

Design and Development of a Mobile Application for the Management of Means of Transport (Smart Taxi)

Jihen HEDHLI

Computer Technology Department/ Higher Institute of Technological Studies, ISET of Nabeul -University
Campus 8000 Nabeul- Tunisia

Abstract:

This article describes the design and development of a mobile application for the management of means of transport, more specifically Taxis (Smart Taxi). The system is proposed for 2 actors represented by the owners of the Taxis and the drivers.

It was modeled by an agile method namely the agile Scrum framework. The mobile application was developed using the FireBase database for real-time data viewing and the Flutter framework.

This work was carried out within a cooperation between ISET of Nabeul and the company CubeIT.

Key Word: Smart Taxi, Mobile application, secured access, FireBase, car location

Date of Submission: 16-10-2021

Date of Acceptance: 31-10-2021

I. Introduction

The technological revolution has affected many domains in our life with the aim of making daily tasks easier to do and faster. These domains include private transport or what we call transport vehicles with driver (TVD), we were particularly interested in the taxis sector.

The taxi has become in recent years the most coveted form of transport in any city in the world. This service has also undergone a significant change thanks to automotive technology and communication [1].

But that does not prevent this sector from experiencing problems. Indeed, taxi owners encounter the problem of monitoring their financial revenues. Between the various expenses and the revenue, a problem of trust arises with the drivers employed.

Following an investigation carried out with several taxi drivers in the region, we noticed that they used traditional methods for recording and monitoring their charges linked to the costs of maintaining their cars, and the daily charges, which does not allow them to make a precise follow-up.

II. Research objective

The proposed solution consists of developing a mobile application for two users. The application offers the first user who is the taxi owner, these services:

1. Driver management,
2. Car management,
3. The management of car expenses (taxes, maintenances)
4. Statistics monitoring,
5. Real-time consultation of the vehicle's position.

The application is also intended for Taxi driver, who represent the second user and will be embedded in the car, this application offers:

1. Route management
2. The cash register management
3. The management of daily expenses
4. At the end of the day, display the financial part of the driver.

III. System development

The system described in this article, consists of designing and developing a mobile application for managing and monitoring taxis. This system represents a cross-platform that was produced using the following tools:

Android studio: Android Studio is the official Integrated Development Environment (IDE) for Google's Android operating system, built on JetBrains IntelliJ IDEA software and uses the Gradle production engine. It is designed specifically for Android development. It was used as an IDE in our application [2].

Firestore: Firestore is Google's mobile platform that facilitates the creation of both scalable and efficient back-ends. This platform offers a set of hosting services for any type of application (Android, iOS, JavaScript, Node.js, etc.).

It offers real-time NoSQL hosting of databases, content, social authentication (Google, Facebook, Twitter and GitHub), and notifications.

This platform was used in our system to allow the 2 actors (taxi owner and taxi driver) to manage data in real time [3].

Flutter: Flutter is an open-source mobile application development framework created by Google based on the Dart language. It is used to develop apps for Android and iOS and is also the main method of creating apps for Google Fuchsia.

Flutter takes the same level as React, Native or Xamarin, which means that from code written in Dart will then be generated Android and iOS code. Flutter is used to develop our mobile application.

Dart: Dart is an open-source programming language developed by Google. It is used for the development of mobile, Web, desktop and other applications incorporating Internet of Things (IoT) technology. Dart can also be used for server-side programming, thanks to it, Flutter has become the fastest cross-platform framework that offers native performance for Android and iOS [4]. We used Dart in our application.

IV. Design model

To design our taxi management and tracking mobile application, we have chosen an agile design model, widely used in software development, which is based on an iterative and incremental development cycle. It is carried out in a collaborative spirit and generates a product considering the changing needs of customers. Among the existing agile frameworks, we have chosen the Scrum framework.

The Framework Scrum Agile

Scrum is based on "sprints" which are short space-times that can range from a few hours to a month, and which have a goal to achieve and always result in the delivery of a potentially usable prototype.

This framework is based on a Scrum team made up of:

- Product Owner: who generally represents the customer, who defines the needs of the project, and prioritizes tasks.
- Scrum Master: who represents the team leader who will ensure that the team lives agile values and principles and follows the processes and practices they have agreed to use. He tries to clear obstacles, manage team dynamics, and ensure a good relationship between the team and the Product Owner as well as other people outside the team [5].
- Development team: a team made up of three to nine people who collaborate daily to achieve the expected results.

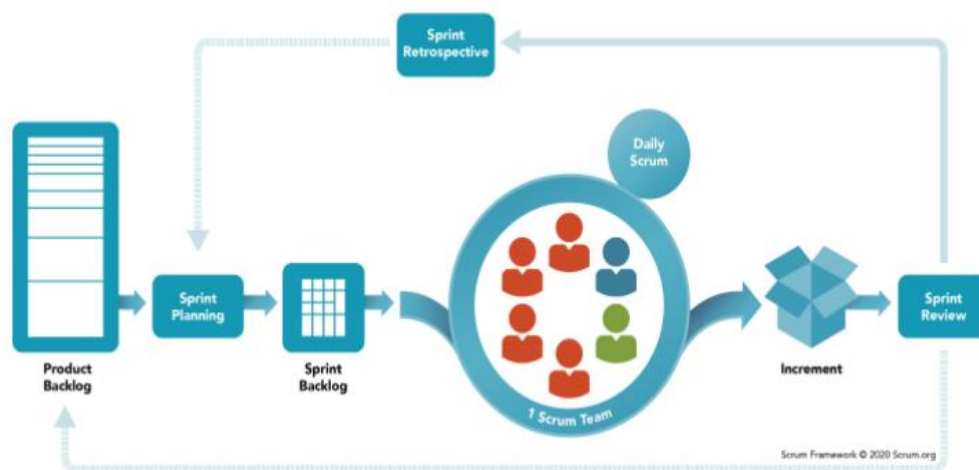


Figure1: The Scrum framework (<https://www.scrum.org/resources/what-is-scrum>)

The Process Design Diagram

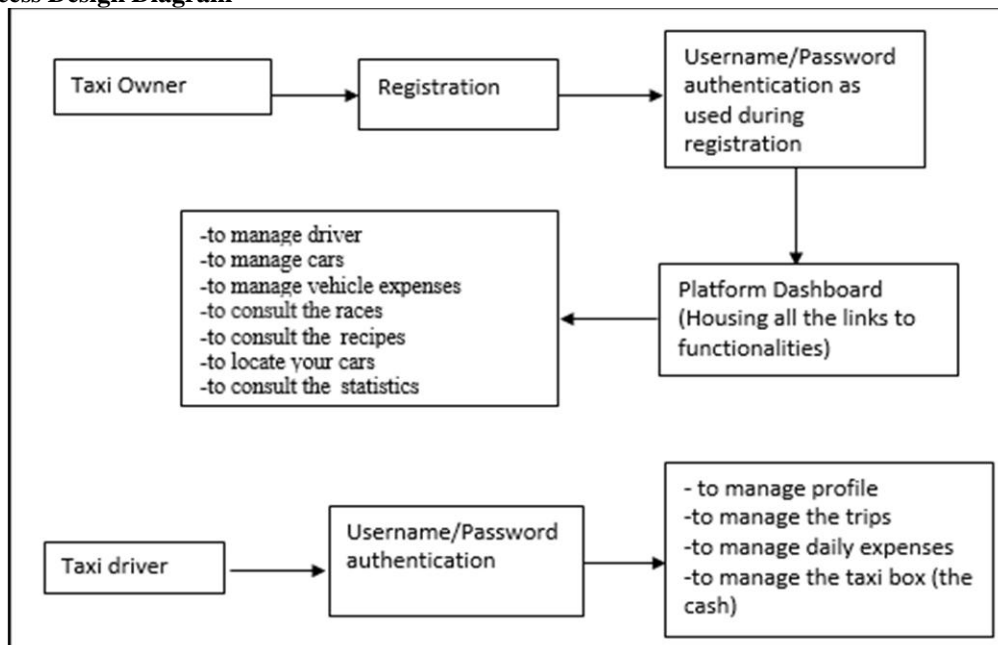


Figure2: The Process Design Diagram

V. The Diagram's Component explanation

Registration

Once the owner of Taxi installs the application, he registers by filling out a form specifying his login and password, then he accesses the application through an authentication interface.

Platform Dashboard

After authentication, the admin /the owner finds his dashboard allowing him to do several managements:

- Car management: the admin can add a new car, search for a car among existing ones, modify the parameters of a car or even delete a car (sold, damaged),
- Driver management: the admin can add a new driver, search for a driver among other drivers and delete a driver (a driver who has resigned),
- Management of car expenses: the owner of the taxi is supposed to manage car expenses, he finds an interface for adding, modifying, and finding expenses such as maintenance and taxes.
- Consulting drivers' expenses: the owner can consult the list of expenses recorded by the driver daily such as car washing, tires checking, fuel, etc.
- Consulting the roads: the owner can consult the roads (journeys) carried out and recorded by the system. Each road has a starting point and an ending point.
- Consulting the taxi box (the cash): The owner can consult in real time the cash recorded by the taxi meter.
- Car location: The owner can locate the position of his car in real time sur Google Map.
- Consulting statistics: The system can generate statistics by period according to the cash recorded and according to the drivers.

Taxi driver' s Dashboard

The admin creates accounts for the drivers who will use the application embedded in the taxi. A driver authenticates with the parameters given by the admin/owner, and on first access, he changes his password and can modify his profile. His dashboard allows him to:

- Record the roads (journeys) traveled,
- manage daily expenses (car washing, tires checking, fuel, etc.),
- manage the cash register which begins with a daily initialization with a sum of money and on each trip, the system will record the value of money for this road and accumulate it to the revenue.

- When the taxi driver finishes his work, he can consult his own cash directly, since the system has the value of the revenue accumulated by the counter and has the value of the daily expenses which are subtracted from the revenue. The system displays the amount earned by the driver according to the percentage on which the taxi owner and the taxi driver agreed.

VI. Deployment diagram

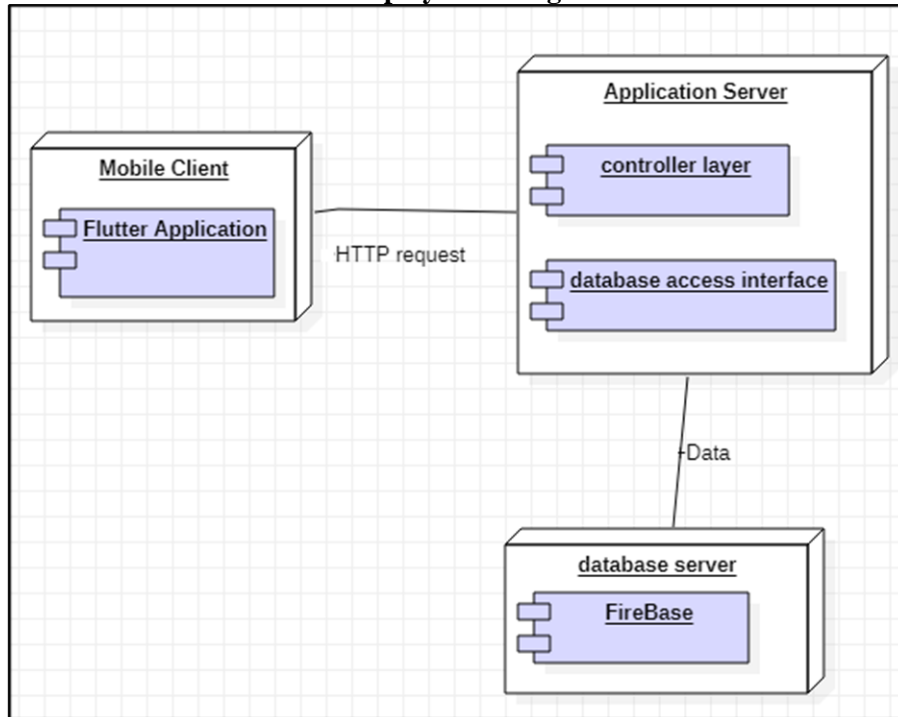


Figure3: Deployment diagram

VII. Result

The result of executing the features described in this article are provided through these interfaces.



Figure4: interface for adding a taxi driver

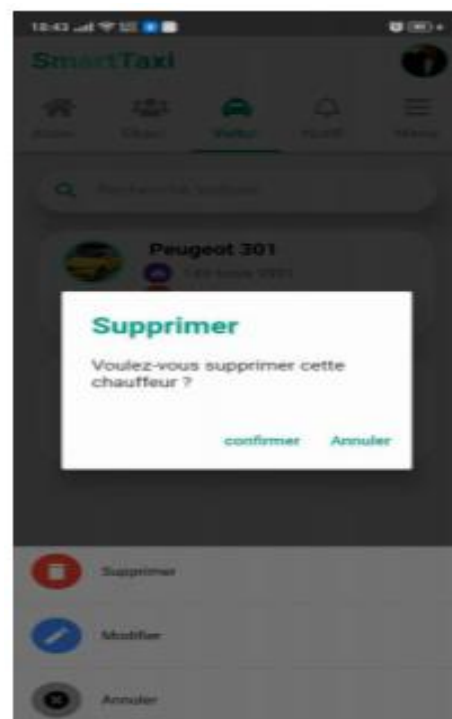


Figure5: interface for deleting a car

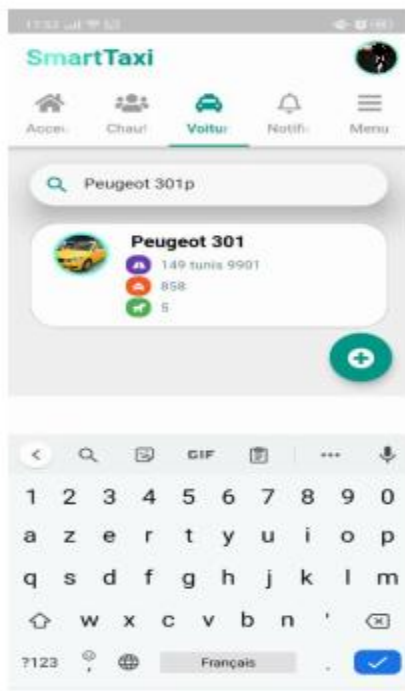


Figure6: interface for searching a car

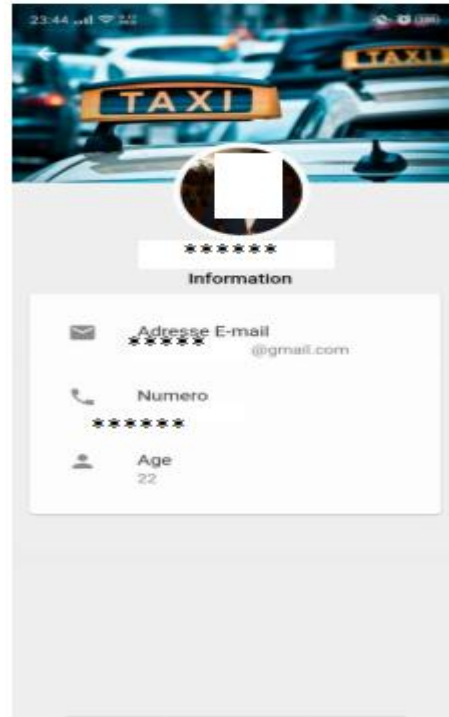


Figure7: interface for consulting the profile



Figure8: interface for adding daily expenses

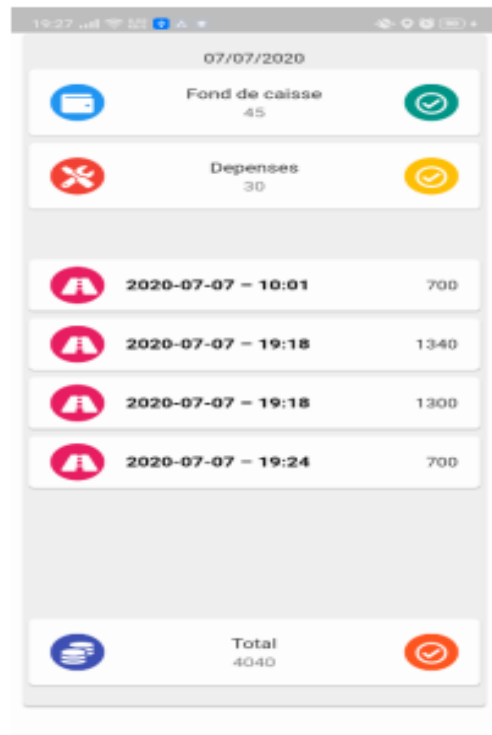


Figure9: interface for consulting the cash register



Figure9: interface for searching a destination



Figure10: interface for locating the car's position

VIII. Conclusion

Initially, this application is offered for the taxi owners and taxi drivers in order to manage journeys and the cash register and establish a regular monitoring illustrated by statistics.

A second version of this system has been dedicated to the merchandise delivery sector specifically for trucks.

A third version in progress which will be offered to the public transport sector, specifically buses, by integrating an intelligent ticket control system in it.

References

- [1]. Transports et déménagements, <https://www.transports-et-demenagement.com/le-taxi-pour-une-qualite-accrue-du-service-du-transport>
- [2]. Android Studio, https://en.wikipedia.org/wiki/Android_Studio.
- [3]. A quoi sert Firebase la plateforme mobile de Google ? ,<https://junto.fr/blog/firebase/>
- [4]. Apprendre le langage de programmation Dart, <https://www.udemy.com/fr/topic/dart-programming-language/>
- [5]. Agile Alliance, Scrum Master, <https://www.agilealliance.org/glossary/scrum-master>

Jihen HEDHLI. "Design and Development of a Mobile Application for the Management of Means of Transport (Smart Taxi)." *IOSR Journal of Computer Engineering (IOSR-JCE)*, 23(5), 2021, pp. 18-23.