

DEMOCRATIC AND PEOPLE'S REPUBLIC OF ALGERIA
MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH MOHAMED
BOUDIAF UNIVERSITY - M'SILA

FACULTY OF MATHEMATICS AND
COMPUTER SCIENCE
DEPARTMENT OF COMPUTER SCIENCE
N°:.....



DOMAIN: Mathematics And Computer
Science
FIELD : Computer Science
OPTION: SIGL(Information System And
Software Engineering)

*Thesis presented for obtaining
the degree of Academic Master*

Directed by: Hedjouli Yacine

Entitled

**design and realization of a smartphone game on the
history and geography of the region
(M'sila)**

Defended before the jury composed of:

Mr. Tahri Zohair

University of M'sila

President

University of M'sila

Rapporteur

University of M'sila

Examiner

Academic year: 2020/2021

Thanks

First of all, i would like to thank Allah the Almighty and Merciful, who gave me the strength and patience to do this modest work.

I would like to express my sincere gratitude to my coach Mr.Tahri zohair for his advices and encouragements throughout this project.

I would like to express my gratitude to all the jury members for their willingness to judge my work.

Also My parents who have always supported and encouraged me during this brief.

I would also like to express my sincere thanks to all the teachers in the department of computer science who have contributed to my training.



Dedications

I dedicate this modest work to:

Dear parents, may no dedication be able to express my sincere sentiments, their help, their encouragement and their unlimited love and patience;

To my dear grandmother ferhati saadia and my aunt delia zohra , for thier help and encouragement ;

To my dear brothers and sisters, Hiba, Mahdi, Iyad, Farah;

a very special dedication to my friends : Saidani Dirar, Ferhati mustapha, Azzeddine Brahim, Meghnia Abdelkader, Hadibi Aziz, Meftah Mohamed Elamine, Daikache Said , Kamel Salah Eddine, Morechta Lamine Slimani Taher , and to all my friends and familly ;

To all promotion 2021 especially "SIGL";

Abstract

The project carried out during this end of studies, consists in developing a smartphone game on the history and geography of the region (M'sila)

The aim of this work is to achieve an enjoyable and easy application, equipped with the necessary functionalities and adapted to the needs of our project. Our project aims to provide self-motivated users with options for learning in gaming environment . our applications is presented in form of quiz application wich try to deliver historical as well as geographical informations related to our state (M'sila)

To implement our solution, we opted,UML as a modelling language, DART as a programming language by using it in the FLUTTER cross platform service under the ANDROID STUDIO platform

Table Of Contents

- Thanks 1**
- Dedications 2**
- Abstract 3**
- Table Of Contents..... 4**
- Liste of figures..... 6**
- General Introduction..... 7**
- Organization of the brief..... 8**
- 1 Mobile application 10**
 - 1.1 Definition..... 10
 - 1.2 Types 10
 - 1.2.1 Native or embedded applications : 10
 - 1.2.2 Web application : 10
 - 1.2.3 Hybrid applications : 10
 - 1.3 Benefits of mobile applications 11
 - 1.4 Disadvantages of mobile applications 11
 - 1.5 Most popular mobile operating systems 11
 - 1.6 Market Share 12
 - 1.7 Discovery of the ANDROID platform..... 13
 - 1.7.1 Presentation of Android..... 13
 - 1.7.2 History..... 14
 - 1.7.3 Chronology..... 15
 - 1.7.4 Versions of ANDROID 15
 - 1.7.5 Architecture of Android 16
 - 1.8 Why did we choose ANDROID ? 17
 - 1.8.1 The system 18
 - 1.8.2 Data management..... 18
 - 1.8.3 Optimization for Google Services 18
 - 1.8.4 The cost 18
 - 1.8.5 System position and rate of progress 19
 - 1.9 Conclusion..... 19

2	Games Application	21
2.1	Definition.....	21
2.2	History	21
2.3	Types	22
2.4	Mobile games advantages.....	23
2.4.1	Huge Potential of Consumers	23
2.4.2	Portability.....	23
2.4.3	Network.....	23
2.5	Mobile gaming applications as an educational platform.....	24
2.6	Conclusion.....	26
3	Conception and realization	28
3.1	Conception.....	28
3.1.1	Introduction	28
3.1.2	UML.....	28
3.1.3	UML diagrams	31
3.1.4	IHM presentation.....	36
3.2	Realization.....	43
3.2.1	Introduction	43
3.2.2	Technical Choices	43
3.2.3	difficulties encountered	53
3.3	Conclusion.....	53
4	General conclusion	54
	Bibliography.....	55

Liste of figures

FIGURE 1 1 DISTRIBUTION OF SMARTPHONE SHIPMENTS WORLDWIDE BY OPERATING SYSTEM BETWEEN 2013 AND 2022	13
FIGURE 1 2 ANDROID LOGO	14
FIGURE 1 3 – EVOLUTION OF ANDROID VERSIONS.....	16
FIGURE 2 1 NOKIA SNAKE GAME	22
FIGURE 3-1 CLASS DIAGRAM OF THE SYSTEM.....	31
FIGURE 3-2 USE CASE DIAGRAM OF THE SYSTEM	32
FIGURE 3-3 SEQUENCE DIAGRAM << FACEBOOK LOGIN >>	33
FIGURE 3-4 SEQUENCE DIAGRAM << GAME PLAY >>	34
FIGURE 3-5 ACTIVITY DIAGRAM << GAME PLAY >>.....	35
FIGURE 3-6 SPLASH SCREEN	36
FIGURE 3-7 HOME SCREEN << ENGLISH >>	37
FIGURE 3-8 HOME SCREEN << ARABIC >>	38
FIGURE 3-9 DASHBOARD << ENGLISH >>	39
FIGURE 3-10 DASHBOARD << ARABIC >>.....	40
FIGURE 3-11 PLAY SCREEN.....	41
FIGURE 3-12 CORRECT ANSWER	42
FIGURE 3-13 INCRRRECT ANSWER	42
FIGURE 3-14 ANDROID STUDIO LOGO	52
FIGURE 3-15 DART LANGUAGE LOGO.....	52

General Introduction

Game based mobile learning is becoming increasingly popular, now that mobile devices providesupport for multimedia content, location awareness, augmented reality and connectivity. However just having technical features does not make a game either engaging or pedagogical. The challenge for designers of games for mobile learning is to embed both effective gaming experiences and worthwhile learning outcomes into the same application

Mobile learning comes into view as one of the best options to solve the educational challenges by using too many different resources and tools that are available, Mobile learning within schoolroom allow students working in correlating way in order to solve many problems, working on their projects and sharing opinions, also allowing them to access into available content anywhere and anytime ,many previous Studies show that tablets, iPod, iPads and other handheld devices are very communal devices to use it in mobile learning because of their availability in suitable cost

In this project we try to build a smartphone gaming application wich responde to the users needs in a enjoyable as well effective way. this application is a quiz application about the history of m'sila state

Organization of the brief

The brief is organized into three chapters:

- In the first chapter, we present the different basic concepts that concern phone applications in general as well as the development technologie we are using.
- The second chapter we go more further to presents one of the most popular types of phone applications wich is game application and its different concepts asa well as the benefits of using application as an educational platform
- The third chapter presents the requirements specification part of our application. These requirements are specified using the different diagrams. And we reserved a part for the presentation of the development tools used as well as the architecture and screenshots of our application.
- Finally, we conclude this brief with a general conclusion .

Chapter I :

Mobile Applications

1 Mobile application

1.1 Definition

A mobile app is a program designed to work on smartphone, tablet And other mobile devices, it can be downloaded for free or for a fee from applications stores. The most popular shops for apps are App store, Google Play and Windows Store [1].

1.2 Types

There are three types of mobile applications [2]:

1.2.1 Native or embedded applications :

a native application is a mobile application that has been developed for use on a particular platform or device. It is installed on the device and responds faster than a web application because that the interface is more direct. A native application is downloaded from a store application and installed on the device .

1.2.2 Web application :

it is an application program that is stored on a server remote as a normal web application. The web application is accessible and executable on all smartphones via their web browser.

1.2.3 Hybrid applications :

hybrid applications are applications that combine the elements of a web application and the elements of a native application. It must be installed in devices.

1.3 Benefits of mobile applications

Mobile applications have many advantages such as:

- Chat and real-time data sharing applications support communication, collaboration and knowledge building. This allows students to consume and create information collectively and individually [3].
- Direct access to the content of the mobile application via the icon on the phone or tablet.
- Offline operation for some applications.

1.4 Disadvantages of mobile applications

- Submission to standards and rules published by mobile platform companies namely Apple, Google, Windows and others
- A heavy investment for the development of a mobile application suitable for each mobile operating system, contrary to the cost development of a mobile site
- When updating a mobile application, the mobile user is obliged to do so through the store while the mobile site is updated in an automatic manner [4].

1.5 Most popular mobile operating systems

Mobile operating system, commonly known as Mobile OS is a software platform that controls all the features of mobile devices. the mobile OS is responsible for the way users interact with a device. This includes the look and feel, overall design, and functionality, with each of these elements crucial to ensuring a positive user experience. Below we represent the best-known mobile operating systems:

- **Android** : is a mobile operating system that is based on a modified version of Linux. It was originally developed by a startup of the same name, Android, Inc. In 2005, as part of its strategy to enter the mobile space, Google has purchased

Android and has resumed its development work (as well as its development team). Android is a free and fully open OS, which means the source code and APIs are either green. Thus hardware manufacturers can add their own proprietary extensions to Android and customize it to differentiate their products from others [5].

- **Windows Phone:** is the name of the operating system for smartphones that Microsoft has published in October 2010, as replace Windows Mobile by introducing a completely redesigned user interface designed for touch screen devices. From 2010 to 2015, Windows Phone has experienced several major updates from the Windows CE (Windows Phone 7) kernel to the Windows NT (Windows Phone 8) kernel and the introduction of the Cortana voice assistant (Windows Phone 8.1). Starting in November 2015, Windows Phone gradually disappears and it is replaced by Windows 10 Mobile [6].
- **iOS:** (formerly called «iPhone OS») is a mobile operating system developed by Apple Inc. For various Apple devices such as iPhone, iPod Touch, iPad and even for the second-generation Apple TV. Unlike Google's Android and Microsoft's Windows Phone, iOS use is limited to Apple devices only [7].
- **BlackBerry OS :** is a proprietary mobile operating system developed by the Canadian company Research In Motion (RIM) for its range of BlackBerry smartphones [8].

1.6 Market Share

In this section we present a statistic published by Statista [9] on the main mobile operating systems,

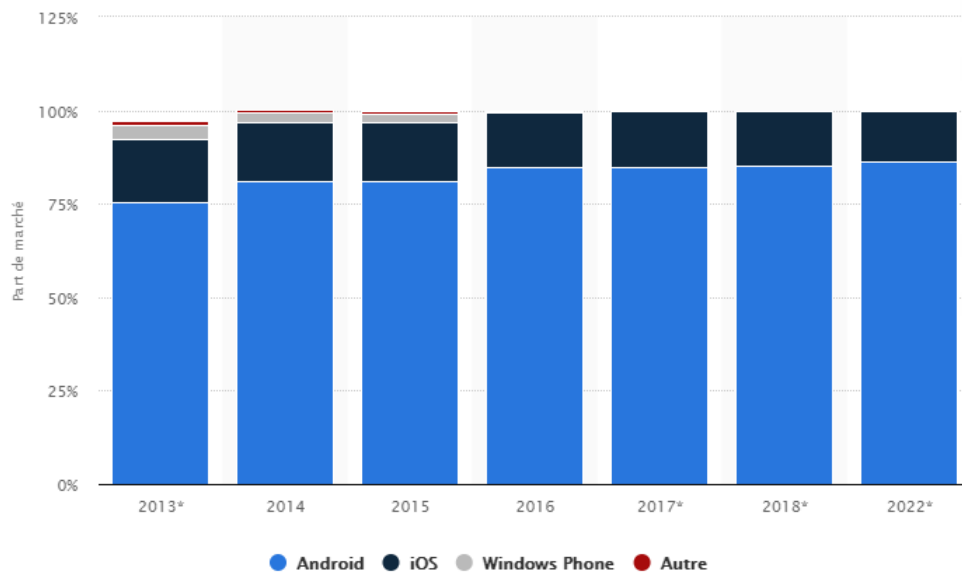


Figure 1 1 Distribution of smartphone shipments worldwide by operating system between 2013 and 2022

1.7 Discovery of the ANDROID platform

1.7.1 Presentation of Android

Android is an open source software platform for mobile devices which integrates different components: an operating system, a middleware, a graphical interface and A range of apps. So Android is not bound to a device, a mobile phone manufacturer, or an operator. Google has federated around thirty android companies (including Samsung, Motorola, Sony Ericsson and LG) inside the Open Handset Alliance to mass broadcast its system.

Android lets developers create their own apps using the Java programming language, and Google provides developers with many tools to write and develop their programs: emulator for tests, SDKs, and plugins for development, as well as provided

documentation. Android relies on a Linux kernel that manages system services like security, memory management, process management, network stack, etc [10]



Figure 1 2 Android Logo

1.7.2 History

The story began in October 2003, in California. A start-up was founded by Andy Rubin, Rich Miner, Nick Sears and Chris White to develop applications for mobile devices. This company was named android. Two years later, in July 2005, Google buys the company and founders continue to work in android Inc., We are bringing fixed internet into the mobile world in a pretty cool way,” said Andy Rubin, the company’s co-founder android now became director of Google’s mobile platform.

The first android version was announced in 2007 and in parallel with the announcement of the android system, Google has created a consortium called “Open Handset Alliance”. It aims to promote the Android system and facilitate its integration into the different terminals that can accommodate it. Because Google does not build its own android certified mobile phones [11].

Unlike Apple, the search engine giant has indeed chosen to ally to different brands that have developed their own devices. Motorola, HTC, LG Electronics and Samsung are

the first manufacturers that have introduced Android smartphones. Today, four of the world's top five manufacturers of mobile phones have adopted this system.

Since October 21, 2008, the Internet search giant has made Android a free system operating the open source license. Therefore, the source code of the system(including network and telephony layer) was available on this date from all the developers of the planet. More precisely, the source code of the system operates a Apache license. Therefore, each developer or constructor can decide to use Android for commercial purposes. And thus save a budget not therefore by compared to similar offers from Microsoft (Windows Phone) or Nokia (Symbian).

1.7.3 Chronology

- Oct. 2003 Birth of Android (co-founder: Andy Rubin).
- August 2005 acquisition of the company by Google.
- Nov. 2007 Open Handset Alliance Foundation: Broadcom Corporation, Google, HTC, Intel, LG, Marvell, Motorola, Nvidia, Qualcomm, Samsung, Sprint, Nextel, T-Mobile, Texas Instruments.
- Nov. 2007 release of the first beta release.
- Sept. 2008 first stable version - first phone.
- Dec. 2009 14 new members (OHA).
- Feb. 2012: 300 million Android devices (850,000 phones activated by day)
- June 2012: 400 million Android devices.
- September 2012: 500 million Android devices.

1.7.4 Versions of ANDROID

Officially announced in November 2007, the Android system has come a long way since its inception. Since version 1.5, the versions have dessert names

Code name	Version number	Linux kernel version ^[1]	Initial release date	API level
No Codename	1.0	2.1	September 23, 2008	1
Petit Four	1.1	2.6	February 9, 2009	2
Cupcake	1.5	2.6.27	April 27, 2009	3
Donut	1.6	2.6.29	September 15, 2009	4
Eclair	2.0 – 2.1	2.6.29	October 26, 2009	5 – 7
Froyo	2.2 – 2.2.3	2.6.32	May 20, 2010	8
Gingerbread	2.3 – 2.3.7	2.6.35	December 6, 2010	9 – 10
Honeycomb	3.0 – 3.2.6	2.6.36	February 22, 2011	11 – 13
Ice Cream Sandwich	4.0 – 4.0.4	3.0.1	October 18, 2011	14 – 15
Jelly Bean	4.1 – 4.3.1	3.0.31 - 3.4.39	July 9, 2012	16 – 18
KitKat	4.4 – 4.4.4	3.10	October 31, 2013	19 – 20
Lollipop	5.0 – 5.1.1	3.16	November 12, 2014	21 – 22
Marshmallow	6.0 – 6.0.1	3.18	October 5, 2015	23
Nougat	7.0 – 7.1.2	4.4	August 22, 2016	24 – 25
Oreo	8.0 – 8.1	4.10	August 21, 2017	26 – 27
Pie	9.0	4.4.107, 4.9.84, and 4.14.42	August 6, 2018	28
Android Q	10.0			29

Legend: ■ Old version ■ Older version, still supported ■ Latest version ■ Latest preview v

Figure 1 3 – Evolution of Android versions

1.7.5 Architecture of Android

The Android platform is composed of different layers:

- **Applications**

Android comes with a set of apps including an email client, a SMS application, calendar, mapping service, browser. . . .

- **Applications Framework**

By providing an open development platform, Android offers developers the ability to create extremely rich and innovative applications. Developers have full access to the same Framework API used by the basic applications. The application architecture is designed to simplify the reuse of components; any application can publish its cited capabilities and any other application can then make use of these capabilities (subject to security constraints applied by the Framework).

- **Android Runtime**

Each Android app runs in its own process and with its own Dalvik Virtual Machine Instance (Dalvik VM). Dalvik VM was designed to optimize multiple execution of virtual machines. the Dalvik VM relies on the Linux kernel that handles functions such as threads or management low-level memory

- **Linux Kernel**

Android is based on the Linux kernel (2.6). so it benefits from these features stability and efficiency in terms of memory management, process management, model security, shared library support, etc. in addition to that the Android kernel is adapted to mobile by context-specific modules such as management energy [12].

1.8 Why did we choose ANDROID ?

In the previous sections, we studied the general knowledge of several platforms. We gave the market statistics and we saw the limitations of certain platforms. In the practical part, we chose the Android platform to develop our app. We cite a few reasons:

1.8.1 The system

Android is an open source platform based on linux. It powerful, modern and safe. Thanks to the open source code and APIs, the developers get permission to integrate, expand and replace existing components. Users can adapt applications to their needs.

1.8.2 Data management

With an Android phone one copies the files where one wants on the SD card (or internal memory) and everything is detected by the system. At Apple logic is different. The only way to put data on the mobile is to pass by iTunes.

1.8.3 Optimization for Google Services

Android has been developed since the beginning with the intention of integrating with Google (which is not right but good). Google services work perfectly on Android and thanks to it Android will benefit from the success of Google. The developers also can more easily integrate Google services into their applications.

1.8.4 The cost

Developing an app for Android will cost you absolutely nothing. All necessary tools are provided by Google and free of charge as well as documentation.

1.8.5 System position and rate of progress

Android dominates the global smartphone market and its users increase rapidly.

1.9 Conclusion

In this chapter, we gave a brief overview of mobile applications, as well as its different development strategies, and then we defined mobile operating systems. Then we made a presentation of android including its description, its architecture its advantages. In the end we made a short brief of flutter technologie wich we used in this project

Chapter II :

Games Applications

2 Games Application

Games have clung to daily activity and society, ranging from children to adults. The development of the games increasingly demonstrates their existence. At first, games are only played as entertainment, but now games have been able to switch their functions. For example, games can be used as educational tools or as media of regional or national competition.

The development of the game itself rapidly increases. Long time ago, game was only played on computer, but now it can be played as a mobile game. The mobile game itself is designed to be played on mobile device such as PDA, smartphone, PCs tablet or portable media player.

2.1 Definition

mobile game is a game that not only can be played via mobile phone and smartphone, but it can be developed in a wide variety mobile handsets such as PDAs, Symbian OS, Android OS, and Microsoft

2.2 History

The mobile game history began in 1997 when Nokia released a new type of game which called “Snake” that could be played by using mobile phones. In 1999, there was a new trend of mobile game which based on WAP, and there was also an SMS game. In 2001, with the development of the mobile phone specifications, the games could be played on the colored screen with a better graphic support and had been able to be downloaded. In 2003, Nokia introduced N-Gage and online game with the same platform. Until now this platform has supported 3D platform with the better quality and resolution that supports the programming platform.

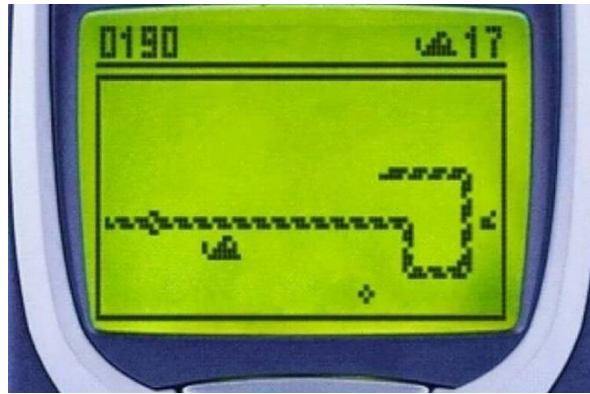


Figure 2 1 Nokia snake game

2.3 Types

More than a means of entertainment, mobile gaming has since become a way of life for many people from various age groups, surpassing its main target—the youngsters. Today, the advancement in mobile game development has brought an array of mobile game genres suitable for children and adults alike.

are the most popular mobile game types

- Action
- Adventure
- Arcade
- Battle Royale
- Casual
- Card
- Role-Playing
- MOBA – Multiplayer Online Battle Arena
- Puzzle
- Racing

- Strategy
- Sports
- Trivia
- Word

2.4 Mobile games advantages

the advantages of mobile game are:

2.4.1 Huge Potential of Consumers

At this time, the number of the populations in developed countries who have mobile phone is bigger rather than those who have computers at home. The potential of mobile phone enthusiasts is greater than the other platform potential market such as PlayStation and GameBoy. Nowadays, the number of mobile phones which are used is more than one billion, and it will increase as predicted.

2.4.2 Portability

GameBoy has sold more console game compared to the other because in the current era people prefer to play games that can be played any time. Portability is an important thing for people. A mobile phone is probably not a good tool for a game rather than modern console or computer, but people can bring it all the time

2.4.3 Network

Mobile phone is a network device which supports multiplayer game although there are still some deficiencies on it.

2.5 Mobile gaming applications as an educational platform

Education is a social and mutual outcome which is accomplished through conversations, spreading of knowledge and social networking. In the educational society, the knowledge is gained from teachers, educational materials, and the reactive contents which are disseminated among the community, huge information are integrated through negotiation, exchange, discussions, sharing, communication, so the members of communities reinforce their knowledge by sharing information and opinions between each other [13] .

Mobile learning comes into view as one of the best options to solve the educational challenges by using too many different resources and tools that are available, Mobile learning within schoolroom allow students working in correlating way in order to solve many problems, working on their projects and sharing opinions, also allowing them to access into available content anywhere and anytime ,many previous Studies show that tablets, iPod, iPads and other handheld devices are very communal devices to use it in mobile learning because of their availability in suitable cost .

Since mobile learning relies on the scale of social confession, many people in developing countries encourage the use of mobile technologies in education and make learning away from traditional schoolrooms. If mobile learning is effectively used in schools, it will allow academic staff and colleagues to communicate easily and send urgent information into each other in sufficient way and the most important thing that mobile learning will open up a revenue stream. Today, all students who are registered in educational foundations need frequently information from their foundations about course schedule, assignment deadlines, feedback from teachers and other important departmental details. So using of mobile technologies will accomplish all of these needs in more effective and efficient way [14].

Smart education is a creativity , innovation and customized systems for evolving new methods to learn by utilizing the new technology trends like using of cloud computing, and allow students

to learn with many materials depending on their qualifications and their intellectual levels by using mobile technologies at anytime and anywhere regardless of their locations.

Smart education describe the shift in students role as receivers into producers of information and knowledge and also it change the role of teachers from producers into learning assistants. To accomplish this, assignments, lectures and even the performance evaluations will be implemented by smart education technologies, smart education will make learning very interesting and will encourage students to do their homework's with high spirit . smart education support teaching and learning techniques by enhancing problem solving and increasing the experience of students' learning , smart education reinforce the flexibility of learning systems and simplify learning in relationship with personal absorption and future career ambition, It also helps educational institutes to promote from delivering the knowledge into supporting personal education according to students' qualification levels , even more smart education come up with free access into valuable contents which is developed by private and public organizations and individuals , it also extend the cooperative use of local and overseas resources ,so building a smart educational environment is very important to encourage students to learn and provide them with well-tailored methods that are suitable to their interests and their intellectual levels [15].

Mobile learning is not provided to be a unique educational paradigm, but to supplement the other fields of our studies such as lecture scenarios or e-learning. Using Smartphone in our life is growing dramatically, even among universities and high school students. Having mobile application that helps students to exam their self or launching mobile learning app becomes essential educational need. Nowadays, Smart phones are the most common platform for m-learning, there are many mobile learning apps in the literature, in this project we try to develop self educational mobile app for university students. Our app will provide game educational environment, where each students play a role as fighter who strive to collect as much as points to go the next level. We believe in that the mobile app will be more interesting and attractive if it has been lunched as game app. The app will allow students to view their questions/answers and who answered their questions. This kind of application will serve as a complement to our original educational environment.

2.6 Conclusion

In this chapter, we gave a brief overview of mobile game applications, as well as its different concepts from the definition to history and genres, then we discussed the benefits of using this type of application as an educational tool to deliver the informations to the user in effective as well as interesting way

Chapter III :

Conception And

Realization

3 Conception and realization

3.1 Conception

3.1.1 Introduction

Analysis and design are the two fundamental steps in the development process- In this chapter, therefore, we begin by drawing up the diagrams of the participating classes, which are UML class diagrams which describe the case-by-case use of the three types of analysis classes, the dialogs, controls and entities and their relationships, these diagrams are grouped together to obtain the global class diagram. Also we present the navigation diagrams that allow to model the navigation between the different links of the IHMs.

3.1.2 UML

3.1.2.1 definition

UML is defined as a graphical and textual modeling language for understanding and describing needs, specifying and documenting systems, sketching software architectures, designing solutions and communicating views [16].

3.1.2.2 Types of UML Charts

UML offers thirteen types of diagrams to model a system. They are divided into two main groups [16] :

1. Structural or Static Diagrams

- **Class diagram :**

A class diagram is a collection of static modeling elements that shows the structure of a model (classes, associations, interfaces, attributes, operations, etc.).

- **Object diagram :**

Allows you to model the instances of structural elements and their links to execution.

- **Package diagram :**

represents the packages that make up a system, as well as the relationships between packages.

- **Component diagram :**

These diagrams describe the components and their dependencies in the realization environment.

- **Deployment diagram :**

Deployment diagrams show the physical layout of the different hardware.

- **Composite structure diagram :**

It shows the internal organization of a complex static element.

2. Behavioural diagrams or dynamic diagrams

- **Use Case Diagram :**

The use cases describe the needs of users and the corresponding objectives of a system.

- **Sequence diagram :**

The aim is to describe how the elements of the system interact with each other and with the actors.

- **Communication diagram :**

formerly called a collaboration diagram is an interaction diagram. Its main purpose is to describe the interactions between the different objects of the system.

- **Activity diagram :**

It shows the sequence of actions and decisions within an activity.

- **State machine diagram :**

It shows the different states and possible transitions of the objects of a class.

- **Overview diagram of interactions :**

It fuses action and sequence diagrams to combine interaction fragments with decisions and flows.

- **Time diagram :**

It merges state and sequence diagrams to show the evolution of the state of an object over time.

3.1.3 UML diagrams

Class diagram

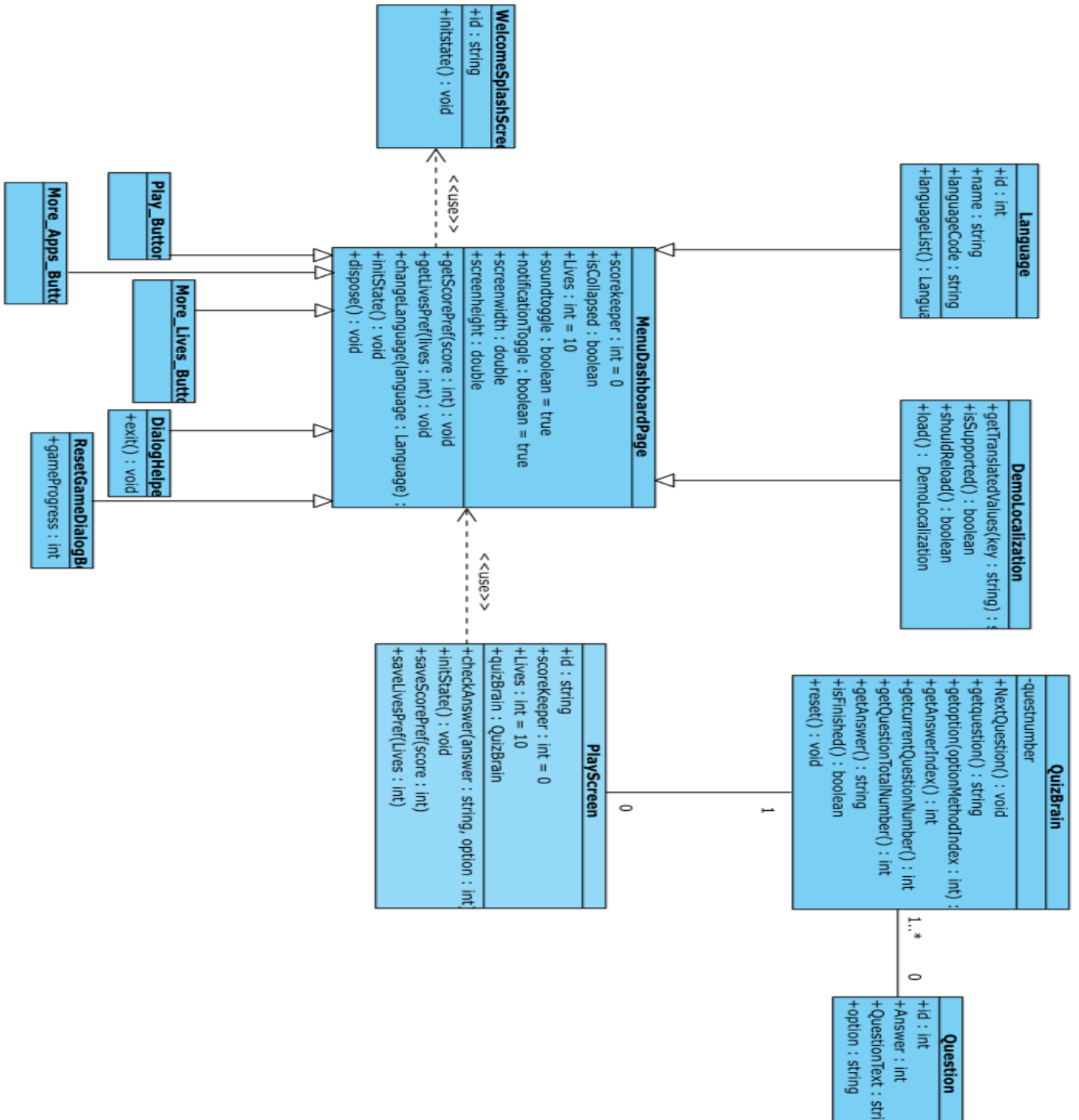


Figure 3-1 Class Diagram of the system

Use case diagram

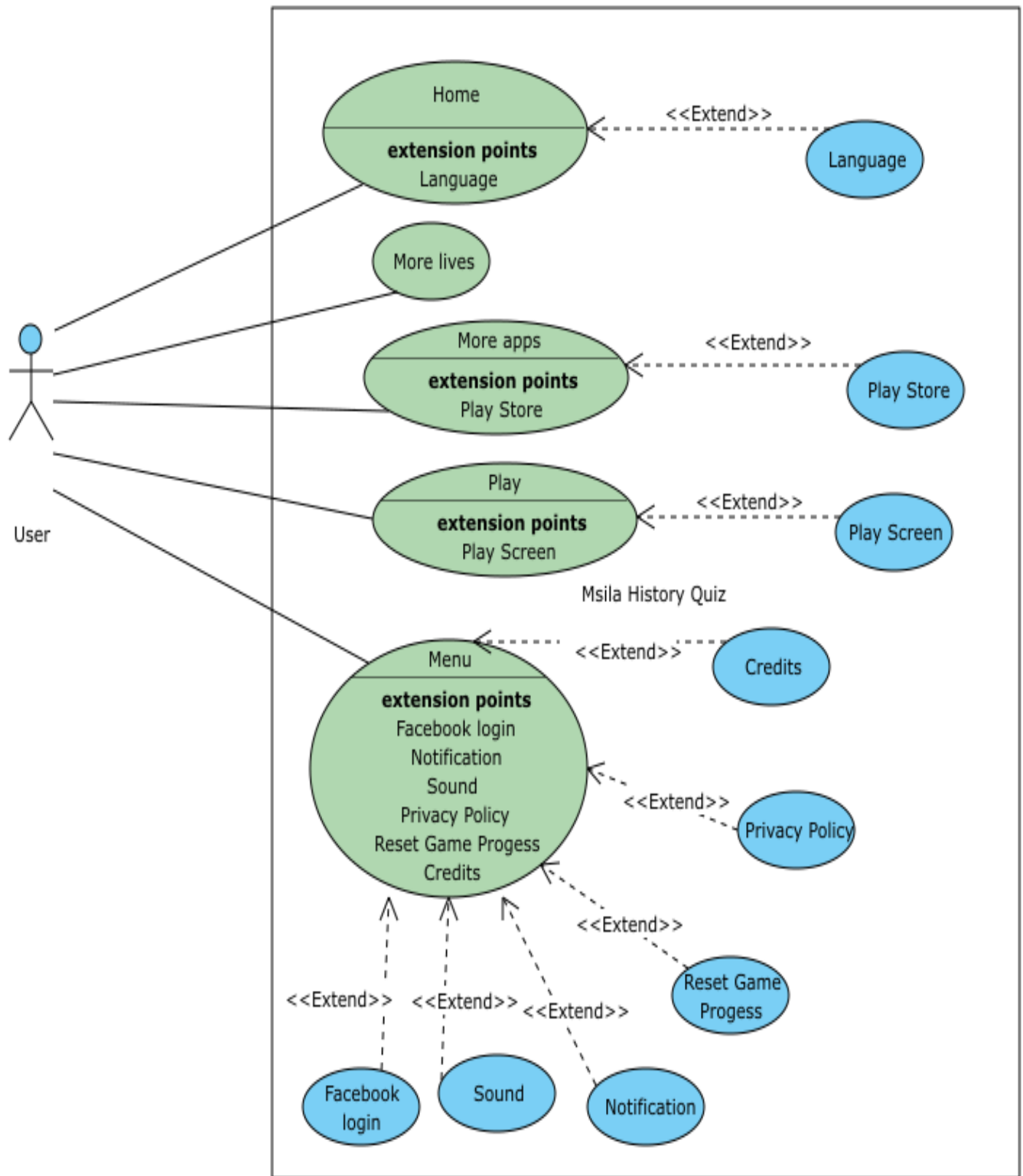


Figure 3-2 Use case diagram of the system

Sequence diagram << facebook login >>

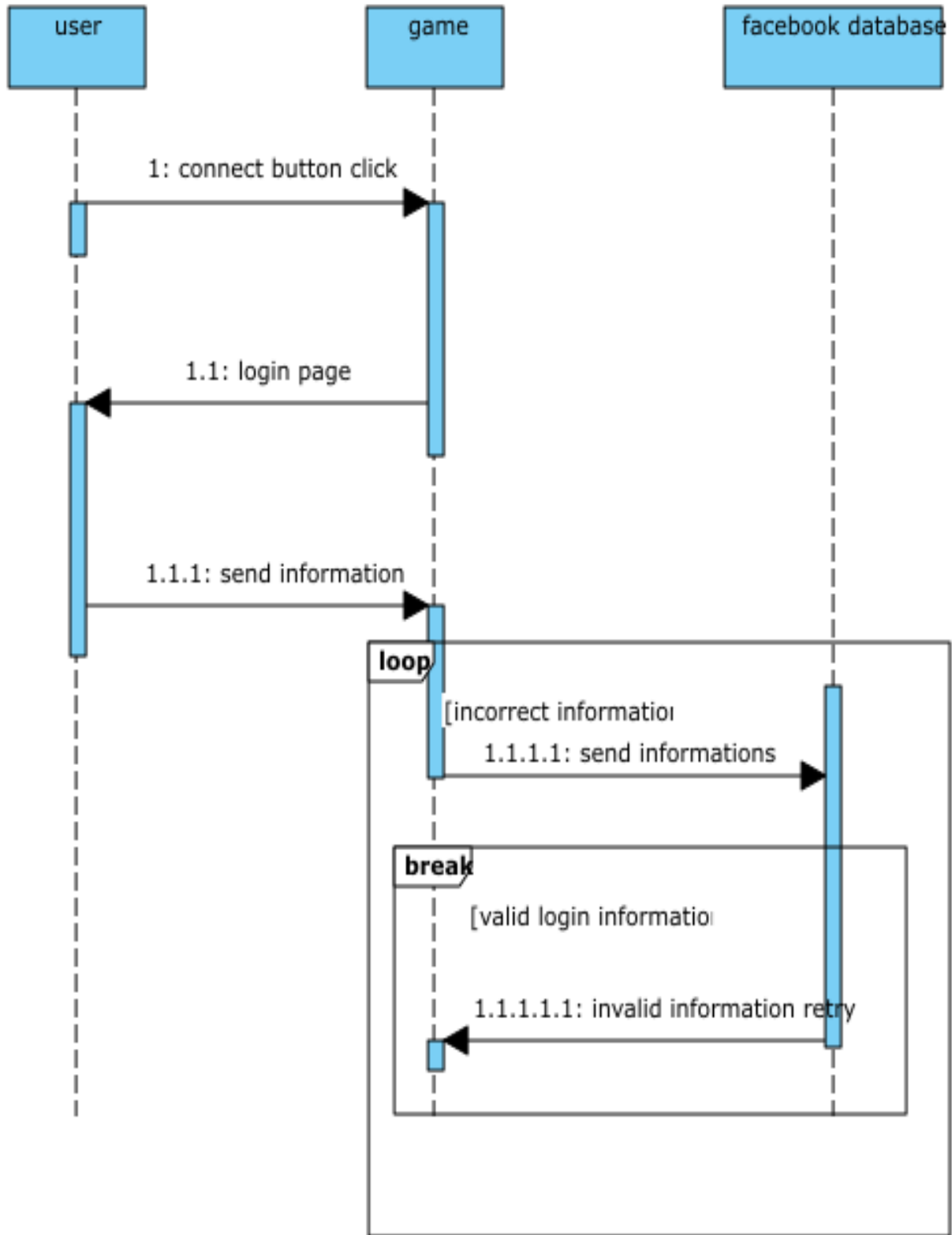


Figure 3-3 Sequence Diagram << facebook login >>

Sequence diagram << game play >>

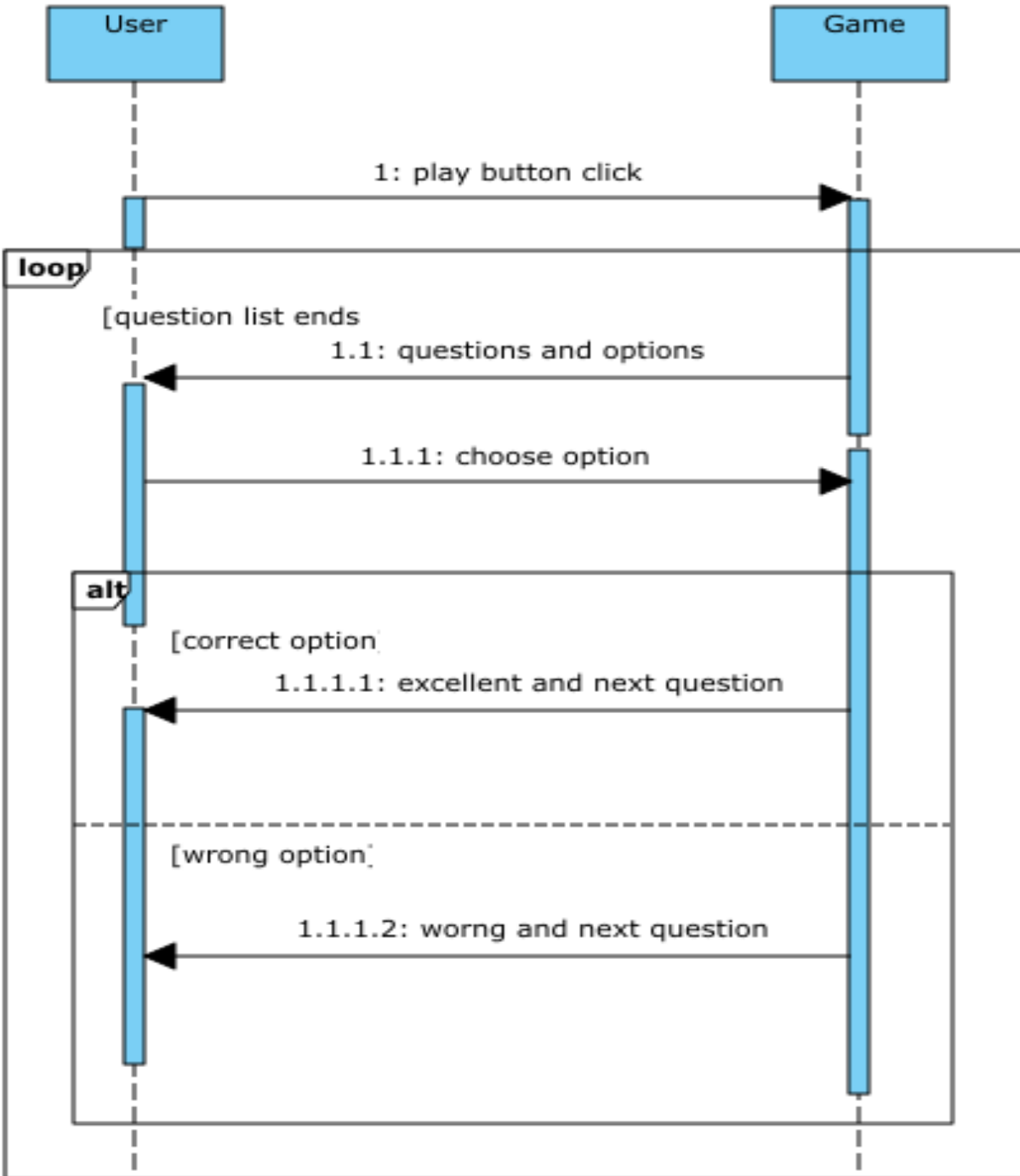


Figure 3-4 Sequence diagram << game play >>

Activity diagram << game play >>

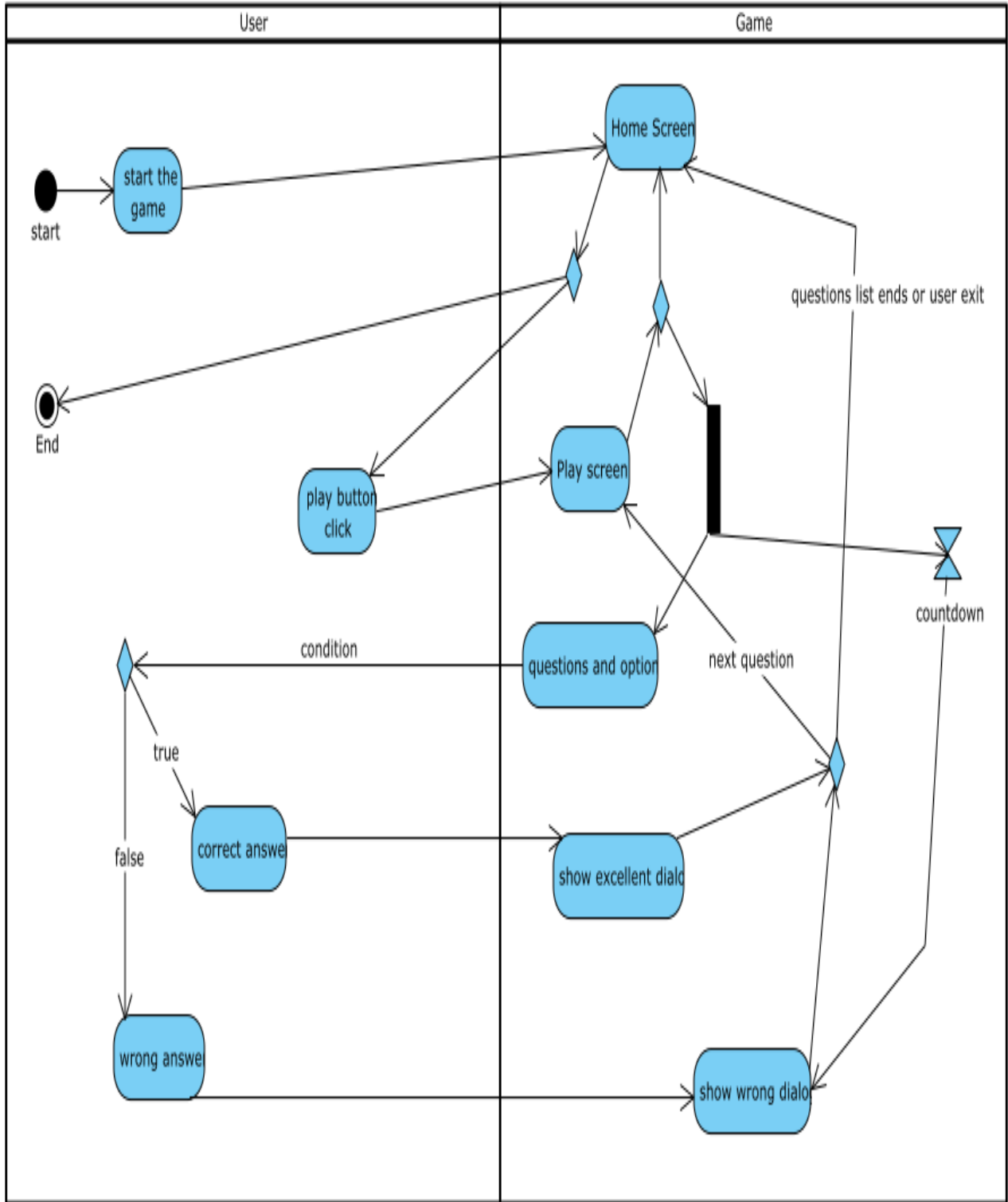


Figure 3-5 Activity diagram << game play >>

3.1.4 IHM presentation

Our application consists of three main Screens . wich are the splash screen and home screen then play screen. In this part, we will present theis IHMs as weel as a brief explanation .

Splash Screen

the splash screen is the first screen that user interacte with. This screen is used to load the application assets with a nice animations .

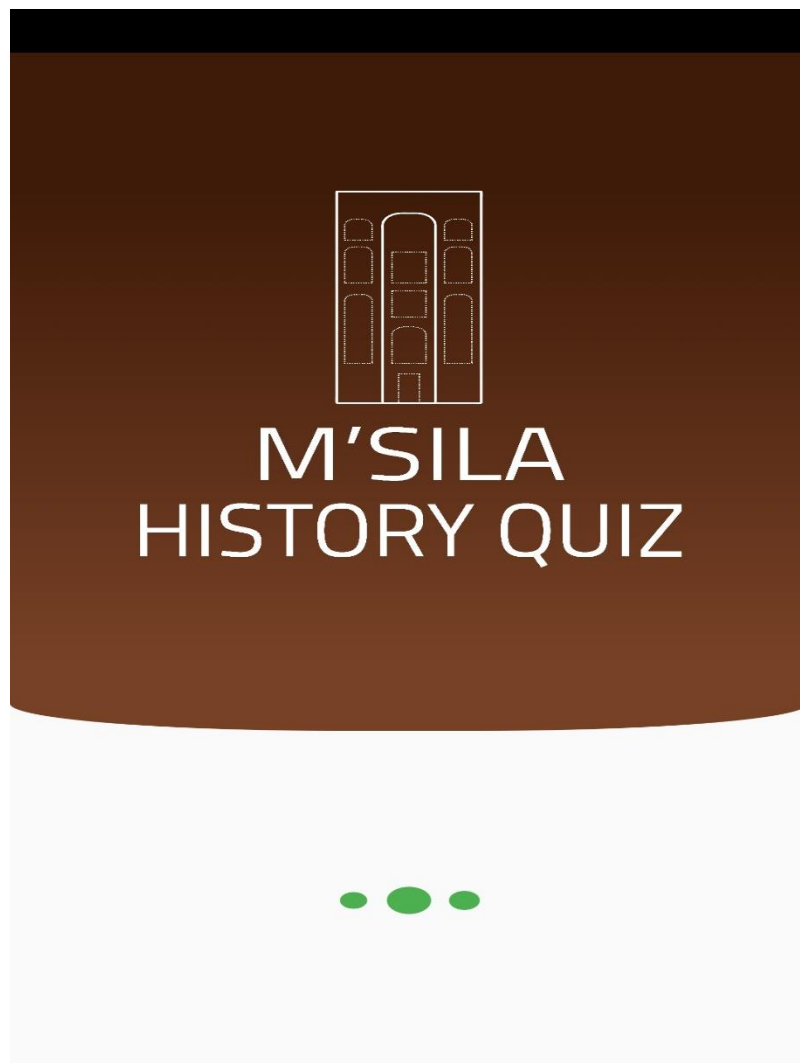


Figure 3-6 splash screen

Home Screen

Home screen is mainly the most important screen in the application because its represente all the game features and the different informations that the user needs to understande the game asa well as interactions buttons . our home screen is devided into two parts . Menu and Dashboard

- **Menu**

In the menu we have various buttons like more apps button , more lives and button to change the language as well as lives indicator and progress indicator but the most important button is play button wich leads to the play screen

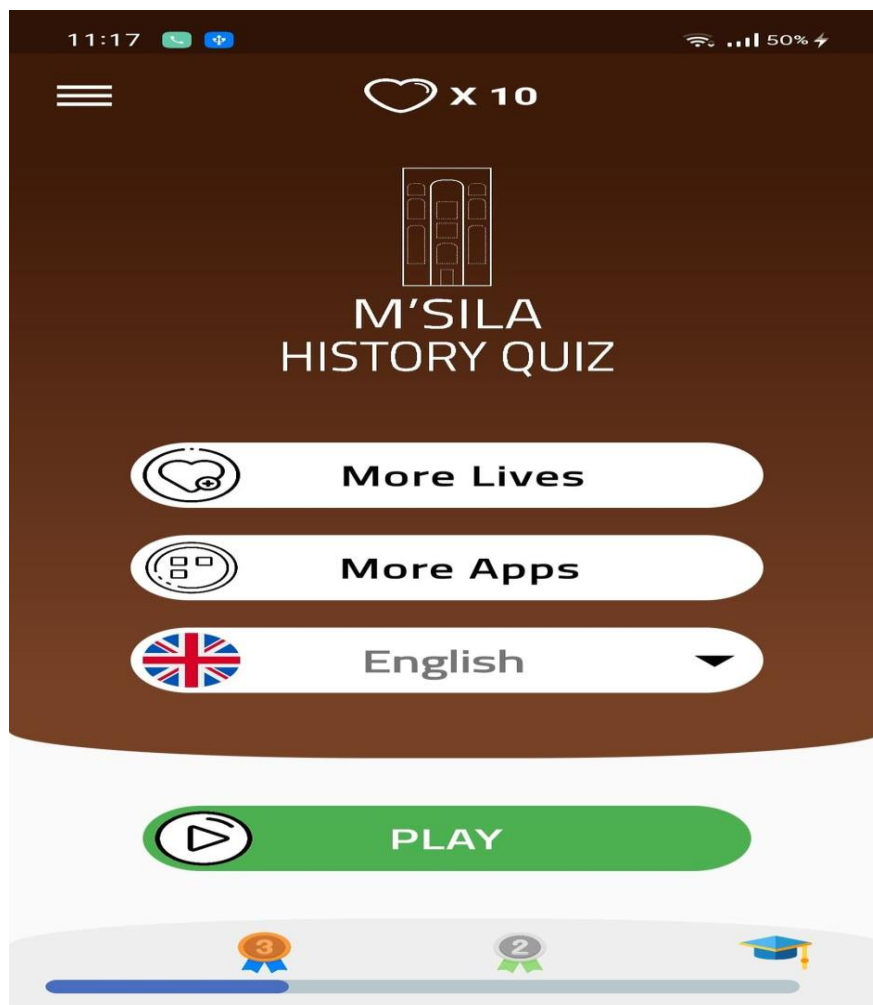


Figure 3-7 Home screen << English >>



Figure 3-8 Home screen << Arabic >>

- **Dashboard**

In the dashboard we have the facebook login button , sound toggle and notification toggle next we have the privacy policy button , the credits button is for the questions refrences and finally we have reset game progress button wich reset the user progress from the beginning .

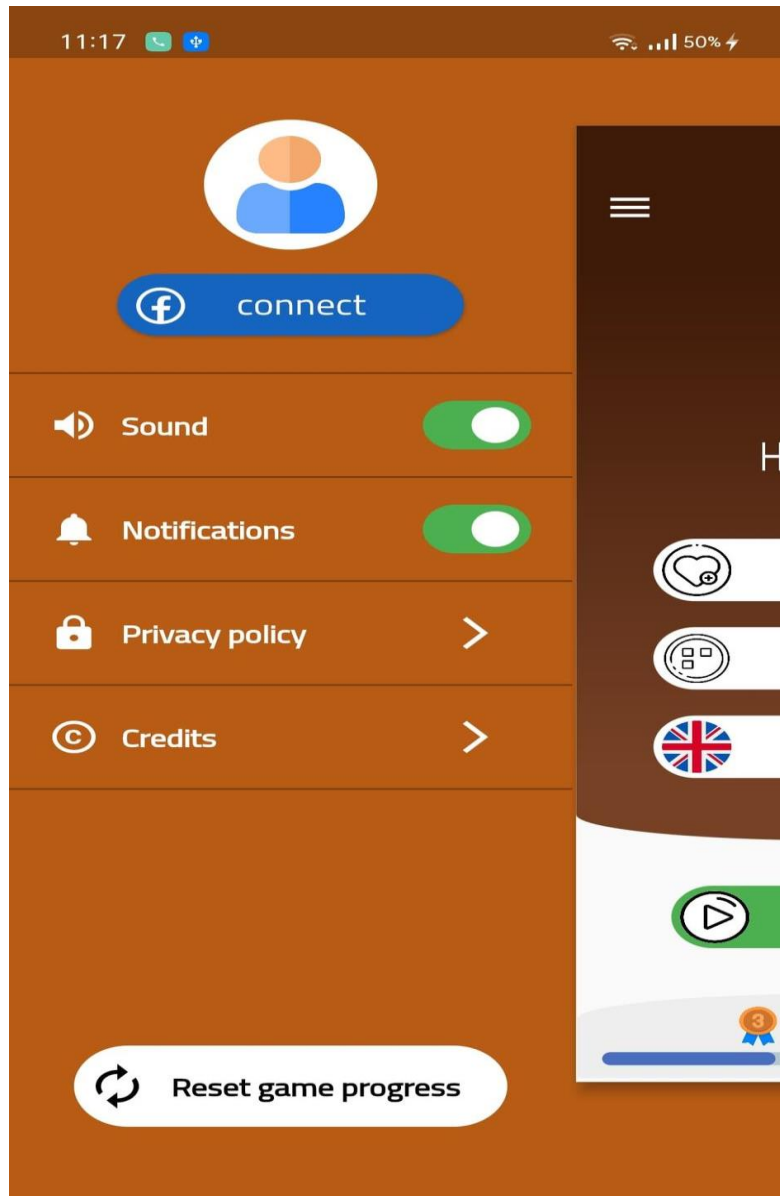


Figure 3-9 Dashboard << English >>

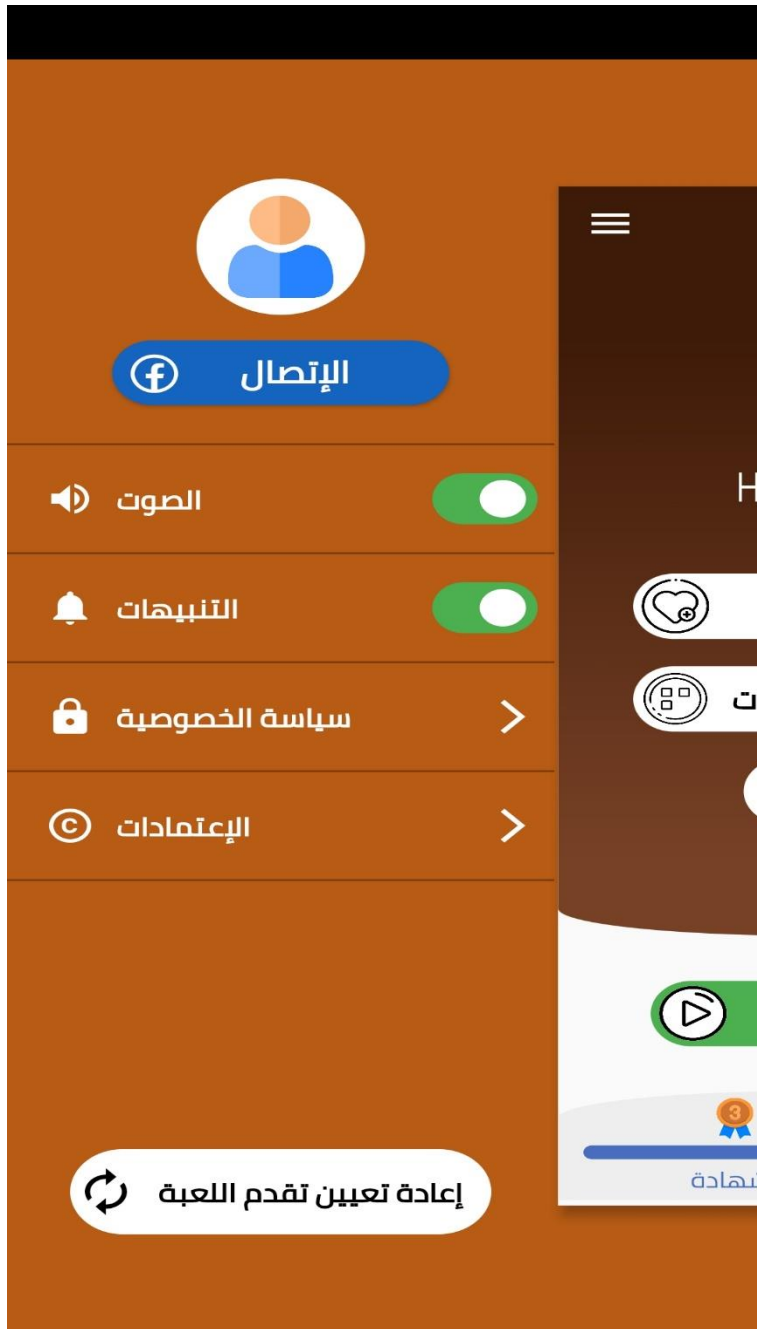


Figure 3-10 Dashboard << Arabic >>

Play Screen

The play screen is basically where the user play the game it contains a countdown timer , the question and four options . wich the user have to choose between them . a dialog will appear when the user answers the question Then the game take him to the next question

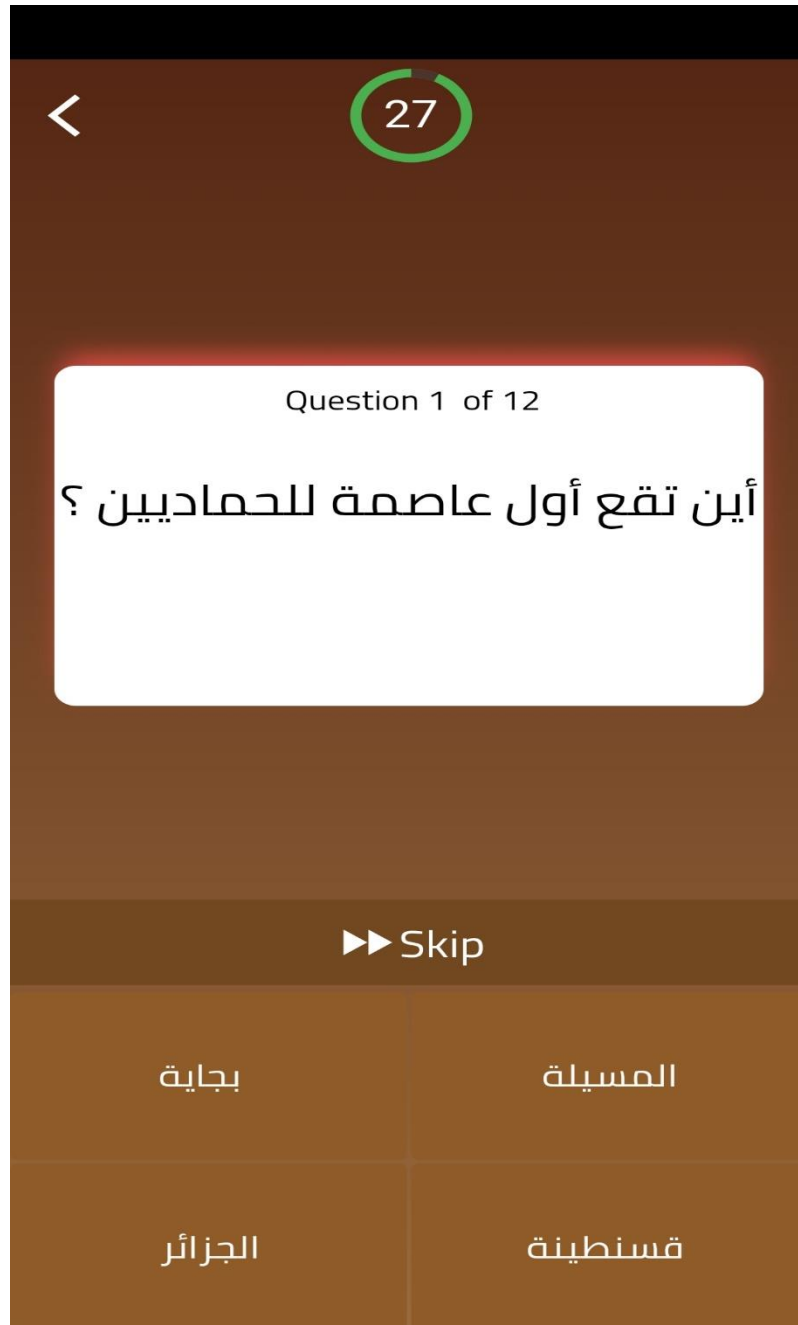


Figure 3-11 Play Screen



Figure 3-12 correct answer

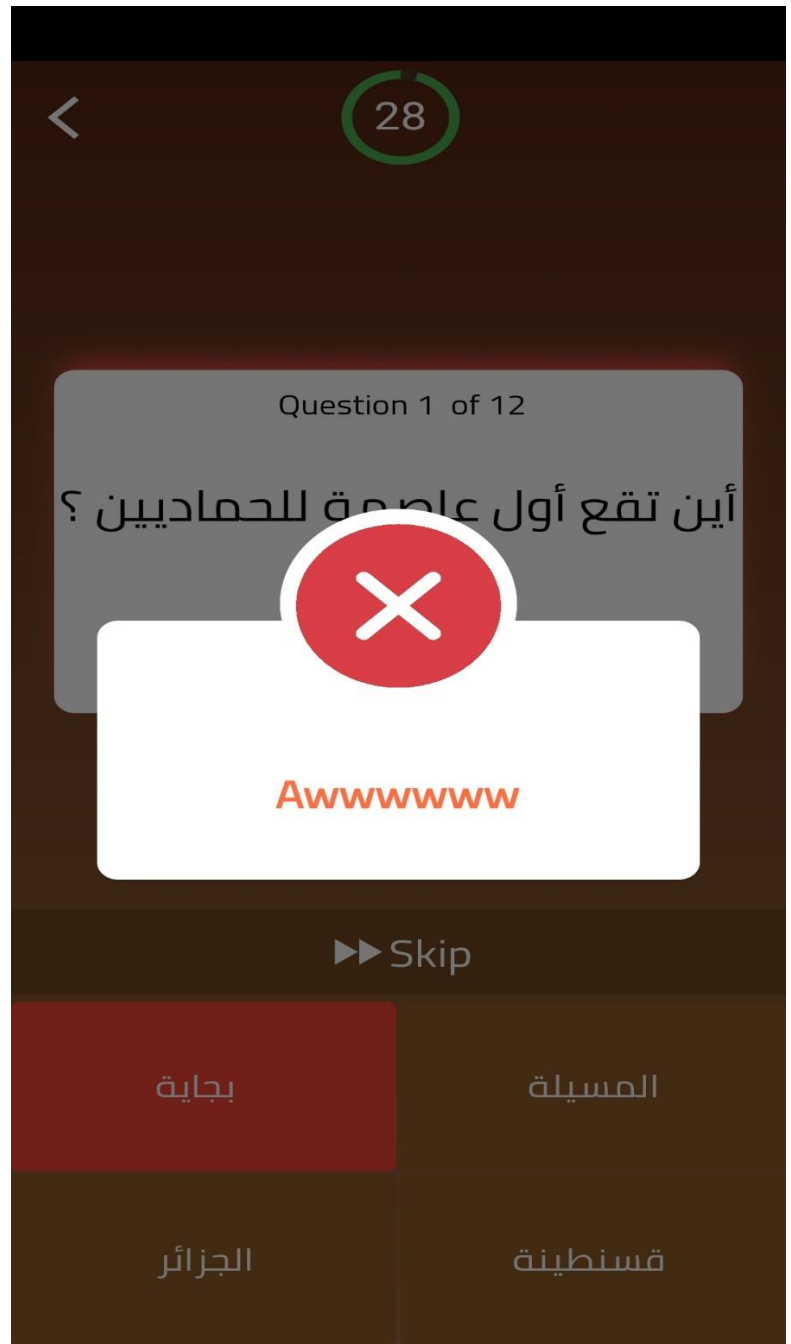


Figure 3-13 incorrect answer

3.2 Realization

3.2.1 Introduction

This last part dedicated to the realization of our application. The aim of this part is to explain the different technical choices : languages, environments and development tools used, as well as the chosen development platforms.

3.2.2 Technical Choices

3.2.2.1 Flutter [17]

- **Whats is flutter ?**

Flutter was introduced by Google as an open-source technology for coding and creating native apps for Android and iOS. Flutter is relatively new as it was officially presented in December 2018 as the first stable version 1.0 at the Flutter Live event.

Flutter combines ease of development with performance similar to native performance while maintaining visual consistency between platforms. Flutter's programming language, Dart, was initially intended as a replacement for JavaScript. Most importantly, Flutter is open-source and completely free. At the moment, Flutter has equal popularity with React Native on both GitHub and Stack Overflow. Speaking of which, we contrasted Flutter's and React Native's pros and cons using nine criteria in a recent article.

Google uses Flutter for various Google Assistant modules and the Google Home Hub user interface. Moreover, there are already 50,000 Flutter apps available in the Google Play Store, and this number is increasing at a high rate. Alibaba Group, eBay, Groupon, and other popular e-commerce providers use Flutter as well to give their web and mobile applications uniform looks.

Tim Sneath, Group Product Manager at Google, says, "Our vision for Flutter is something that many of us have been dreaming of for years — a powerful, general-purpose, open UI

toolkit for building stunning experiences on any device-embedded, mobile, desktop or beyond”.

- **How flutter work**

The framework of Flutter, written in the Dart programming language, has the Flutter engine, Foundation library, and widgets. The approach to development in Flutter differs from others by its declarative UI writing. Here, there is a need to start from the end, meaning before starting the development of some element, the user needs to have in mind a complete picture of what kind of UI it will be. Many developers distinguish this UI writing as a more clear one, but it also causes certain difficulties for developers at first.

The main idea of Flutter is that developers can build the entire user interface by simply combining different widgets. The application interface consists of various nested widgets, which can be any object. This applies to anything from buttons to padding, and by combining widgets, the developer can customize the application radically. Widgets can influence each other and use built-in functions to respond to external changes in the state. Widgets are important elements of the user interface and comply with the design specifications of Android, iOS and conventional web applications.

With Flutter, developers can create custom widgets, which can be easily combined with existing ones. Note that there are no OEM widgets, but [Flutter Gallery](#) provides developers with their own ready-made widgets — a set of application examples that show how to use standard widgets — which look like native Android and iOS design languages (Material and Cupertino).

Flutter also provides developers the ability to view widgets in a reactive style. For the record, Flutter isn't the first one to do this but Flutter is the only mobile SDK that offers a reactive look without the need for a JavaScript bridge. Moreover, Dart comes with a repository of software packages to enhance the capabilities of applications. For example, it offers several packages that help to access Firebase so that developers can create

serverless applications. Another package allows access to the Redux data warehouse or facilitates access to platform services and equipment such as cameras.

- **Benefits of Flutter**

When it comes to developing applications, developers are not restricted to a single cross-platform mobile framework. In fact, statistics show that while the majority of developers use React Native (42%), Flutter usage in 2020 (39%) has highly grown in comparison with Flutter usage in 2019 (30%).

Let's take a closer look at explaining why Flutter application development can be a better choice.

- A. **Quick code development**

Flutter creators wanted to invent a technology with the quickest opportunity to deliver a great-performing, cross-platform mobile application. The following features allow this:

- **Hot reload.** Flutter's hot reloading helps save time while developing by letting the developer see the applied changes in real-time. This capability helps developers be significantly more efficient and productive. Flutter's hot reload works better than competitors' similar features. It allows the developer to pause code execution, make changes to the code, and continue the code from the same place. This greatly speeds up development and allows more experimenting.
- **Widgets.** One of the most significant benefits of Flutter is how it uses ready-made widgets. This ensures that Flutter offers a consistent model for development and design. Widgets are Google-based, so they are high code quality and perform better than other open-source frameworks. As most of them are extremely customizable, they save developers' time like no other framework.

In addition to the major layout widgets, Flutter widgets follow both the Material and Cupertino looks, which is a huge advantage.

- **Minimal code and access to native features.** Flutter lets developers use Dart, which is compiled directly into the ARM code of mobile devices and helps not only to speed up applications, it allows them to launch quicker also. Flutter helps make developers' lives easier through its third-party integrations and native codebase. With Flutter, developers can access native features.

B. Great UI

Will Larche, Software Engineer at Google, says, "Flutter's architecture is designed for building beautiful, custom UI. Flutter's main goal is to make building polished, custom app interfaces a faster, more delightful experience for designers and developers. Flutter is powerful enough to draw anything designers dream up."

- **Beautiful, custom design.** The powerful thing of Flutter is Skia, the open-source, high-performance graphics engine used by Adobe, Chrome, and Amazon Kindle. Flutter allows users to develop applications with custom designs, which will look equally good on iOS and Android devices. Applications developed on Flutter unlike its competitors have no risk that there will UI failures when updating the software.
- **Same business logic and UI.** Possible for Flutter sharing the UI and business logic on Android and Apple devices allow developers to achieve a seamless experience regardless of the OS. This is primarily important for brands with a unique and outstanding corporate style.

Flutter doesn't need any platform-specific UI components to render its UI.

The one and only thing Flutter needs to show the application UI is a canvas to draw on

- **UI customization potential.** A huge advantage of Flutter is the ability to customize everything you see on screen, regardless of the complexity of the element. The amount of effort required is fundamentally lower from that required on native platforms' development software.

- **Why we Choose Flutter on our Project**

There are things that are of great value to the business — platform stability, its performance, a guarantee of successful support, and improvement of technologies and products. Any problems and shortcomings in any of these aspects may lead to risks, including direct and indirect financial losses.

With this in mind, Flutter lowers risks for your business due to the following points:

- A. The best in class (cross-platform) performance and resource consumption** due to the compilation of native code and high-performance rendering engine. The first provides an easy way of establishing communication between platform-native code and Dart through platform channels. Thus, developers can implement into a Flutter app anything that a native app can do, just with a little more effort on the native side. Because of the engine (Flutter uses Skia for rendering itself), a UI built in Flutter can be launched on virtually any platform, assuming this platform supports Flutter. Putting it differently, developers no longer have to adjust the UI to transfer it to a platform, which simplifies the development process vastly.

A good UX is incredibly dependent on the performance of the application. Flutter application performance in most cases will be indistinguishable from the native application. This is because Flutter doesn't rely on intermediate code

representations or interpretation. Application on Flutter is compiled directly into machine code, eliminating any performance errors in the interpretation process. This provides it with the highest performance and makes it the most resource saving software among cross-platform technologies.

- B. Better developer productivity** is achieved due to Flutter being primarily designed for quicker code writing. It consists of ready-to-use widgets, its syntax requires less code to be written, and hot reload speeds increase the searching for and correction of bugs. All this results in fewer man-hours for developers. Also, finding Flutter engineers in 2020 is not a problem the community of fans is growing, especially among Android developers. According to Stack Overflow Developer Survey 2020, 68.8% of developers which is among the top 3 — love to use Flutter and have expressed interest in continuing to develop with it.
- C. Quicker time to market.** Due to the greater productivity of Flutter developers, it takes less time to create an application, which means that compared to other programming languages and frameworks, applications in Flutter are written quicker and enter the market earlier with equal effort. Thus, the less coding and support effort needed, the quicker the time to market.
- D. Low-cost app development.** Flutter provides more efficient development work and, accordingly, to develop an application requires less man-hours. At the same time, the cost of an hour is at the average market (and sometimes even lower) level. As a result, the cost of the application on Flutter is lower than when using other cross-platform languages or native development.

- **Flutter's Downsides**

While Flutter has a lot of benefits that businesses can take advantage of, there are some areas in which it still needs work.

- **Lack of third-party libraries.** Flutter is a newer technology. As such, the volume of third-party libraries currently available for Flutter is limited. Third-party libraries help speed up development time significantly, so this is a definite downside to developing in Flutter.
- **Large file size.** Many, if not most, of the apps developed through Flutter are destined for mobile devices only. Although current mobile devices have large storage capacities, file size is still important. For example, the creation of a hello world app in Flutter could account for 4.7MB to 6.7MB. The same app created in native Java is closer to 500KB.
- **New skills required.** While Flutter is easy to use and can be learned by non-programmers, it does require developers to learn Dart first. This adds an additional phase of learning, which can increase the time and money for any project. That being said, if a developer knows Java/C#, he or she can easily upskill to Dart. Moreover, Flutter's Dart programming language is pretty easy to learn for those with little programming experience.

- **Apps Built with Flutter**

Although Flutter is young enough, it is already used by such global services as Google Ads, Alibaba, AppTree, Reflectly and many others.

- **Flutter's Popularity and Perspective**

Flutter's popularity is easy to understand, and while half a million developers use Flutter monthly, the top five territories for Flutter developers are the US, China, India, Brazil, and the EU.

“You see today, how Flutter is already enabling developers all over the world to deliver beautiful apps to hundreds of millions of people throughout the globe. But this is also just the beginning.” says Eric Seidel, Engineering Manager at Google. “Flutter is

highly portable and already works on many form factors beyond phones... So if I look forward a few years, I see Flutter running in a ton more places.”

A technical preview of Flutter Web was introduced by Google, which allows developers to run Flutter applications in a browser in a clean form without changing the source code. It means that with Flutter, developers can 100% go further beyond app development on mobile. Moreover, this marks Flutter’s transition from a cross-platform mobile application framework to a full-blown cross-platform development tool.

Despite the fact that everything, except the mobile part of the framework, is not yet considered ready for production, an experienced Flutter developer can make a pure flutter application, running today on any major platform — including Android, iOS, Web browser, Windows, macOS, Linux and even embedded devices — and the application will function properly without changing and customizing the code for each platform separately.

- **Flutter’s 2020 Major Updates**

While Flutter is new, its creators are not resting on their laurels. They are continually innovating and improving Flutter to make it a more useful and powerful tool for developers. Some key improvements that are worth calling out in 2020 are as follows:

- **Branching model** – Flutter will be changing the way they release versions. From April onwards, they will create a new branch at the beginning of each month as a beta release. During that month and the months following, they will work to stabilize this release and will promote it to the newest version quarterly.
- **Alignment with Dart** – In a similar fashion, the Dart release process has also been changed. Now Dart offers a beta channel too, and the release discrepancy between Flutter and Dart will be aligned.

- **Adobe XD support for Flutter** — Adobe XD to Flutter plugin is now available for public testing with early access. “The ability to export designs to Flutter further reduces the latency between creative ideas and product development, as an XD-prototype can now become working Flutter code within minutes. Adobe XD supports design on Windows or macOS, and includes a free starter plan to get you up and running,” announced Tim Sneath.
- **Flutter Windows Alpha** — Adding Windows to Flutter, with support for Windows 7 and above, gives adventurous developers something to get started with. This alpha release offers a solid foundation that must be stabilized over the coming month
- **Sound null safety** — When a developer chooses null safety, the types in code cannot be null by default, which means that values cannot be null unless the developer says they can be null. With this update, the runtime null-dereference errors become edit-time analysis errors.
- **Mobile Autofill Support** — Text autofill support in Flutter apps. With Flutter 1.20, Flutter has added the basic autofill functionality so there is no need to re-enter data already gathered by OS.

3.2.2.2 Android SDK

A SDK is a set of tools for creating apps. Android SDK is an indispensable tool for developing Android apps. The latter was developed by Google, however it requires the installation of the Java Development Kit (JDK).

3.2.2.3 Android studio

is the official Integrated Environment (IDE) for the development of Android applications. As Google’s official embedded development environment, Android Studio includes everything the developer needs to develop an

application: smart code editor and debug, performance analysis tools, emulators, and more [18].



Figure 3-14 Android studio Logo

3.2.2.4 Android AVD

An Android Virtual Device (AVD) is a configuration that defines the characteristics of an Android phone, tablet, Wear operating system, Android TV, or automotive operating system that you want to simulate in the Android emulator. AVD Manager is an interface that you can launch from Android Studio that helps you create and manage AVD [18].

3.2.2.5 Dart

Dart is a client-optimized language for developing fast apps on any platform. Its goal is to offer the most productive programming language for multi-platform development, paired with a flexible execution runtime platform for app frameworks [19].



Figure 3-15 Dart Language Logo

3.2.3 difficulties encountered

3.2.3.1 The choice of emulator :

for running our app we used the Android Virtual device, but it never helped us because it was very heavy and takes a lot of time to run, and then we tried to use another emulator that was Memu but unfortunately we encountered the same problem, so we used the smart phone to avoid this problem.

3.3 Conclusion

In this chapter, we gave the general idea of our conception and realization for this project , we divided this chapter into two parts , the first for the conception by showing the UML presentation of our project . and the second part is dedicated for the realization and we talked about our technical choices as well as some difficulties we encountered in this project .

4 General conclusion

Educational games are games explicitly designed with educational purposes, or which have incidental or secondary educational value. All types of games may be used in an educational environment, however educational games are games that are designed to help people learn about certain subjects, expand concepts, reinforce development, understand a historical event or culture, or assist them in learning a skill as they play. Game types include board, card, and video games.

In this application we tried to deliver a helpful gaming environment as well as educational, by providing various information about our state Msila both historically and geographically

The work we have done has allowed us to improve and enrich our skills in mobile, so we have gained experience with a set of languages, methods and technologies like flutter, adobe, dart

As perspectives, we intend to complete the implementation of certain functions such as managing ads.

Bibliography

- [1] M. Qiu , W. Dai , K. Gai, Mobile Applications Development with Android, Chapman and Hall/CRC, 2016.
- [2] D. M. Mahmud and N. A. S. Abdullah, "Mobile application development feasibility studies : A case study in universiti teknologi mara," in *IEEE Conference on Open Systems (ICOS)*, 2014.
- [3] Y. Park, A pedagogical framework for mobile learning : Categorizing educational applications of mobile technologies into four types, vol. 12, 2011, p. 78–102.
- [4] R. H, Introduction aux systèmes d’exploitation mobiles, 2013.
- [5] W.-M. Lee, "Beginning android 4 application Development," 2012.
- [6] "smartphone-windows-phone," [Online]. Available: <https://www.futura-sciences.com/tech/definitions/smartphone-windows-phone-15584/>.
- [7] M. H. Miraz, "Development platforms : Android, ios, windows phone, blackberry and symbian,," *Computer Barta*, vol. 17, p. 110–112, 2014.
- [8] "Blackberry os - definition.," [Online]. Available: <https://www.gsmarena.com/glossary.php3?term=bb-os..>
- [9] "Distribution of smartphone shipments worldwide by operating system," [Online]. Available: <https://www.statista.com/>.
- [10] "Taktıl communication .Application mobile Vs site mobile," [Online]. Available: <https://www.taktılcommunication.com/blog/applications-mobile/site-mobile-vs-application-mobile-avantages-et-inconvenients.html>.
- [11] L. P. Cyril Mottier, Développez pour Android, 2011.
- [12] "Phone android .Toute l’histoire et la chronologie d’Android," [Online]. Available: <https://www.phonandroid.com/toute-l-histoire-et-la-Bibliographie>.
- [13] Wu, Ting-Ting, Using smart mobile devices in social-network-based health education practice: A learning behavior analysis, *Nurse education today*, 2014, pp. 958-963.
- [14] Mehdipour, Yousef, and Hamideh Zerehkafi, "Mobile learning for education: Benefits and challenges," *International Journal of Computational Engineering Research*, pp. 93-101, 2013.
- [15] Jang, Sanghyun, "Study on service models of digital textbooks in cloud computing environment for SMART education.," *nternational Journal of u-and e-Service, Science and Technology*, vol. 7.1, pp. 73-82, 2014.
- [16] P. Roques, UML2-Modéliser une application Web, EYROLLES, 2008.
- [17] [Online]. Available: <https://nix-united.com/blog/the-pros-and-cons-of-flutter-in-mobile-application-development/>.
- [18] G. Developers, "Android studio.," [Online]. Available: <https://developer.android.com..>
- [19] G. developers, "dart overview," [Online]. Available: <https://dart.dev/overview>.