005. Forward possible fare to customer / driver both

##### Picking up (Process Code: AJ)

Code	Descriptions	Developer	QA
AJ-001			
AJ-002			
AJ-003			
AJ-004			
AJ-005			

#### 4.7 Travelling

Driver will make request to customer for initiate the journey request (until system will not add start values), where customer aspect this request then system automatically update to driver and fare will start. At destination point driver will again make request to customer to aspect finish journey (until system assume journey will not finish) and then customer make confirm it close.

- AK-001. Develop API's for start journey and close journey notifications
- AK-002. At start journey update Google map table "map" with customer and driver profile update (table update)
- AK-003. At close journey update customer and driver profile as well
- AK-004. At finish journey update Google map table "map" for record tracking
- AK-005. Update all values to related tables

##### Traveling (Process Code: AK)

Code	Descriptions	Developer	QA
AK-001			
AK-002			
AK-003			
AK-004			
AK-005			

## 4.8 E-Commerce

After getting confirmation from customer (Taxi App) system will make request to customer credit card (customer bank) for deducting fare to automatically forward to driver bank account (system billing to driver is weekly bases) where Taxi app system commission also be deduct. In whole calculation values of commission comes from commission module. Furthermore system generate two separate invoices one for customer (with route and fare detail) and second for driver (with route and transfer funds detail).

- AL-001. Develop funds deduction method customer bank account (via credit card)
- AL-002. Develop funds transfer method to driver bank account as well add commission in Taxi app system bank account
- AL-003. Make notification / invoices for both customer and driver
- AL-004. Update customer and driver table for history

### E-Commerce (Process Code: AL)

Code	Descriptions	Developer	QA
AL-001			
AL-002			
AL-003			
AL-004			

## 4.9 System Updates

In closing updates all open tables with respective values and make busy off for driver, as well release customer and driver (both) sessions from system.

- AM-001. Update all tables with respect to customer and driver
- AM-002. Change driver status for available (in 10 to 15 mins)
- AM-003. Release customer and driver session from Taxi App system

### System Updates (Process Code: AM)

Code	Descriptions	Developer	QA
AM-001			
AM-002			
AM-003			