

République Algérienne Démocratique et Populaire
الجمهورية الجزائرية الديمقراطية الشعبية
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
وزارة التعليم العالي و البحث العلمي



Université Hassiba
Benbouali -Chlef-
جامعة حسيبة بن بو علي
Faculté des séances
Département INFORMATIQUE



First cycle graduation thesis

Theme

Mobile application of Train Ticket Reservation

Realised by :

- Lamri Younes
- Hanifi Ikbal

Advisors:

- o Mme. Belkadi Khadidja

College year: 2018/2019

I. ACKNOWLEDGEMENTS

We thank ALLAH for giving us the courage and strength throughout our study until the development of this work. To our promoter Mme. Belkadi Khadidja for the trust she has given us proposing this project and also for her sound advice and guidance that surrounded us during this period of work. To all jury members for agreeing to review our work and to all computer science teachers for the knowledge they gave us during those years. We also try to thank the people who helped and encouraged us throughout this work.

II. DEDICATION

We dedicate this work to our parents, that even when u keep failing over and over again they will keep there trust in us and taught us to surpass our limits (PLUS ULTRA).

To all the frens that helped through stressful moment and made this joyful journey wishing success and happiness.

III. ABSTRACT

Considering Algeria I a third world country, we Algerian suffer from public transportation.

Most of public transportation in Algeria stop working at 6 pm and wont be available not to mention people living in isolated areas will have a hard time to be there in time to actually get a ticket and even if they do make it they will waste even greater time standing in slow queues so we thought if application that let you reserve your ticket while sitting at home not having to worry about it being available or rushing to just book a ticket.

IV. CONSTRAINS

- Trying to make the application clear and easy to use for all generation even elderly.
- Trying to make the application available at anytime and place.
- Finding a secure database server to insure user's data security

V. OBJECTIVES

- Choosing the right and the most useful devise to support our app
- Integrate a strategy to insure confidentiality and trust
- A simple functional design
- Secure users data

TABLE OF CONTENT

I. ACKNOWLEDGEMENTS	01
II. DEDICATION	01
III. ABSTRACT.....	01
IV. CONSTRAINTS	01
V. OBJECTIVES	02

CHAPTER I: INTRODUCTIONS TO MOBILE APPLICATIONS

I.1 INTRODUCTION	07
I.1.1 Tracking the history of mobile apps	07
I.1.2 Facts about mobile apps.....	08
I.2 MOBILE PHONES OPERATING SYSTEM.....	08
I.2.1 Android Operating System.....	09
I.2.2 Some of Android features	09
I.3 THE OBJECTIVE OF OUR PROJECT	10
I.3.1 What are the advantages and disadvantages of online booking systems?	11
I.3.2 9 good reasons to have an online booking system.....	12

CHAPTER II : CASE STUDY

II.1 INTRODUCTION.....	16
II.2. UML	16
II.2.1. Use case diagram	17
II.2.2. Sequence diagram.....	17
II.2.3. Class diagram.....	17
II.3. OUR APP USE CASE DIAGRAM	17
II.3.1. User activities.....	18

II.3.2. Admin activities.....	18
II.4. OUR APP CLASS DIAGRAM	19
II.5. OUR APP SEQUENCE DIAGRAMS.....	20
II.5.1. Sequence diagram “Register”	20
II.5.2. Sequence diagram “log in”.....	21
II.5.3. Ticket reservation sequence diagram.....	21
II.5. CONCLUSION	22

CHAPITRE III: IMPLEMENTATION OF THE APPLICATION

III.1. INTRODUCTION.....	23
III.2. ANDROID STUDIO USED TOOLS	23
III.2.1Used Dependencies.....	25
III.3 XML CODE	26
III.3.1 Different XML Files Used in Android	27
III.3.2 User interfaces.....	28
III.3.3 Icons and Images	29
III.4 APPLICATION FONCTIONALITIES.....	30
III.5 FIREBASE DATABASE	33
III.5.1 Database structures.....	34
III.5.2 FIREBASE ADMIN CONSOL.....	35
III.5.3 Firebase used functionalities	35
III.6. CONCLUSION	36
IV. GENERAL CONCLUSION	37
V. REFERENCES	38

LIST OF FIGURES

CHPTER I

Figure I.1: Mobile Applications.....	08
Figure I.2: Different Operating System.....	09
Figure I.3: features of Android Operating system	10
Figure I.4: Online Booking System advert	12

CHAPTER II

Figure II.1: UML LOGO	16
Figure II.2: Use Case Diagram(User activities).....	18
Figure II.3: Use Case Diagram(Admin activities).....	19
Figure II.4: Class Diagram.....	19
Figure II.5: Sequence Diagram (Registration).....	20
Figure II.6: Sequence Diagram (Login)	21
Figure II.7: Sequence Diagram (Booking)	22

CHAPTER III

Figure III.1: Android Studio LOGO	24
Figure III.2: The Project Files in our App.....	25
Figure III.3: The Application Dependencies.....	26
Figure III.4: Diagram of hierarchy for View and ViewGroup objects	27
Figure III.5: Layout xml files	27
Figure III.6: XML files	28
Figure III.7: Colors used in the application.....	29

Figure III.8: Some application layouts	29
Figure III.9: The logo of our application.....	30
Figure III.10: Icons used in the application.....	30
Figure III.11: Register and authentication system	31
Figure III.12: User profile security options.....	31
Figure III.13: Application different functionalities	32
Figure III.14: Application Settings	33
Figure III.15: Firebase Features.....	34
Figure III.16: Firebase data structures	35
Figure III.17: Authentication used providers	36
Figure III.18: Realtime database rules.....	36

a. GENERAL ORGANISATION OF MEMORY

The rest of document is organized as bellow: First chapter: Introduction to mobile applications Second chapter: Case Study
Third chapter: Designing a Train Ticket Reservation System

Chapter I :INTRODUCTION

I.1.1 tracking the history of mobile apps:

Today our lives move around mobile applications. Whether we need to wake up in the morning or do a quick meditation or navigate the route to client's office or book a dinner table, Apps have made everything just a click away.

But let's rewind a few years (precisely over two decades) and see the journey of mobile applications. How apps became such a vital part of our lives and how they went through a dramatic change and development. On the 3rd day of April 1973, the first mobile phone call was made by Martin Cooper of Motorola to Dr. Joel S. Engel of Bell Labs. That device or instrument weighed 1.1 kg and measuring (23 x 13 x 4.45) cm. And, it took two decades of Research and development to get first mobile application for smartphones, and the credit goes to IBM Simon, who introduced the world with the first mobile apps for smart phones. If you go back to the history of the mobile applications, then you can clearly figure out that a few Java games, a calculator or monthly calendar were all that came under the category of mobile apps. However, the first smart phone was announced for the general use by IBM in 1993 that was equipped with the features like calculator, world clock, calendar and contact book. The BlackBerry Smartphone released in 2002 was the next major achievement in the field of mobile application development and it was marked by BlackBerry Limited, formerly known as Research In Motion Limited (RIM) and integrated with the innovative concept of wireless email. Some facts about early mobile phones:

- You could only make calls for 30 minutes a day due to the terrible battery life and users had to charge their mobile phone for more than 10 hours.
- Mobile users could only call a few people a day in your area because of network range issues.

Since then, technology has moved at an astonishing rate including features on phones we would never have thought possible like controlling the temperature of house or monitoring your heart rate and blood pressure, all through mobile apps! But the real birth of mobile phones was by hands of neither the less steve jobs and Apple inc. The iphone changed

everything and became an instant hit all across the world and had its own vision and new way to access our phones, through touch screen . Most people have smart phones and iPhones today, so apps are easy to access and simply make your life better as a result. There are millions of mobile apps at present. For examples, apps for social networks, travel, health, banking, fitness, calendars, games, news and much more. The Apple App store adds countless apps every month.[1]

I.1.2 facts about mobile apps:

- Tablet sales are expected to match computer sales within the next two years.
- Mobile apps are used more than websites.
- Mobile apps can be developed on multiple platforms.



Figure I.1: Mobile Applications

I.2 Mobile phones Operating Systems (OS):

A mobile operating system (or mobile OS) is an operating system for phones, tablets, smartwatches, or other mobile devices. While computers such as typical laptops are 'mobile', the operating systems usually used on them are not considered mobile ones, as they were originally designed for desktop computers that historically did not have or need specific mobile features. This distinction is becoming blurred in some newer operating systems that are hybrids made for both uses.

Mobile operating systems combine features of a personal computer operating system with other features useful for mobile or handheld use; usually including, and most of the following considered essential in modern mobile systems; a touch screen, cellular, Bluetooth, Wi-Fi Protected Access, Wi-Fi, Global Positioning System (GPS) mobile navigation.[2]

The most used operating system is googles Android



Figure I.2: Different Operating System

I.2.1 Android Operating System:

The Android operating system is a mobile operating system developed by Google primarily for touchscreen devices, cell phones, and tablets. Its design allows users to manipulate mobile devices intuitively, with phone interactions that mirror common motions, such as pinching, swiping, and tapping. In addition to mobile devices, Google employs Android software in televisions, cars, and wristwatches, each fitted with unique user interfaces.

Software developers who want to create applications for the Android OS can download the Android Software Development Kit (SDK) for a specific version. The SDK includes a debugger, libraries, an emulator, some documentation, sample code and tutorials. For faster development, interested parties can use graphical integrated development environments (IDEs), so we use android studio which include both SDK and IDE.[3]

I.2.2 Some of Android features:

- Integrated browser, based on the open source WebKit engine
- Optimized 2D and 3D graphics, multimedia and GSM connectivity
- Bluetooth

- 3G
- WiFi
- Camera
- GPS
- Compass
- Accelerometer



Figure I.3: features of Android Operating system

I.3.1.The objective of our project:

1. In train ticketing system there has been a collection of trains, agent who are booking tickets for customer's journey which give train ticket number and departure time of the train. According to its name it manages the details of all agent, tickets, rental details, and timing details and so on. The main objective of this project is to provide the better work efficiency, security, accuracy, reliability, feasibility.
2. All the manual work should be converted into computerized so that the load of employees should decrease. The database should be stored in computer rather than in register/manually. The train ticketing software is an easy-to-use self-service system which enables the customer buys train ticket online. After process buys train ticket is successfully, the customer can get the train ticket by print out the train ticket.

3. -Searching of data is easy.
 - Passengers don't have to wait for a long time.
 - Information is accurate. It is fast process.
 - Data efficiency is more.
4. Train Ticket Reservation System is company online system, which enable customer to check availability Train ticket. It makes the customer easy to get train ticket instead of queue up to buy the train ticket. Train ticketing reservation system is commonly used in other's company. It can use this idea to develop another system to easy their customer to buy train ticket. This concept electronic ticket or e-ticket should be widely applied in all the sector industries.[4]

I .3.2.What are the advantages and disadvantages of online booking systems?[5]

Advantages:

- Online booking system facilitates remote bookings and instant payments which is convenient to both; booker and the facility provider.
- Unlike traditionally, bookings can happen round the clock and all the information can be directly entered into the system which reduces human errors.
- Instant payment order on the web-site.
- The customer immediately receives a guarantee of obtaining services. Often online booking system uses a loyalty program, providing discounts and bonuses, thereby attracting more clients.
- The client chooses the desired services of a set of additional services, the exact time and date of booking.
- Business does not need to communicate with the client, as the book takes place automatically without administrator intervention.
- The system operates autonomously 24/7.
- It is possible to send SMS and push notifications.
- It is possible to receive payments via PayPal and other payment systems.
- No need to install software on your server or device.
- Time economy.

Disadvantages:

- Need stable internet access for the guest to book, while the facilitator has to do all the technical setups to offer online booking systems may it be railways, bus bookings or hotel bookings.
- Security: Though downtime and security breach instances are rare, they are for real. Various precautions can be taken by service provider like secure socket connection, PCI-DSS compliance, using reliable cloud service providers like Amazon AWS and so on.

Transaction drops: In some instances, booker is being charged for the services but the transaction is dropped at last moment resulting failure in allocation of services like train tickets, bus tickets etc. Though these can be taken care off by auto payment reversal mechanism, it takes time to get the amount credited.

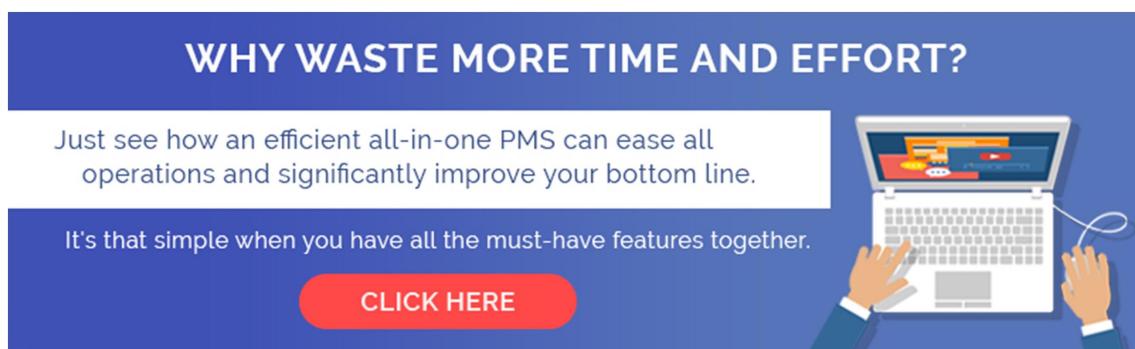


Figure I.4: Online Booking System advert

I .3.3. 9 good reasons to have an online booking system:[6]

The Internet is here to stay, and many businesses have made their first leap to the Internet with a web site, and many are now moving forwards using the Internet, not only as an advertising and promotion platform, but as an active business platform. The great thing about the Internet is that it can be used as the great equalizer for small and mid-sized businesses. The smallest Bed and Breakfast can today have as great a web presence as the largest 5 star hotels through their web site. Whether your site is for yacht charters, horse riding classes, or anything else, if your products and services can be booked, reserved or rented, you may want to try to make the web site work for your business all year round.

You may have asked yourself why so many small businesses are so interested in getting an online reservation system on their web site.

1. Open 24/7 to receive reservations

Many customers surf the Internet outside of business hours and they are more likely to make reservations on the spot that trying to remember to call back the next day. Statistics show that more and more bookings today are made during the evening at home online. The reasons for this are two fold. Firstly, many companies now monitor and control Internet access in the workplace, and secondly, nearly every home today has an Internet connection and a PC or laptop.

If your normal business hours are 9 to 5, then you have no one to take bookings after 17:00, and this is when we now know most online bookings occur.

An Online booking system allows you to receive booking 24 hours a day; therefore, your booking service is always open.

2. Commission Free

Whether you are paying for direct advertising, or paying a commission for bookings through booking portals, in either case, you are paying a portion of your income to these sources. If you have an online booking system on your own website, you have then cut out the middleman. Another important point when using commissioned portals is that your business is promoting them as much as they are promoting you, and with the Internet, you might as well just focus on promoting your own business.

3. It's simply good customer service

When we speak about online bookings here, we are not talking about a contact form. A contact form does not provide your customers with a clear view of your availability. A contact form does not allow your customers to book directly and pay and get confirmed automatically.

A proper online booking system must, at the least be able to perform these two tasks as they are core to providing good customer service.

4. Minimize your workload

If you are not using an online booking system today, then you may still be processing all your bookings manually. A good online booking system will handle all the aspects of the booking. It must, at a minimum, do the following:

- Automatically ensure that bookings can only be received when you have availability
- Get ALL the information required during the booking process so you don't have to waste time asking for more information
- Send out an automated email to the booking party as a confirmation of the booking

- Automatically update your availability when the booking has been processed A good booking system will do all this.

Also make sure your booking system doesn't just handle the online bookings, but handles email, phone and walk-in bookings as well.

5. All your customer data in a structured system

If you use a good online booking system which also handles the walk-in, email and phone bookings, then you will have all your previous customers data in a structured system. This is important today as it allows you to market yourself to your previous customers. This customer list is often one of your best-valued resources in a business, and you want this data both accessible and structured.

6. Up selling

This is not just the domain of large companies.

When a customer is booking a car or an apartment for their holidays, it's very common to have extras that the customer can choose. For apartment rentals it might be a welcome basket, for a car rental company, it may be a GPS system.

Your booking system should allow for the easy addition of extras so that you offer your customers all your great features and options.

7. Discount Codes

Discount codes are a great way to market your business in either off-season, or when you have some last spots available. Make sure your online booking system can support discount codes.

WebReserv makes it easy to download email addresses from previous customers so you can send them a comeback discount code. The post-visit email is great for this.

8. Clear and Simple Overview

Your online booking system should have at least one screen where you can see your availability very quickly and clearly. This removes the need for the old paper book, and allows your reception staff to easily see whether you have availability or not.

9. Online payments

For the fully comprehensive business, you may want to integrate an automatic payment system. With this in place, you reduce manual workload even more, and you provide your customers with a complete easy and safe way to both book and pay for your offerings

There are of course even more advantages, but the above list should at least get you thinking about whether you should have an online booking engine or not

WebReserv was founded in 2004 and provides all the above features and more.

The WebReserv booking and reservation system consists of a comprehensive back office reservation management system to handle all aspects of bookings, including, online, email, phone and walk-in bookings.

It includes automated emails, custom fields to handle extras, discount codes, reporting and many more features. The WebReserv booking widget is then placed on your web site, and handles your online booking 24/7.

You can create a free account with WebReserv to try the system and upgrade if you need more features.

CHAPTER II :CASE STUDY

II.1. INTRODUCTION

In this chapter, you are going to see the application interact with the users and the database, and also all the users activities all in the form of diagrams and UML modelization structures.

II.2. UML

Unified Modeling Language (UML) is a general purpose modelling language. The main aim of UML is to define a standard way to visualize the way a system has been designed. It is quite similar to blueprints used in other fields of engineering.

UML is not a programming language, it is rather a visual language. We use UML diagrams to portray the behavior and structure of a system. UML helps software engineers, businessmen and system architects with modelling, design and analysis. The Object Management Group (OMG) adopted Unified Modelling Language as a standard in 1997. Its been managed by OMG ever since. International Organization for Standardization (ISO) published UML as an approved standard in 2005. UML has been revised over the years and is reviewed periodically.[7]



Figure II.1: UML LOGO

II.2.1. Use case diagram:

A use case describes how a user uses a system to accomplish a particular goal. A use case diagram consists of the system, the related use cases and actors and relates these to each other to visualize: what is being described? (system), who is using the system? (actors) and what do the actors want to achieve? (use cases), thus, use cases help ensure that the correct system is developed by capturing the requirements from the user's point of view. [8]

II.2.2. Sequence diagram:

Sequence diagrams, commonly used by developers, model the interactions between objects in a single use case. They illustrate how the different parts of a system interact with each other to carry out a function, and the order in which the interactions occur when a particular use case is executed.

In simple words, a sequence diagram shows different parts of a system work in a 'sequence' to get something done.[9]

II.2.3. Class diagram:

The class diagram offers a prime example of the structure diagram type, and provides us with an initial set of notation elements that all other structure diagrams use.

The purpose of the class diagram is to show the types being modeled within the system. In most UML models these types include, a class, an interface, a data type and component.[10]

II.3.USE CASE DIAGRAM

The following diagrams will show how can users interact with our app and which action they may perform including the available actions for admins.

1. User activities:

The user here can perform several action while using the application in order to enquire a ticket

Check for ticket availability, fill an online form ,book a ticket and cancel it and in order to get the refund he must visit the clerk and fill a form and get refunded.

A user must log in to his session to perform any action on the application, otherwise any none registered one is not considered as a user because he cannot perform any action, so he must create an account and then log in.

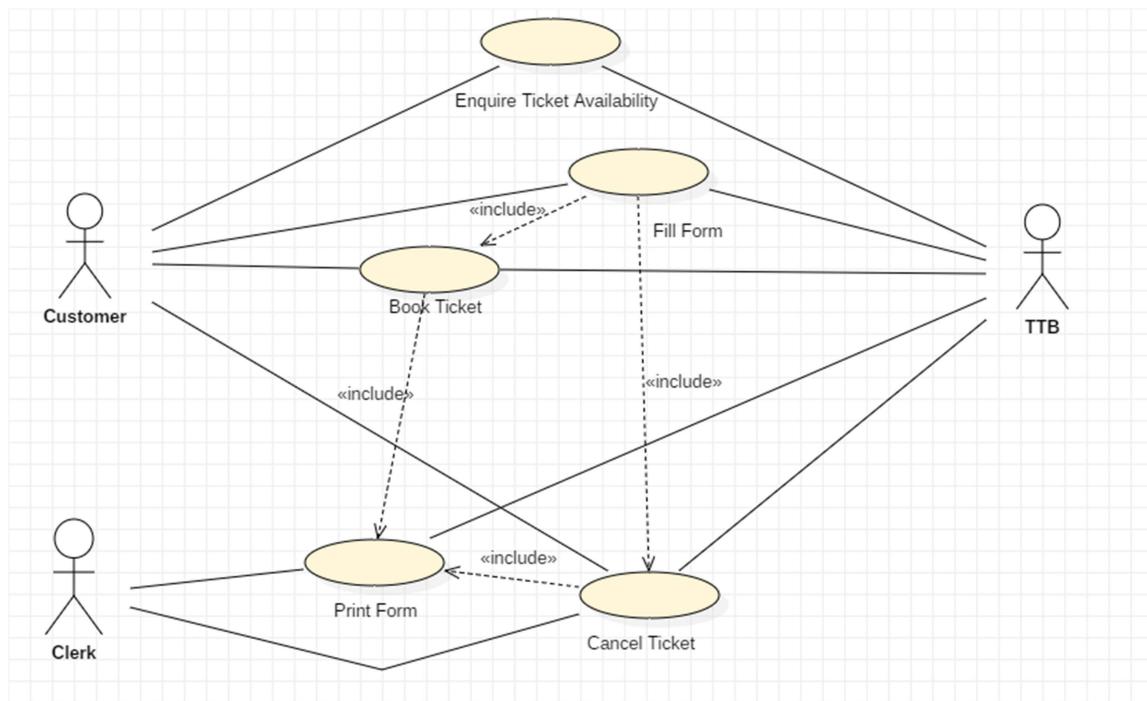


Figure II.2: Use case diagram (User activities)

1. Admin activities :

As shown in the diagram below admins are able to manipulate database from the console to add

Trips ,delete trips and modify trips information.

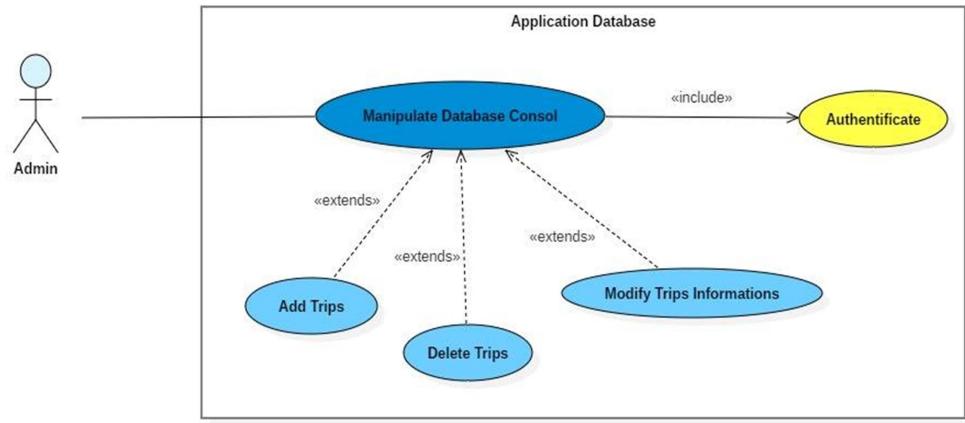


Figure II.3: Use case diagram (Admin activities)

II.4. App class diagram:

The class diagram will show the elements and attributes of the app and the relation between every element.

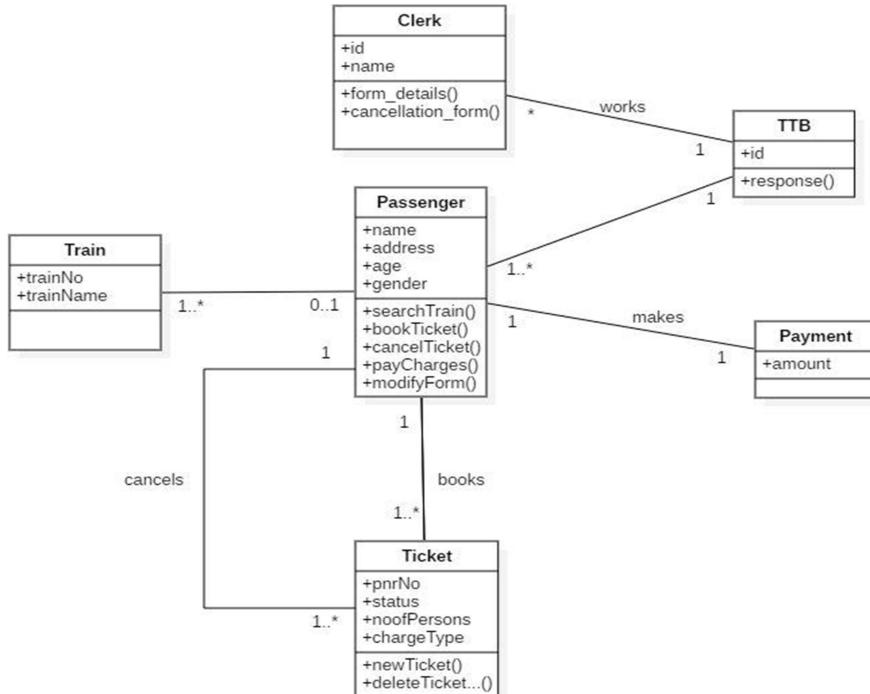


Figure II.4: Class Diagram

II.5. App sequence diagram:

II.5.1 Registration sequence diagram:

The following diagram will show how new user will be able to register in order to use the application and how its linked to our database .

1. After clicking on sign in ,the user must fill up all field requirement then click register then the system will process the data and verify the syntax ,if it find an blank or syntax error the system it will show where the error is if not it will send the information to the database.
2. The database register the users information and then send a verification email to the inbox included in the register.
3. In order to use the application the user must go to his inbox and verify the email and then his register process is finished.

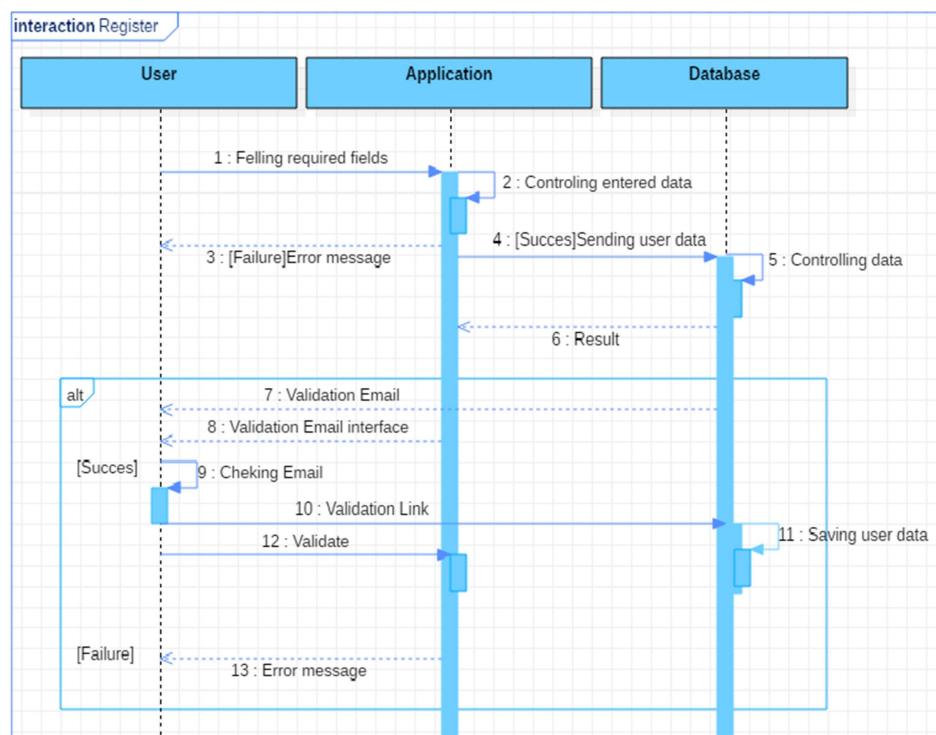


Figure II.5: Sequence Diagram (Registration)

II.5.2. Log in sequence diagram:

After registering, users must log in to their sessions in order to use the application.

As shown below in the diagram, user enter his Email and password then clicks on log in to his session, if the user is registered before he will be redirected to his profile, otherwise in case if he's not registered or entered an invalid email or password he receives an error message.

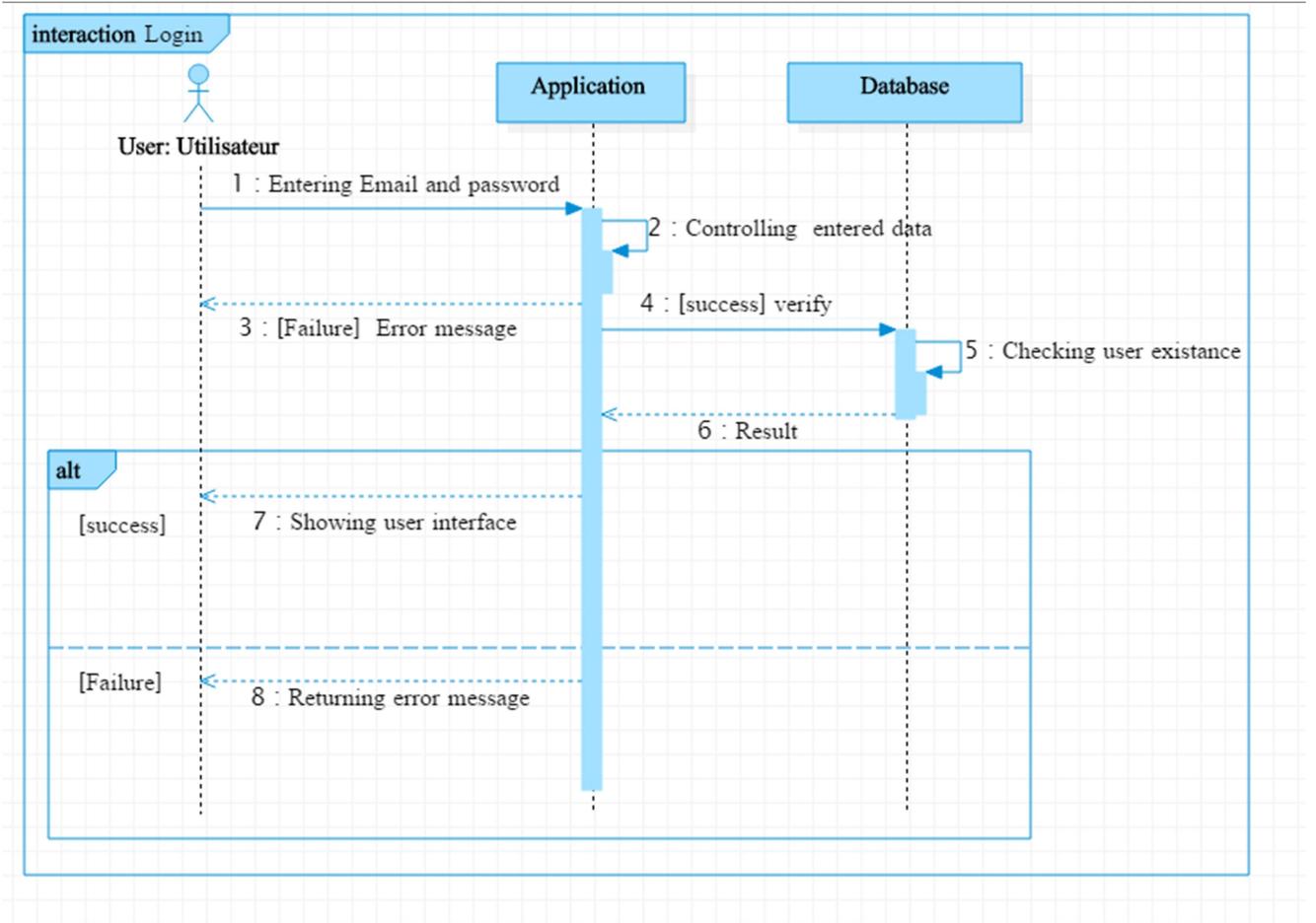


Figure II.6: Sequence Diagram (Login)

II.5.3. Ticket reservation sequence diagram:

The following diagram will explain how to book a ticket and view your booked tickets.

1. At first the user need to fill the trip information (starting destination, wanted destination, date , booked seats) then the system will control the entered data ,if it find a syntax error it will show error message if not send trip data to the database.
2. The database will check the trip availability ,if the trip is available it will register the user and the booked ticket will show on ticket interface, if not it will show a message that the trip isn't available anymore.

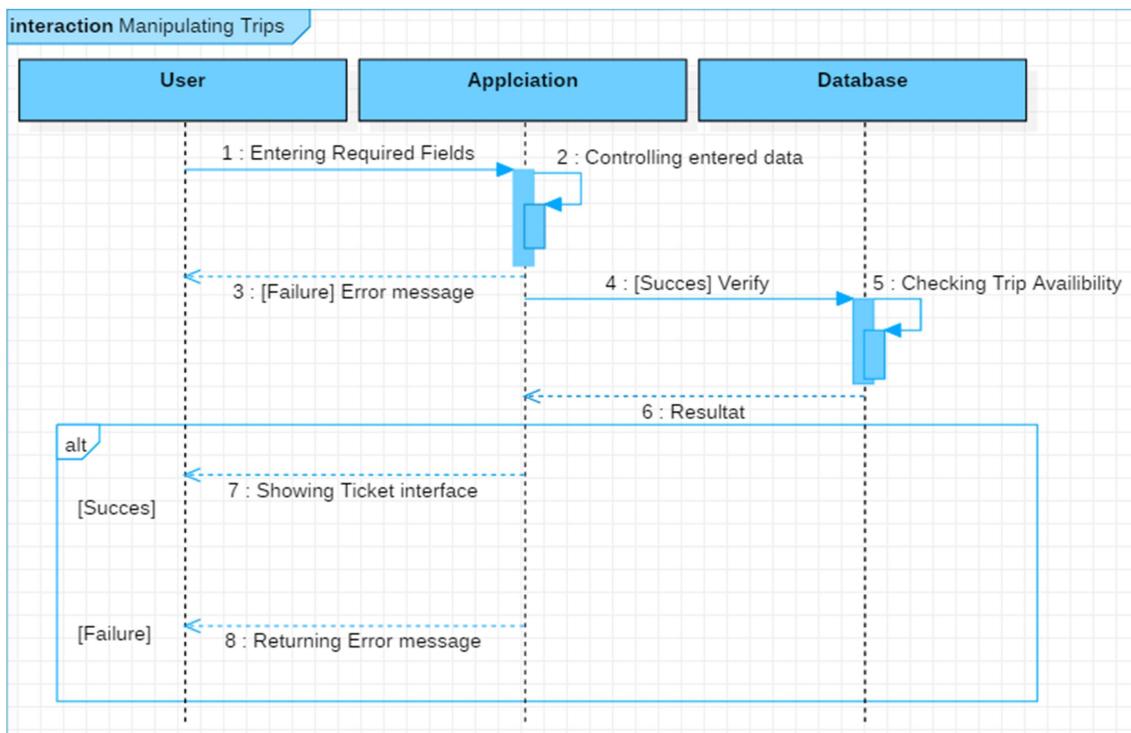


Figure II.7: Sequence Diagram (Booking)

II.6. CONCLUSION

A good UML realization is the most basic and most important step toward making a good application

Since it explain all the function and activities one can perform and the restriction , since after seeing all the diagrams one can understand the main available activities and actions.

CHAPITRE III: IMPLEMENTATION OF THE APPLICATION

III.1. Introduction:

In this chapter we are going to discover the used tool and programs used to develop this application and functionalities, it's components, parts and the backend system tools and implementation.

III.2. ANDROID STUDIO USED TOOLS

Android Studio is the official Integrated Development Environment (IDE) for Android app development, based on IntelliJ IDEA. On top of IntelliJ's powerful code editor and developer tools, Android Studio offers even more features that enhance your productivity when building Android apps, such as:

- A flexible Gradle-based build system
- A fast and feature-rich emulator
- A unified environment where you can develop for all Android devices
- Instant Run to push changes to your running app without building a new APK
- Code templates and GitHub integration to help you build common app features and import sample code
- Extensive testing tools and frameworks
- Lint tools to catch performance, usability, version compatibility, and other problems
- C++ and NDK support
- Built-in support for Google Cloud Platform, making it easy to integrate Google Cloud Messaging and App Engine[11]



Figure III.1: Android Studio LOGO

Project Structure

Each project in Android Studio contains one or more modules with source code files and resource files. Types of modules include:

- Android app modules
- Library modules
- Google App Engine modules

By default, Android Studio displays your project files in the Android project view, as shown in “figure III.2”. This view is organized by modules to provide quick access to your project's key source files.

All the build files are visible at the top level under Gradle Scripts and each app module contains the following folders:

- manifests: Contains the AndroidManifest.xml file.
- java: Contains the Java source code files, including JUnit test codes. Contains all non-code resources, such as XML layouts, UI strings, and bitmap images.[12]

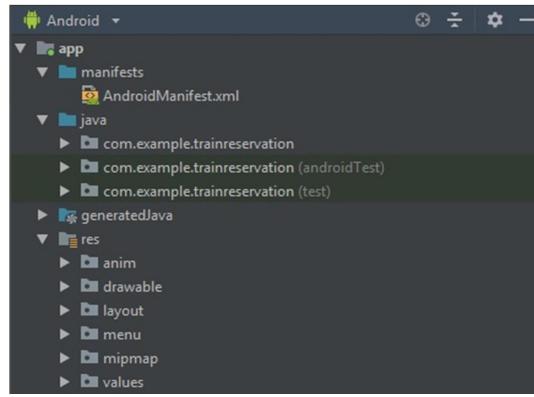


Figure III.2: The Project Files in our App

III.2.1.Used Dependencies:

The Gradle build system in Android Studio makes it easy to include external binaries or other library modules to your build as dependencies. The dependencies can be located on your machine or in a remote repository, and any transitive dependencies they declare are automatically included as well.[13]

As shown in the figure below those are the main used dependencies in our application:

- o Firebase data base dependencies
- o Card view dependencies
- o List view dependencies
- o Recycler view dependency
- o And other dependencies for gradle build and kotlin

```

dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation 'com.android.support:appcompat-v7:28.0.0'
    implementation 'com.android.support.constraint:constraint-layout:1.1.3'
    implementation 'com.google.firebase:firebase-auth:16.0.3'
    implementation 'com.google.firebase:firebase-database:16.0.1'
    implementation 'com.google.firebase:firebase-storage:16.0.1'
    implementation 'com.google.firebaseio:firebase-core:15.0.0'

    implementation 'com.android.support:support-v4:28.0.0'
    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'com.android.support.test:runner:1.0.2'
    androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.2'
    implementation group: 'com.squareup.picasso', name: 'picasso', version: '2.5.0'
    implementation 'com.google.code.gson:gson:2.8.5'
    implementation 'de.hdodenhof:circleimageview:3.0.0'
    implementation 'com.github.arcadefire:nice-spinner:1.3.8'

    implementation 'com.github.bumptech.glide:glide:4.7.1'
    // Glide v4 uses this new annotation processor -- see https://bumptech.github.io/glide/doc/generatedapi.html
    annotationProcessor 'com.github.bumptech.glide:compiler:4.7.1'
    //BottomNavigationViewEx
    api 'com.github.ittianyu:BottomNavigationViewEx:2.0.2'
    implementation 'com.android.support:recyclerview-v7:28.0.0'
    implementation 'com.android.support:cardview-v7:28.0.0'

    // circle image
    implementation 'de.hdodenhof:circleimageview:3.0.0'

    // Recycler view
    implementation 'com.android.support:recyclerview-v7:28.0.0'
}

```

Figure III.3: The Application Dependencies

III.3. XML CODE

XML stands for Extensible Markup Language. XML is a markup language much like HTML used to describe data. XML tags are not predefined in XML. We must define our own Tags. XML as itself is well readable both by human and machine. Also, it is scalable and simple to develop. In Android we use XML for designing our layouts because XML is a lightweight language so it doesn't make our layout heavy.[14]

The whole concept of Android User Interface is defined using the hierarchy of View and ViewGroup objects. A ViewGroup is an invisible container that organizes child views. These child views are other widgets which we use to make the different parts of UI. One ViewGroup can have another ViewGroup as a child element as shown in the figure given below.[15]

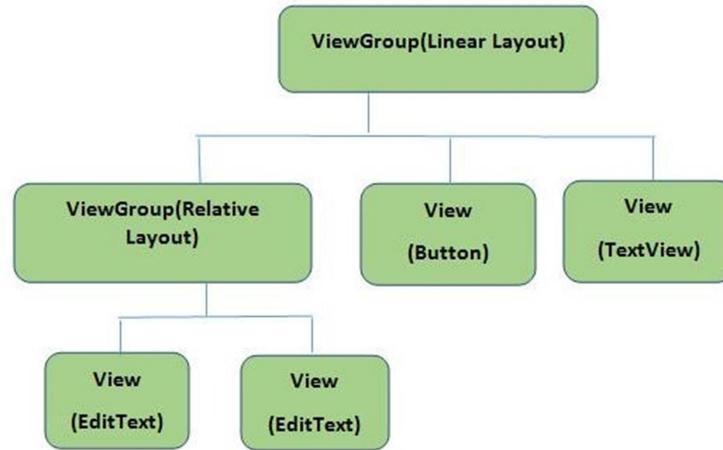


Figure III.4: Diagram of hierarchy for View and View Group objects

III.3.1.Different XML Files Used in Android:[15]

- Layout XML Files: Layout xml files are used to define the actual UI(User interface) of our application. It holds all the elements(views) or the tools that we want to use in our application.

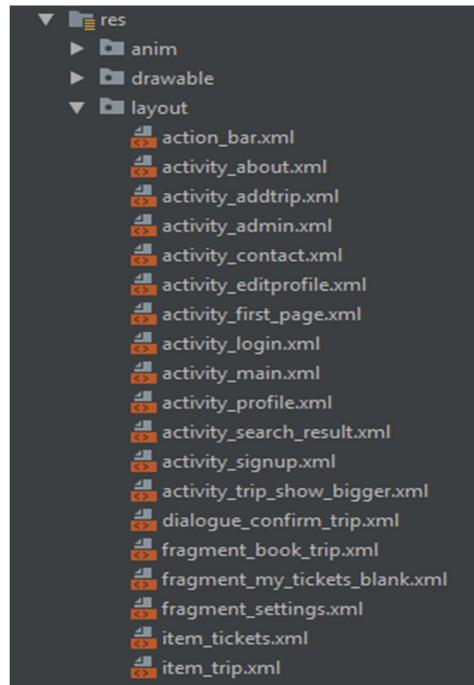


Figure III.5: Layout xml files

- Manifest xml File(Mainfest.xml): This xml is used to define all the components of our application. It includes the names of our application packages, our Activities, receivers, services
1. Color xml File (colors.xml): This file is used to define the color codes that we used in our app. We simply define the color's in this file and used them in our app from this file.
 2. Strings xml File (strings.xml): This xml file is used to replace the Hard-coded strings with a single string. We define all the strings in this xml file and then access them in our app(Activity or in Layout XML files) from this file. This file enhance the reusability of the code.
 3. Drawable xml Files: These are those xml files that are used to provide various graphics to the elements or views of application. When we need to create a custom UI we use drawable xml files. Suppose if we need to define a gradient color in the background of Button or any custom shape for a view then we create a Drawable xml file and set it in the background of View.

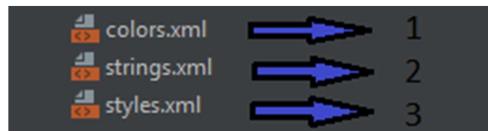


Figure III.6: XML files

III.3.2.User interfaces:

What is considered as a good user interface design is the one that the user can easily and intuitively interact with the application or the system[16]. To build any powerful application, you need to build first a good UI, with the appropriate design, colors, icons and system interactions for the use purpose of the application and users nature, because every kind of application has a design pattern to follow.

In our case, a social network we chosen the common colors for almost any social network, Blue, White, Dark, green and grey and some other secondary colors.

```

3   <color name="colorPrimary">#17394d</color>
4   <color name="colorPrimaryDark">#00574B</color>
5   <color name="colorAccent">#026aa7</color>
6
7   <color name="white">#FFFFFF</color>
8   <color name="ivory">#FFFFF0</color>
9   <color name="light_yellow">#FFFFE0</color>
10  <color name="yellow">#FFFF00</color>
11  <color name="snow">#FFFafa</color>
12  <color name="floral_white">#FFFaf0</color>
13  <color name="lemon_chiffon">#FFFACD</color>
14  <color name="cornsilk">#FFF8DC</color>
15  <color name="seashell">#FFFSEE</color>
16  <color name="lavender_blush">#FFF0F5</color>
17  <color name="papaya_whip">#FFEFD5</color>
18  <color name="blanched_almond">#FFEBCD</color>
19  <color name="misty_rose">#FFE4E1</color>
20  <color name="bisque">#FFE4C4</color>
21  <color name="moccasin">#FFE4B5</color>
22  <color name="navajo_white">#FFDEAD</color>
23  <color name="peach_puff">#FFDAB9</color>
24  <color name="cold">#FFD700</color>
25  <color name="pink">#FFC0CB</color>
26  <color name="light_pink">#FFB6C1</color>

```

Figure III.7: Colors used in the application

Blue and white as primary colors of the application because they provide a lighter relaxing look for the eyes and make photos, icons and the content view very clear and appropriate.
 Red color for the alerts and verification buttons, and Black color for the text colors. Some layouts of the application in the figure

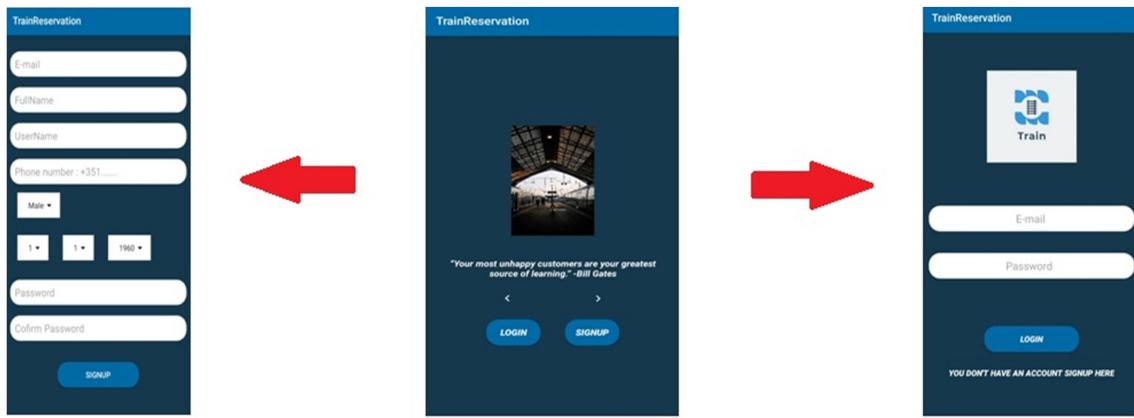


Figure III.8: Some application layouts

III.3.3. Icons and images:

TTB (Train Ticket Booking) is the name of the application ,simple and easy to understand even if u are none english speaker from only train ticket most of the common people will understand what it means.

The logo below shows the railroad which imply for the train services and have train written for anyway that missed the point .



Figure III.9: The logo of our application

The folder drawable contain all the icons used in the app with all the deferent sizes (Mdpi, Xhdpi, XXhdpi, XXXhdpi), to give the app a proper design

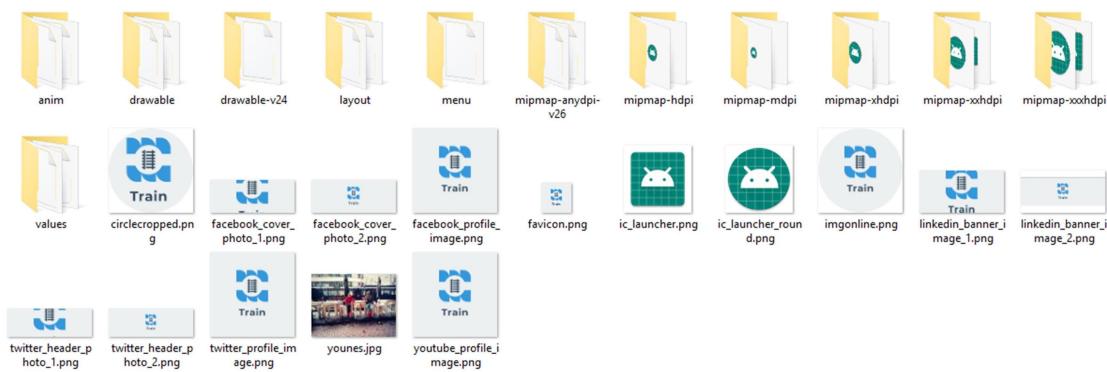


Figure III.10: Icons used in the application

III.4. APPLICATION FONCTIONALITIES

Every user after being registered to our system can perform the following actions:

- Log in and log out any time secure and safely, because we insure exclusive access to sessions by following a verification and validation protocols (email validation) while registering every user in our system.

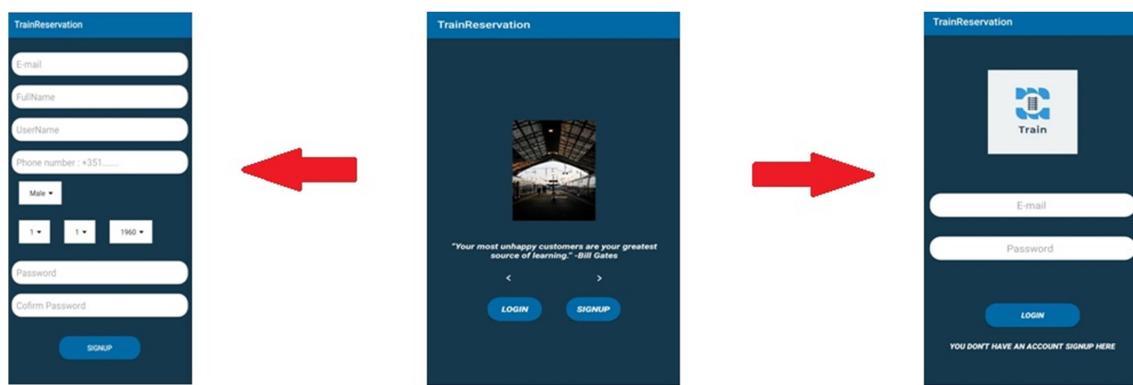


Figure III.11: Register and authentication system

- Users can change their own personal information from their profiles (Password reset ...) which insure flexibility and availability in case of forgetting password or to protect themselves from security threats.

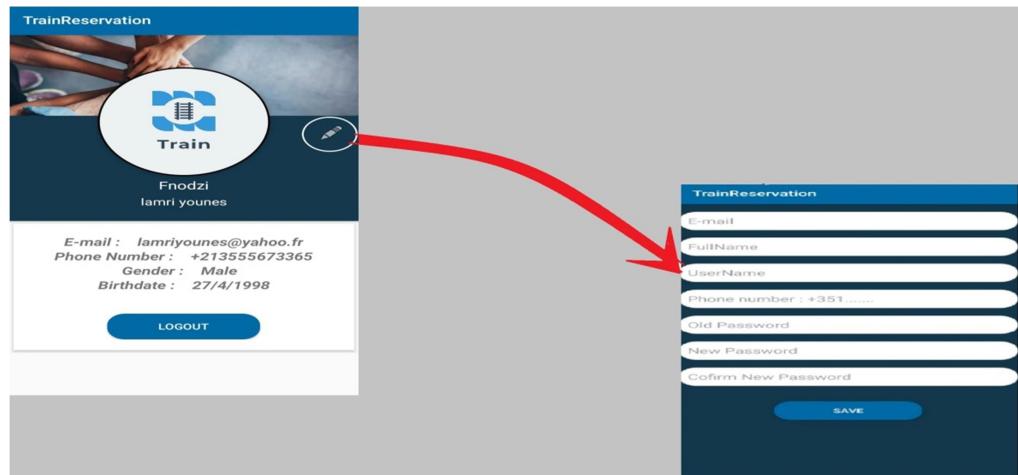


Figure III.12: User profile security options

- Users can select book trips to reserve a ticket but they must fill all the required information and then search and will find matching results.
- Afterwards booked ticket will appear on my tickets tap at first it will blank but later on after booking it will be filled with all there reserved data.

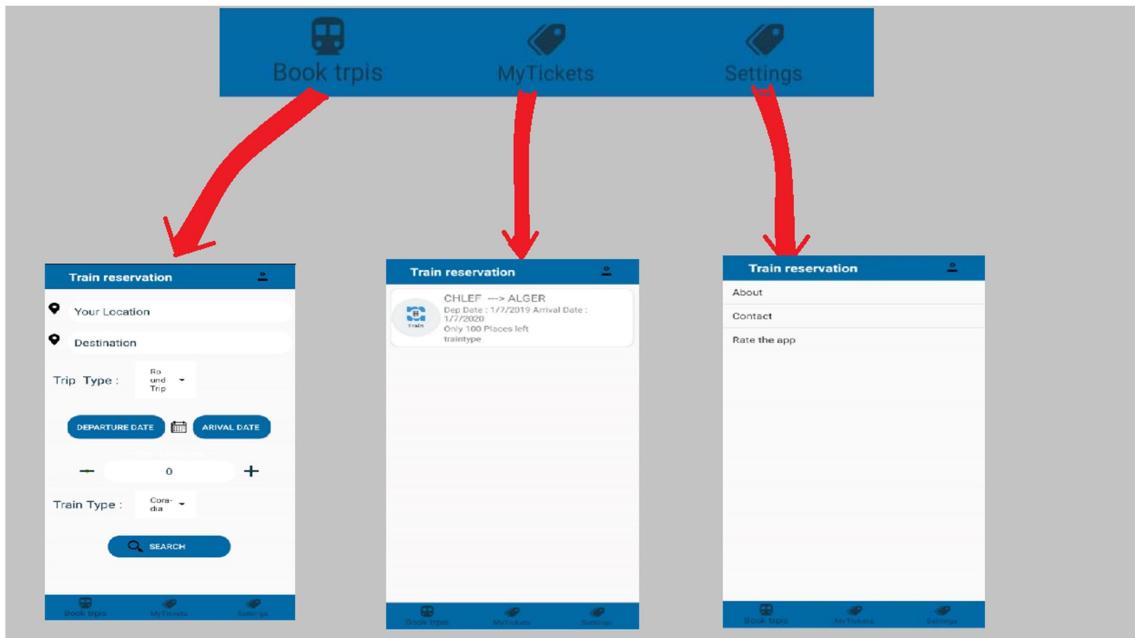


Figure III.13: Application different functionalities

- If the user is looking to contact us to report a bug and know more about the app he can select contact.
- Or if he want to rate our application on the android store he click rate up and give his feedback to try and improve whatever the inconvenient was.

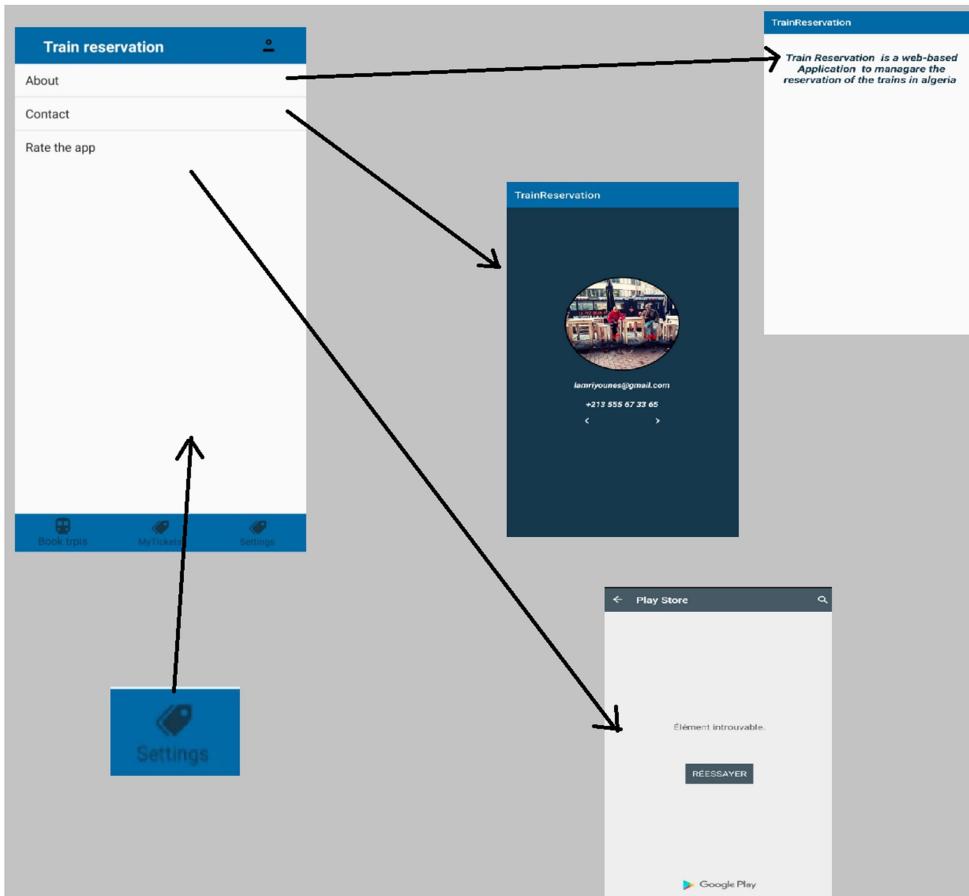


Figure III.14: Application Settings

III.5. FIRE BASE DATABASE

Store and sync data with our NoSQL cloud database. Data is synced across all clients in realtime, and remains available when your app goes offline.

The Firebase Realtime Database is a cloud-hosted database. Data is stored as JSON and synchronized in realtime to every connected client. When you build cross-platform apps with our iOS, Android, and JavaScript SDKs, all of your clients share one Realtime Database instance and automatically receive updates with the newest data.[17]

We used firebase because it's the most common and the most secure, it allow different ways of authentications and register, such as (facebook, google, github, twitter....), also many rules, statistics...

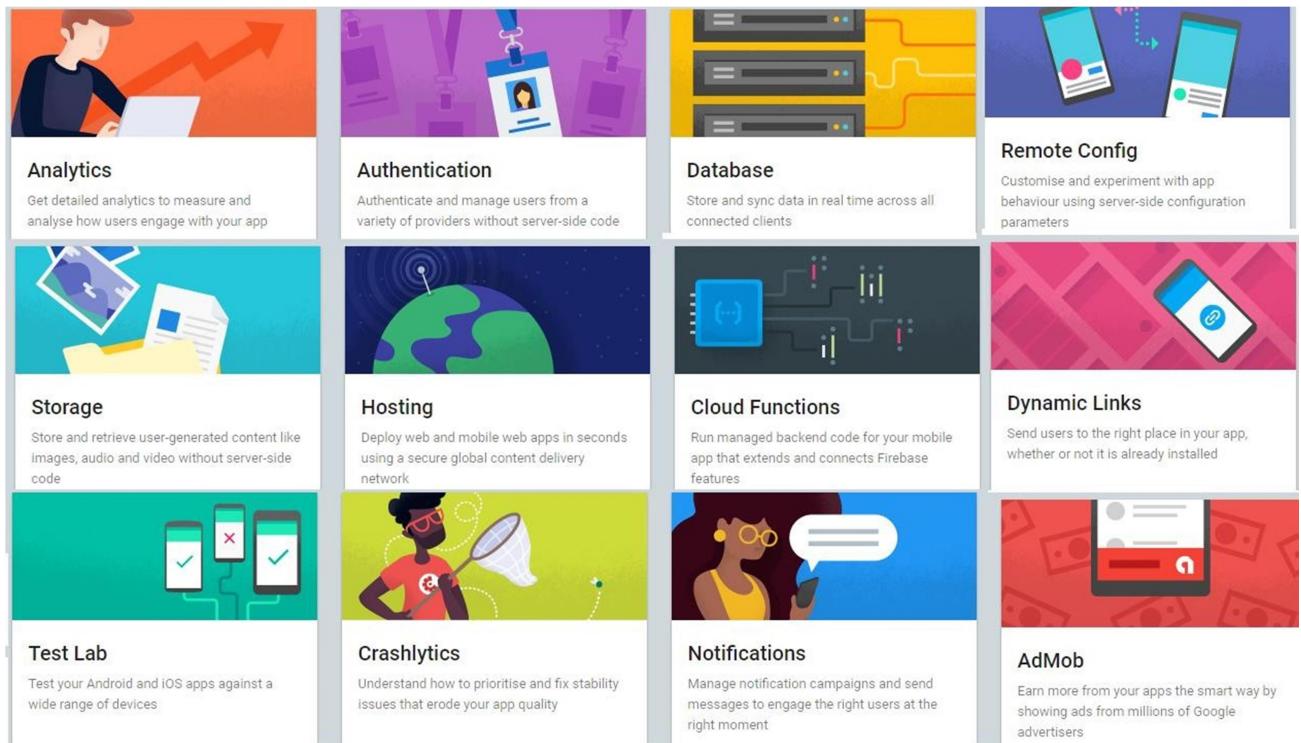


Figure III.15: Firebase Features

III.5.1.Database structures:

All Firebase Realtime Database data is stored as JSON objects. You can think of the database as a cloud-hosted JSON tree. Unlike a SQL database, there are no tables or records. When you add data to the JSON tree, it becomes a node in the existing JSON structure with an associated key. You can provide your own keys, such as user IDs or semantic names, or they can be provided for you using push().[18]

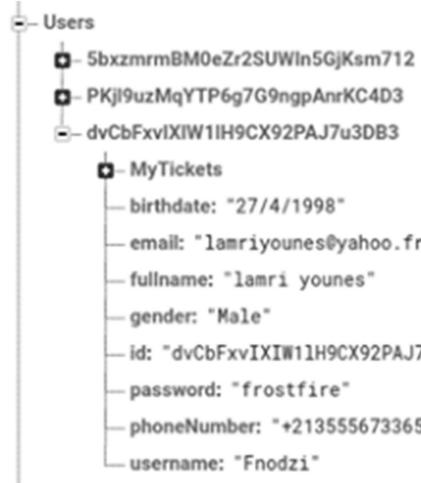


Figure III.16: Firebase data structures

III.5.2. FIREBASE ADMIN CONSOL

Is the console where admins can manipulate and control the application database, such as stored data, security rules, authentication methods, data structures, enable or disable features, check statistics, enable email and password reset options, and many other functionalities we have already seen in Firebase features figure.

The access to the console needs a google account authentication, which provides a high level of security (Google company) and exclusive access for admin.

III.5.3. Firebase used functionalities:

In our applications we've used:

A registered user can access to his session authenticating using email and password.

We used Email address verification for security reasons and to avoid using invalid emails and robot problems.

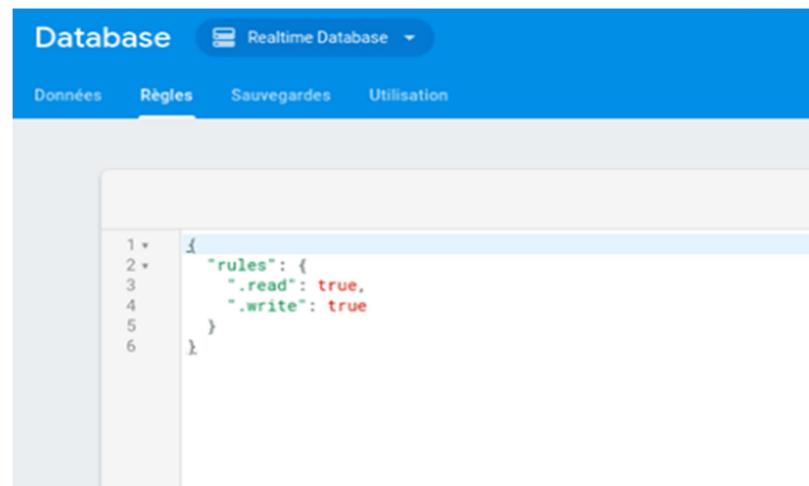
We used Password reset in order to allow users to reset their passwords in case of insecurity, possible threats or leaked informations.

Fournisseur	État
✉ Adresse e-mail/Mot de passe	Activé
📞 Téléphone	Désactivé
GOOGLE Google	Désactivé
PLAY Jeux Play Jeux	Désactivé
GAME Game Center Beta	Désactivé
FACEBOOK Facebook	Désactivé
TWITTER Twitter	Désactivé

Figure III.17: Authentication used providers

Realtime database: to store synchronized information and data, like trips information, friend lists, chat messages....

In addition, fix security rules for enabling reading and writing in the database.



```

Database   Realtime Database
Données   Règles   Sauvegardes   Utilisation

1 * {
2 *   "rules": {
3 *     ".read": true,
4 *     ".write": true
5 *   }
6 *
}

```

Figure III.18: Realtime database rules

III.6.CONCLUSION

We learnt from this chapter that to build a powerful application we have to use the perfect tools and equipment that is needed to facilitate processes and simplify the work and to insure security, availability, flexibility and users satisfaction.

IV. GENERAL CONCLUSION

When we were young we used to wonder how these kind of application were made, seeing now that we made one our own fills me with joy and fulfillment

This project allowed us to implement our minds of study, analysis and criticism. To put into practice our knowledge and knowledge acquired during our training at the university but also to discover some undeveloped points in Course. We also learned a lot. Humanly, thanks to the communication with our supervisor to establish the specifications and follow the progress of the work, and between us to divide the tasks and manage our differences. The first phase was devoted to the study of the existing at which we could present the field of study. This part was followed by the study of the problem, which allowed us to lay the first stone of the building by proposing a solution, Finally by specifying the expected requirements of the system, to clarify our needs and objective. The design phase allowed us to present the system in more detail, removing all abstraction. This phase indicated the structure and method to follow during the implementation phase. Finally, the production phase was the object of the presentation of the tools and technologies used as well as the presentation of the final product obtained through the description of certain interfaces of the application.

Developing a perfect application within a time frame that has been granted can be difficult. However, our application is operational and meets the specifications established at the beginning of the project.

V.REFERENCE

- [1] <https://www.bebee.com/producer/@mithunbanerjee/do-you-know-the-future-of-mobile-app-development>
- [2] https://en.wikipedia.org/wiki/Mobile_operating_system
- [3] James Chen :<https://www.investopedia.com/terms/a/android-operating-system.asp>
- [4] <https://www.slideshare.net/NimRaHNaZaR/train-ticketing-system-65518560>
- [5] EZee Technosys, Complete Hospitality Technology Solutions Provider
- [6] <https://www.geeksforgeeks.org/unified-modeling-language-uml-introduction/>
- [7] <https://online.visual-paradigm.com/tutorials/use-case-diagram-tutorial/>
- [8] <https://creately.com/blog/diagrams/sequence-diagram-tutorial/>
- [9] <https://www.lucidchart.com/pages/uml-class-diagram>
- [10] Android Developer Website: <https://developer.android.com/studio/intro>
- [11] UNIVERSITY OF VAASA FACULTY OF TECHNOLOGY - Tritonia
- [12] Android Developer Website
<https://developer.android.com/studio/build/dependencies>
- [13] <https://searchmicroservices.techtarget.com/definition/XML-Extensible-Markup-Language>
- [14] visited 20/05/2018, URL: <http://abhiandroid.com/ui/xml>
- [15] Radoslava Kraleva, Designing an Interface For a Mobile Application Based on Children’s Opinion, South-West University “Neofit Rilski”, Blagoevgrad, Bulgaria,p57, URL: <https://doi.org/10.3991/ijim.v11i1.6099>.
- [16] licensed under the Creative Commons Attribution 3.0 License, mai 8, 2018, visited 20/05/2018, URL: <https://firebase.google.com/docs/database/>
- [17] Licensed under the Creative Commons Attribution 3.0 License, mai 8, 2018, visited 20/05/2018, URL:
<https://firebase.google.com/docs/database/web/structure-data>