**Graduation Thesis**

**Online Learning Application Design**

**Do Tuan Kiet**

xxxx@gmail.com

**Ngành Điện Tử Viễn Thông**

**Chuyên ngành Kỹ Thuật Máy Tính**

|  |  |
| --- | --- |
| **Giảng viên hướng dẫn:** | ThS. Vũ Song Tùng  Chữ ký của GVHD |
| **Bộ môn:** |  |
| **Viện:** | Điện Tử Viễn Thông |
| **HÀ NỘI, 6/2022** | |

**Graduation Topic**

Biểu mẫu của Đề tài/khóa luận tốt nghiệp theo qui định của viện, tuy nhiên cần đảm bảo giáo viên giao đề tài ký và ghi rõ họ và tên.

Trường hợp có 2 giáo viên hướng dẫn thì sẽ cùng ký tên.

Giáo viên hướng dẫn

Ký và ghi rõ họ tên

CONTENTS

[**CHAPTER 1. OVERVIEW OF GRADUATION TOPIC** 7](#_Toc104580246)

[1.1 Problem 7](#_Toc104580247)

[1.2 Internet 7](#_Toc104580248)

[1.2.1 What is internet? 7](#_Toc104580249)

[1.2.2 How does the internet work? 8](#_Toc104580250)

[1.2.3 Benefit 8](#_Toc104580251)

[1.3 Introduce about application 9](#_Toc104580252)

[1.4 Summary 9](#_Toc104580253)

[**CHAPTER 2. SOLUTION ORIENTED TO DEVELOP APPLICATION** 10](#_Toc104580254)

[2.1 Problem 10](#_Toc104580255)

[2.2 Solution 10](#_Toc104580256)

[2.3 Introduce about IDE 11](#_Toc104580257)

[2.3.1 Visual Studio 11](#_Toc104580258)

[2.3.2 Xamarin 12](#_Toc104580259)

[2.3.3 Model MVVM 18](#_Toc104580260)

[2.3.4 NoSQL 24](#_Toc104580261)

[2.3.5 Firebase 28](#_Toc104580262)

[2.4 Summary 36](#_Toc104580263)

[CHAPTER 3: BUILD APPLICATION ILEARN ON XAMARIN 37](#_Toc104580264)

[3.1 Pattern application 37](#_Toc104580265)

[3.2 Application analysis 38](#_Toc104580266)

[3.2.1 Usercase of application 38](#_Toc104580267)

[3.2.2 Login usercase 39](#_Toc104580268)

[3.2.3 Classroom usercase 40](#_Toc104580269)

[3.2.4 Information user usercase 41](#_Toc104580270)

[3.2.5 Assignment manager usercase 41](#_Toc104580271)

[3.3 system operation 44](#_Toc104580272)

[3.3.1 Login and register activity 44](#_Toc104580273)

[3.3.2 Classroom activity 44](#_Toc104580274)

[3.3.3 Store activity 46](#_Toc104580275)

[3.4 Deploy in Xamarin 47](#_Toc104580276)

[3.4.1 Create project Xamarin on Visual Studio 2019 (Windows/Mac) 47](#_Toc104580277)

[3.4.2 Build pattern MVVM 50](#_Toc104580278)

[3.4.3 Implementation of MVVM 51](#_Toc104580279)

[3.5 Examples of some functions 55](#_Toc104580280)

[3.5.1 Function login 55](#_Toc104580281)

[BIBLIOGRAPHY 60](#_Toc104580282)

[SUMMARY 61](#_Toc104580283)

IMAGES

[Figure 1: What is Internet? 8](#_Toc104580324)

[Figure 2: Social network 9](#_Toc104580325)

[Figure 3: Development process Xamarin 13](#_Toc104580326)

[Figure 4: Strong point of Xamarin 13](#_Toc104580327)

[Figure 5: List platform supports 14](#_Toc104580328)

[Figure 6: Powerful of Xamarin 14](#_Toc104580329)

[Figure 7: Example code login design in Xamarin 15](#_Toc104580330)

[Figure 8: UI build in 3 platform: IOS, Android, Window Phone 15](#_Toc104580331)

[Figure 9: Share code in everywhere 16](#_Toc104580332)

[Figure 10: Performent when build code 17](#_Toc104580333)

[Figure 11: Model MVVM 18](#_Toc104580334)

[Figure 12: MVVM 20](#_Toc104580335)

[Figure 13: Model MVC 22](#_Toc104580336)

[Figure 14: Model MVP 23](#_Toc104580337)

[Figure 15: Firebase 28](#_Toc104580338)

[Figure 16: Realtime database 29](#_Toc104580339)

[Figure 17: Firebase Authentication 30](#_Toc104580340)

[Figure 18: Firebase Hosting 30](#_Toc104580341)

[Figure 19: Fast development speed 31](#_Toc104580342)

[Figure 20: Firebase + Google 32](#_Toc104580343)

[Figure 21: Firebase ML 33](#_Toc104580344)

[Figure 22: Not open source 34](#_Toc104580345)

[Figure 23: Firebase + Google cloud 35](#_Toc104580346)

[Figure 24: Pattern application Ilearn 37](#_Toc104580347)

[Figure 25: Main function in the system 38](#_Toc104580348)

[Figure 26: Login Usercase 39](#_Toc104580349)

[Figure 27: Classroom usercase 40](#_Toc104580350)

[Figure 28: Information user usercase 41](#_Toc104580351)

[Figure 29: Assignment manager usercase 41](#_Toc104580352)

[Figure 30: File manager usercase 42](#_Toc104580353)

[Figure 31: Store usercase 42](#_Toc104580354)

[Figure 32: Cart usercase 43](#_Toc104580355)

[Figure 33: Notification usercase 43](#_Toc104580356)

[Figure 34: About application usercase 43](#_Toc104580357)

[Figure 35: Login and register 44](#_Toc104580358)

[Figure 36: Classroom activity 45](#_Toc104580359)

[Figure 37: Detail classroom activity 46](#_Toc104580360)

[Figure 38: Store activity 47](#_Toc104580361)

[Figure 39: Visual Studio 2019 47](#_Toc104580362)

[Figure 40: Search Xamarin 48](#_Toc104580363)

[Figure 41: Write information of Project 49](#_Toc104580364)

[Figure 42: Template project 50](#_Toc104580365)

[Figure 43: Create project success 50](#_Toc104580366)

[Figure 44: Pattern MVVM 51](#_Toc104580367)

[Figure 45: Create file in pattern MVVM 52](#_Toc104580368)

[Figure 46: Binding context code 52](#_Toc104580369)

[Figure 47: BaseViewModel codebehind 53](#_Toc104580370)

[Figure 48: Example HomeViewModel 54](#_Toc104580371)

[Figure 49: Binding Text property 54](#_Toc104580372)

[Figure 50: Usermodel and Notification property data change 55](#_Toc104580373)

[Figure 51: Code xaml UI Login 55](#_Toc104580374)

[Figure 52: LoginViewModel code behind 56](#_Toc104580375)

[Figure 53: Init method in contructor 57](#_Toc104580376)

[Figure 54: handle command login 57](#_Toc104580377)

[Figure 55: Request data to firebase server and get response 58](#_Toc104580378)

[Figure 56: LoginUI………………………………………………………………………59   
Figure 57: Wrong password………………………………………………...……………59   
Figure 58: Wrong email 59](#_Toc104580379)

**CHAPTER 1. OVERVIEW OF GRADUATION TOPIC**

Chapter 1 introduces an overview of learning online applications, describes how the application works.

* 1. Problem

In the face of the Covid-19 pandemic that is happening all over the world in general and Vietnam in particular. The pandemic has had a huge impact on the economy, life, health, etc. In which, the most important issue that has been hit the hardest is Education. The situation is complicated. Students can't go to school to study and work, and people who go to work can't go to the company must stay at home.

At that time, the best solution is to work and study at home. Many schools and companies offer online learning applications to work online to solve this urgent problem.

Before such developments, I also developed an online learning and homework application. Not only that, the application is also commercialized and expanded so that it can be developed into an application that brings a lot of excitement and benefits to all users.

* 1. Internet
     1. What is internet?

The Internet is a global network that includes billions of computers and other electronic devices. With the Internet, you can access almost any information, communicate with anyone else in the world, and so on.

You can do all this by connecting your computer to the Internet, also known as going online. When someone says that a computer is online, that is another way of saying that it is connected to the Internet.



Figure 1: What is Internet?

* + 1. How does the internet work?

The Internet is a global network of physical cables, which may include telephone copper wire, TV cable, and fiber optic cable. Even wireless connections like Wi-Fi and 3G/4G/5G rely on these physical cables to access the Internet.

When you visit a website, your computer sends a request over these wires to a server. The server is where web pages are stored, and it acts like your computer's hard drive. When the request comes in, the server retrieves the web page and sends the correct data back to your computer. This query operation, all done in seconds.

* + 1. Benefit

One of the best features of the Internet is its ability to communicate almost instantly with anyone in the world. Email is one of the oldest and most popular ways to communicate and share information on the Internet with the billions of people using it. Social networks allow people to connect in different ways and build communities online.

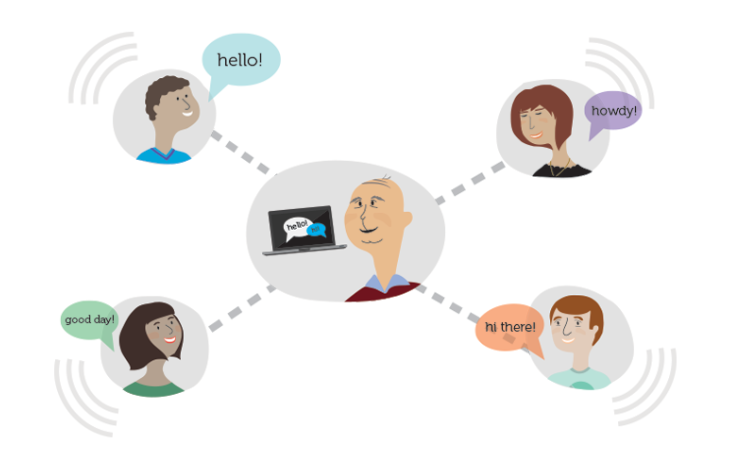


Figure 2: Social network

There are many other things you can do on the Internet. Not only that, today with current technology, many integrated modern devices can connect to the Internet to help people easily access and achieve their personal needs.

* 1. Introduce about application

The application name is: “ILearn”. The application meets the most important requirement of students to study and do homework anywhere. In addition, there is a mechanism to redeem points to create more motivation for users and buy goods, vouchers.

Points are redeemed at a separate Store with many attractive items. Items are updated by Vendors who are partners with the company or Vendor of the App.

In the future, when the application has certain results, it will be developed more widely and towards many convenient functions, meeting the needs of users.

* 1. Summary

With the goals and orientations presented above, the topic that I am aiming for is an Application on the phone on both platforms that are being used the most: Android and IOS.

To do that, we need a suitable framework for developing applications with minimal effort and still achieving high efficiency and goals. In addition, it must also meet the scalability and application to be able to develop and improve continuously to meet all the needs of users and customers.

**CHAPTER 2. SOLUTION ORIENTED TO DEVELOP APPLICATION**

Chương 2 trình bày về vấn đề được đưa ra của đề tài, hướng giải quyết và các công cụ để xây dựng hệ thống, ứng dụng học trực tuyến ILearn trên hai nền tảng Android và IOS.

2.1 Problem

The given problem needs a compact, easy-to-use device so that everyone can learn and work at any time and in any environment. Nowadays, the mobile phone is no longer a stranger to us, it not only provides calling applications but also many entertainment applications, supporting work, ... But today, wherever people go, they almost carry them. the phone with you. Therefore, building ILearn applications on the mobile platform for learning and working brings great benefits and convenience.

However, in addition to developing applications, another way is to develop a Website that functions like an application. However, it will cause some disadvantages like we have to remember the url of the website and have to log in often, using the browser. Instead, developing an app has a huge advantage of being able to store more data, save logins, apps available on the store can be downloaded.

2.2 Solution

I will develop mobile applications. Besides, now there are many operating systems of phones: Android, IOS, Windows Phone.

In which, the number of users of Android/IOS operating systems accounts for the majority of domestic and foreign markets.

So the current tool that best fits this pattern is: Xamarin on Visual Studio. The reason is that I already have a background in Csharp (the language used in Xamarin).

In addition, to create consistency, optimization between platforms and reuse of pre-built resources (Models, Dlls, Databases), we will choose a common pattern model between them. The pattern must be easily accessible for coding. Therefore, the MVVM model is the right choice.

* 1. Introduce about IDE

2.3.1 Visual Studio

2.3.1.1 What is Visual Studio?

Visual studio is one of the most famous programming and website design support tools of Microsoft today and no other software can replace it. Visual Studio is written in two languages, C# and VB+. These are 2 programming languages that help users to program the system easily and quickly through Visual Studio.

Visual Studio is a software programming system produced directly from Microsoft. Since its inception, Visual Studio has had many different versions used. That help for the user can to select the version of the same as the host line of as well as configure to use the most appropriate.

Besides, Visual Studio also allows users to choose the main interface of IDE.

2.3.1.2 Features in Visual Studio

* Code: Like any other IDE, Visual Studio includes a code editor that supports syntax highlighting and code completion using IntelliSense not only for functions, variables, and methods, but also for use with configurations language structures such as: Query or control loop.
* Debug: Visual Studio has a debugger that features both User-level debugging and Kernel-level debugging. This feature works with both machine language-like management code and can be used to debug applications written in languages supported by Visual Studio.

2.3.1.3 Design

* Windows Forms Designer: Used to build GUIs using Windows Forms, designed to build a control or management system that needs a basic interface, the goal is functionality.
* WPF Designer: This feature is similar to Windows Forms Designer, supporting drag and drop metaphor. Use targeted human-computer interaction in Windows Presentation Foundation. But towards a more beautiful interface and easier operation, more expansion.
* Web Designer/ Development: Visual Studio also has a website editor and designer that allows web pages to be designed with drag and drop functionality.
* Xamarin: Tools added to build cross-platform mobile apps.

2.3.1.4 Strong point of Visual Studio

* Visual Studio supports programming in many languages such as: C/C++, C#, F#, Visual Basic, HTML, CSS, JavaScript.
* As a tool to support debugging easily and powerfully such as: Break Point, view the value of variables during runtime, support debugging each statement.
* Visual Studio interface is very easy to use for beginners.
* Visual Studio supports the development of applications: desktop MFC, Windows Form, Universal App, mobile application Windows Phone 8/8.1, Windows 10 and so on
* Visual Studio supports building applications professionally with drag and drop tools.

Visual Studio is used by many programmers around the world.

2.3.2 Xamarin

2.3.2.1 What is Xamarin?

Xamarin is a framework for building cross-platform mobile applications created by the mobile software company of the same name founded on May 16, 2011 by the same engineers who created Mono, Mono for Android and MonoTouch with more than 500 thousand employees. This company uses C# language to write applications that run on Mac, Android, the idea of ​​deploying cross-platform applications using Common Language Infrastructure (CLI) and Common Language Specifications (commonly known as Microsoft .NET).

In November 2013, Microsoft acquired and integrated it to run on Windows 8 environment. With C# open source codebase, developers can use Xamarin tools to write native Android and iOS applications. and Windows with native user interfaces and code sharing across multiple platforms, including Windows and macOS.

According to Xamarin, more than 1.4 million developers are using Xamarin products in 120 countries worldwide as of April 2017

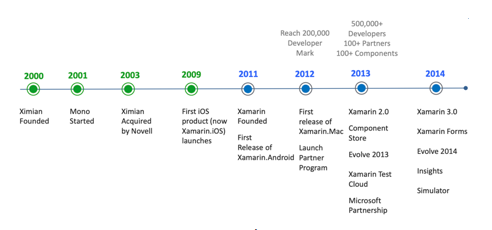


Figure 3: Development process Xamarin

Xamarin tool was introduced to the Vietnamese market by Microsoft on April 18, 2014 for the community of lecturers of colleges and universities in Ho Chi Minh, Ha Noi City and transferred documents and copyrights at preferential prices to students. Based on Microsoft Visual Studio and Microsoft experts demo Xamarin programming to run the app on iOS, Android, Windows Phone and Windows 8.1 environments with different Tools.

At the Microsoft Build 2016 event, Microsoft announced that they will open source Xamarin SDK and provide tools to develop applications using Xamarin for free on Visual Studio.

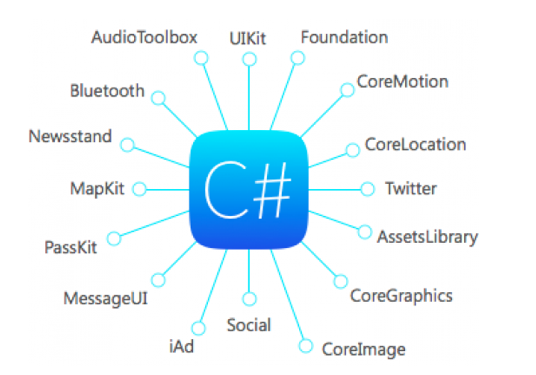


Figure 4: Strong point of Xamarin

2.3.2.2 Xamarin Forms

Xamarin.Forms is a tool that makes user interface code shareable across iOS, Android and Universal Windows Platform apps with 100% C#. Xamarin.Forms includes more than 40 controls and layouts, which are mapped to native controls at runtime. It supports the following platforms:

* Android 4.4 +
* IOS 8+
* Windows 10 Universal Windows Platform, Xamarin Forms also supports the following platforms but is still in Preview mode or developed by a third party.
* Samsung Tizen
* macOS
* GTK#
* WPF

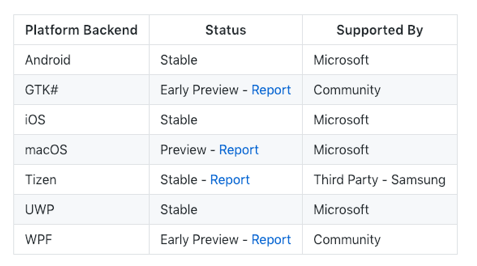


Figure 5: List platform supports

Below is a comparison of code sharing of traditional Xamarin usage and Xamarin.Forms usage. Obviously when using Xamarin.Forms, we have used most of the common code for the entire application.

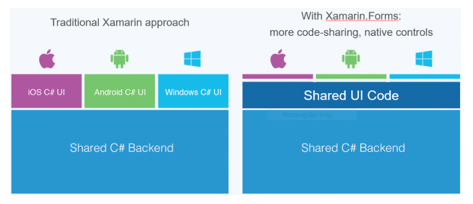


Figure 6: Powerful of Xamarin

The interface of applications using Xamarin is written in a XAML file. It has the structure of an XML file, quite similar to Winform or Android XML.



Figure 7: Example code login design in Xamarin

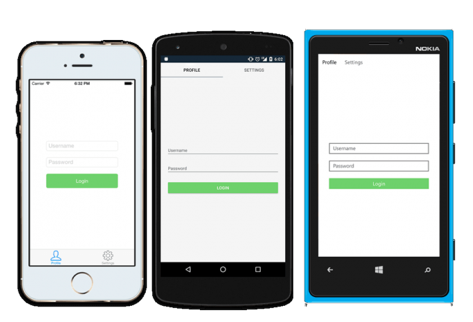


Figure 8: UI build in 3 platform: IOS, Android, Window Phone

2.3.2.3 Advantagas and Defect

2.3.2.3.1 Advantagas

* Share code everywhere: When you write an app using Xamarin's toolkit, you're essentially using an abstraction layer on top of the actual iOS and Android SDKs. This means that you will end up with a fully native app with a native look and feel on each platform. When you create apps on Xamarin, you use the same languages, APIs, and data structures to share an average of 75% of your code across all mobile platforms. This application logic can be easily shared across multiple platforms. Thereby, it is possible to significantly reduce the cost and time of developing mobile applications for the 3 most popular platforms.

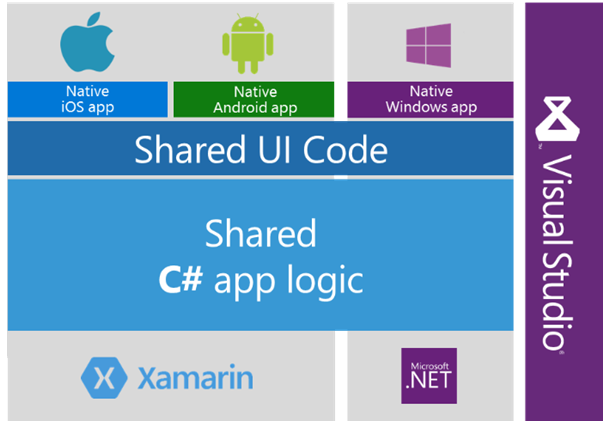


Figure 9: Share code in everywhere

* Performance like native: Unlike the traditional hybrid approach based on web technologies, a cross-platform application built with Xamarin can also be viewed as native. Performance metrics are comparable when compared to performance metrics for Java, Kotlin for Android, and Objective-C or Swift for native iOS app development. What's more, the performance is continuously improved to fully match the standards of native programming. The Xamarin platform provides a solution for testing and monitoring application performance. Xamarin Test Cloud combined with Xamarin Test Recorder tool allows you to run automated UI tests and identify performance issues before the application is released. However, there is a fee for this service.

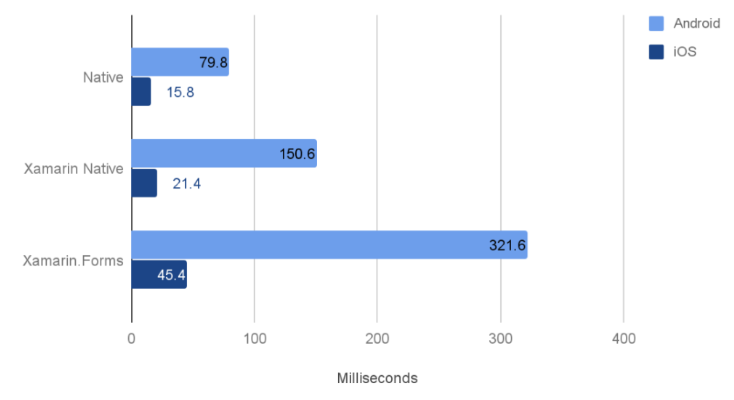


Figure 10: Performent when build code

* Some advantage: UI layout options, OAuth integration, Remote REST APIs integration, Real-time navigation and signal processing technology for location apps, Social network integration, Embedded SQLite database available, Mail XAML library that allows building a wide range of applications, Supports data binding.

2.3.2.3.2 Defect

* Apps perform slower and require more space on the device: Xamarin apps are larger and heavier than native apps. Compared to native application it takes up several Mb more than Java/Objective C respectively. The size of an application coded in xamarin is 3Mb, while the code in Objective C takes up only 172 Kb. The more APIs are used, the more storage is occupied on the device.
* Cons of AOT Compiler: Xamarin Forms also has cons of AOT compiler. It doesn't compile code as neatly as Xcode does. Xamarin's limitations affect both development, testing and uploading the app to the store. The main limitation for iOS is that it does not support automatic code generation. Visual Studio biuld long code. This is what all Xamarin programmers are living with, one build will take a long time from 3 to 10 minutes for a solution with 30 projects.
* Small community support: Xamarin community is less than iOS or Android community, so it is not easy to find an experienced Xamarin developer even though Xamarin is a platform developed with support from Microsoft . According to various sources, Xamarin community accounts for 10% of the global mobile programming community based on the fact that the platform provides maximum support for developers. Specifically, there is a specialized education platform at Xamarin University, providing knowledge and training for those who are new to this technology. As a result, the learning curve of an experienced C#/.Net engineer will be minimized.

2.3.3 Model MVVM

2.3.3.1 Concept

MVVM (Model-View-ViewModel) can be said to be an architectural model favored by many programmers in the programming community. This model was born for applications that use XAML language to define application interfaces such as: Windows Phone 8.0, 8.1 Silverlight/RT, WPF, Silverlight, Windows RT, Universal Apps,... Benefits from this model bring is a lot. But the most prominent benefit is separating the interface design and programming logic code that are not dependent on each other.

MVVM is not a framework or a library, an api… it is merely a guide for you to define the structure of your application. MVVM is developed based on MVP . architecture.

The mvvm model allows separation of data (Model), executable code (logic or ViewModel) and user interface (View).

In traditional models, we usually handle Click event and write code to execute directly on a Button, but MVVM model does not allow to do this.

In the MVVM model, controls like Button, ListView, SearchBar, etc. cannot bind directly to the data, but through the Command property - which is a property of type ICommand.



Figure 11: Model MVVM

* + - 1. How does MVVM work?
* View: Components of the application interface. Similar to the MVC pattern, the View is the only component that the user can interact with in the program, it is the element that describes the data. Another difference compared to other models is that the View in this model is more active. It has the ability to perform behaviors and respond to users through the following features: binding, command.
* Model: The same is true in the MVC pattern. Models are objects that help access and manipulate real data.
* ViewModel: Intermediate layer between View and Model. The ViewModel can be seen as an alternative to the Controller in the MVC pattern. It contains the necessary code to perform data binding, command.

In addition, to make it easy to understand the ViewModel, it will take care of synchronizing data from the model to the View. The relationship between View and View-Model is that the View will be mapped to the View-Model, but the View-Model does not know anything about the View. It is hidden through the use of Data-binding and the mechanics of the Observer pattern. A View-Model can be mapped from multiple Views.

Note: As in the MVVM model, the layers below will not know anything about the layer above it. ViewModel knows nothing about View, a ViewModel can be used for multiple Views (one-to-many). The ViewModel uses the Observer design pattern to communicate with the View (often called binding data, can be 1-way or 2-way depending on the needs of the application).

* + - 1. MVVM structure in project

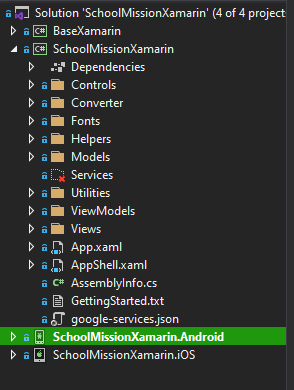


Figure 12: MVVM

* Views: In the Views folder there are interface files. And each interface file has a code-behind class included. Especially the code-behind file we will not use, everything we need to do will be moved down to the ViewModel class. Of course you can code in the XAML code-behind file but that would break MVVM convention. You can declare a datacontext attribute or some other setting, but keep the code to a minimum here. Views are used in conjunction with MVVM patterns… It is intended to provide a neat conceptual separation between UI and presentation logic and data.
* Models: In the Models folder where you create classes containing data and any validation links and business logic to ensure data integrity, you can separate out another Repositories folder. They are used as part of the MVVM . model.
* ViewModel: Similarly, we also create a folder ViewModels. Usually an interface file, we create a corresponding ViewModels class (sometimes we create many subclasses to help simplify the code file and call them in the main ViewModels class).

ViewModels will use models if need to define data. The connection between View-ViewModel helps them to send and receive data, to understand clearly we need to understand the concepts of Binding, DataContext, Behaviors SDK. Thus, we separate the code-behind of the View and bring it down to the View Model.

In addition, a ViewModels class contains the presentation logic and state of the application. ViewModels need to contain the functionality of the application. ViewModels define properties, commands, and events, to convert controls in the view need data-bind.

2.3.3.4 Data Binding

Data Binding is a technique used to create a connection between the interface (UI) and data through the business logic. Thanks to Data Binding, the UI can automatically re-update to show changes in the data. In addition, Data Binding in WPF also supports different dimensions, meaning that changes can be updated from the UI to the data. The binding technique in the mvvm model is really a new step forward, satisfying what most programmers expect.

If you learn about this feature, you will not be surprised when many people say that data binding is the core component that makes up the working mechanisms in WPF. You can bind source and destination data from any object: like a window, simple controls like TextBlock to a complex usercontrol.

It's all done easily, quickly, efficiently, and possibly without the use of any lines of code-behind (C#, VB.NET, ...).

2.3.3.5 Data Template

Data Template is a technique used to create an interface template. Templates are only applied to Controls. A template in WPF defines how and how data or controls should be displayed on the screen.

Particularly about Data Template, this function helps data (non-visual) to be attached to a structure consisting of one or more visible components. And so, the data will be displayed to the window visually according to the programmer's wishes. Like Data Binding, this feature does not require you to know in the code-behind of the application.

2.3.3.6 Command

Data Binding and Data Template in wpf help users see what is in the data and can update that data. However, to receive user interaction and processing, WPF provides a feature called command. Commands can be viewed as data and provided to the user through the binding . function.

A command binding allows you to optionally define which handler methods, keyboard shortcuts, or mouse actions to trigger.

2.3.3.7 Advantages and Defect of MVC, MVP và MVVM

2.3.3.7.1 Advantages of MVC

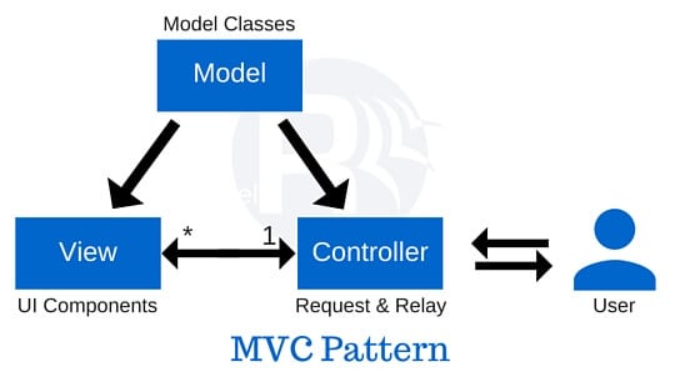


Figure 13: Model MVC

Advantages: The MVC pattern has many advantages, namely

* Lightweight, saving bandwidth: MVC does not consume much viewstate, so it saves bandwidth. The operations of sending and receiving data are happening continuously. Therefore, using this model the website / application works more stable.
* Can check, detect software errors easily.
* Easy to separate Model and View parts.
* This model has a simple structure. You don't need too much technical knowledge to use it.
* Good support for SEO development platforms: You can easily create SEO URL codes to attract traffic to any application.

Defect**:** Besides the above advantages, MVC also has some disadvantages

* Controller and View are related to each other. So, when changing in View, it means you will have to change in Controller.
* It is difficult to execute unit tests because the Controller and Android API are closely related.
* Over time, the Controller will become difficult to control as more and more code is written.
* MVC is suitable for large projects. For small projects, this model is quite cumbersome and time-consuming in transferring data.
* Make it difficult to navigate the project's code.

2.3.3.7.2 Advantages and defect of MVP

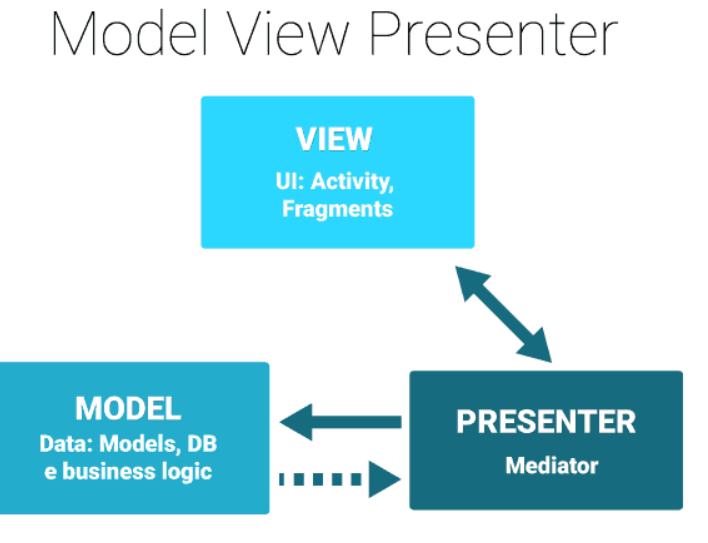


Figure 14: Model MVP

Advantage:

* + We can easily write unit test for presenter because it is not tied to any view, it works independently of View and is not tied to any Android API.
  + MVP has a clearer code structure than MVC, so it is quite easy to understand and use. Less bugs, easier code review.

Defect:

* + MVP model over time, Presenter will gradually grow up due to scattered business logic. It will be difficult for users to control and split the code when the Presenter is too big.
  + It becomes cumbersome when we build with small applications, or with simple Activity.
  + Difficult to reuse logic code in Presenter for other Views.

2.3.3.4.3 Advantages and defect of MVVM

Advantages:

* + Doing Unit testing now will be very easy, because you really don't depend on the view.
  + MVVM will create effective interaction between designer and developer.
  + Increase the ability to reuse components or change the program interface without having to rewrite too much code.
  + Fast, simple application development, easy to upgrade, maintain...

Defect :

* + Maintainability as the view can assign both variables and expressions, the extraneous logic will grow over time, affecting the addition of code to the XML.
  + For small projects, the application of MVVM model is cumbersome and time consuming in the development process. Time consuming data transfer of components.
  + For larger project, it makes it difficult and time consuming to design ViewModels.
  + Data binding for all components makes debugging difficult when the database is complex.

2.3.3.7.4 Applicability of model MVC, MVP, MVVM

You should use MVP if it is not possible to bind data via DataContext (for example with Windows Forms).

The MVVM model should be used in cases where it is possible to bind data through the DataContext (eg: WPF).

As for MVC, it should be used when the connection between the View and the rest of the application is not always available (eg web API). At that time, you cannot use MVP or MVVM effectively and must choose the MVC model.

MVVM can be said to be an architectural model favored and used by many programmers. MVVM has inherited the inherent advantages of MVP, combined with the advantages of data binding to provide a pattern capable of dividing components with separate functions, easy to maintain, redesign. MVVM also offers the ability to test very easily, making work more efficient.

2.3.4 NoSQL

2.3.4.1 What is NoSQL?

A NoSQL Database is a Database built specifically for data modeling and has a flexible schema for building modern applications. NoSQL databases are widely recognized for their ease of development, functionality, and performance at scale. This page has resources to help you understand more about NoSQL databases and get started.

2.3.4.2 Operate

NoSQL databases use multiple data models to access and manage data. These types of databases are optimized specifically for applications that require a flexible data model with large amounts of data and low latency, which can be achieved by easing some of the uniformity constraints consistency of data from other databases.

Schematic model for a simple book database:

* In a relational database, records about a book are usually decomposed (also known as "normalized") and stored in separate tables, and relationships are specified by constraints foreign key and primary key. In this example, the Books table has columns for ISBN, Book Title, and Version Number, the Author table has columns for author ID and Author Name, and finally, the Author–ISBN table has columns for author ID and ISBN. The relational model is designed to allow these databases to enforce referential integrity between multiple tables in the database, normalized to reduce redundancy, and generally optimized for caching purposes.
* In NoSQL databases, records about a book are typically stored as JSON text. For each book, the item, ISBN, Book Title, Version Number, Author Name and Author ID are stored as attributes in a single document. In this model, data is optimized for intuitive development and horizontal scalability.

2.3.4.3 Advantages

* Flexibility: NoSQL databases typically offer flexible schemas that make development faster and more repeatable. The flexible data model makes NoSQL databases the ideal choice for unstructured or incompletely structured data.
* Scalability: NoSQL databases are typically designed to scale using distributed clusters of hardware rather than scale with expensive and powerful server additions. Some cloud service providers treat these activities non-publicly as a fully managed service.
* High performance: NoSQL databases are optimized for specific data models and access patterns giving greater performance gains than trying to achieve the same level of functionality with a database whether the relationship.
* Extremely practical: NoSQL databases provide extremely practical APIs and data types built specifically for each respective data model.

2.3.4.4 Difference between SQL and NoSQL

|  |  |  |
| --- | --- | --- |
| Characteristic | SQL | NoSQL |
| Optimal workload | Relational databases are designed for online transaction processing (OLTP) applications in a highly stable transaction and are suitable for online analytical processing (OLAP). | NoSQL databases are designed for data access patterns, including low latency applications. NoSQL search databases are designed for analysis of incomplete structured data. |
| Data model | The relational model normalizes data into tables made up of rows and columns. Diagrams clearly specify tables, rows, columns, indexes, relationships between tables, and other database elements. The database will enforce referential integrity in relationships between tables. | NoSQL databases offer a variety of data models such as key-values, documents, and graphs, optimized for optimal scale and performance. |
| Attribute ACID | Relational databases have the following attributes: Elemental, Consistency, Distinction, and Perseverance (ACID):  • Elementality requires that the transaction be either fully executed or not executed at all.  • Consistency requires that when a transaction is made, the data must be consistent with the database schema.  • Isolation requires that concurrent transactions be executed separately from each other.  • Sustainability requires being able to recover from a system failure or sudden power failure to the last known state. | NoSQL databases often make trade-offs by relaxing some of these ACID properties of relational databases for a more flexible data model capable of scaling horizontally. This makes NoSQL databases a great choice for low-latency, high-throughput use cases that need to scale horizontally beyond the limitations of a single instance. |
| Performance | Performance often depends on the drive's subsystem. Usually, optimization of queries, indexes, and table structure is required for maximum performance. | Performance is often seen as a function of implicit hardware cluster size, network latency, and the application making the call. |
| Area | Relational databases typically scale by increasing hardware compute capacity or by adding copies of read-only workloads.. | NoSQL databases are often highly fragmented as key-value access patterns are scalable using a distributed architecture to increase throughput, providing consistent performance at near-scale. like unlimited. |
| API | Data storage and retrieval requirements are communicated using queries that are consistent with the structured query language (SQL). These queries are analyzed and executed by the relational database. | The object-based API allows application developers to easily store and retrieve data structures in memory. Fragment key looks for key-value pairs, column sets, or incomplete structured text containing chained application objects and properties. |

2.3.5 Firebase

2.3.5.1 Concept

Firebase is a platform for web and mobile app development, including simple and powerful APIs, without the need for a backend or server.

Firebase is a cloud-based database service. Accompanied by an extremely powerful server system of Google. Its main function is to help users program applications by simplifying operations with databases.

Specifically, simple API application programming interfaces. The aim is to increase the number of users and get more profits.

In particular, it is also a versatile service and extremely good security. Firebase supports both Android and IOS platforms. It's not surprising that many developers choose Firebase as the first platform to build apps for millions of users worldwide.



Figure 15: Firebase

2.3.5.2 Firebase work

* Firebase Realtime Database?

When you sign up for an account on Firebase to create apps, you already have a real-time database. The data you receive is as JSON. At the same time, it is always synchronized in real time to all client connections.

For cross-platform applications, all clients use the same database. It is automatically updated with the latest data whenever the programmer develops the application. Finally, all of this data is transmitted over a secure SSL connection secured with a 2048-bit certificate.

In the event of a network outage, data is saved locally. So when there are any changes are automatically updated to the Firebase Server. Besides, for local data older than the Server, it is automatically updated to get the latest data.

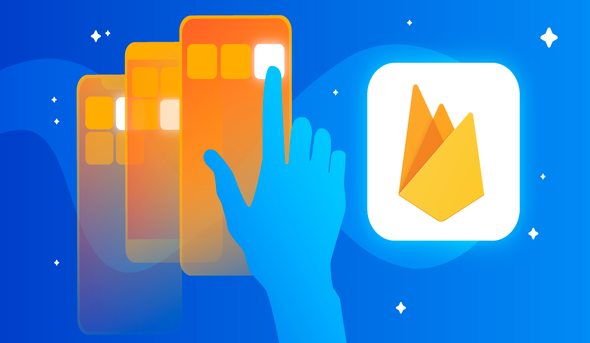


Figure 16: Realtime database

* Freebase Authentication?



Figure 17: Firebase Authentication

The outstanding activity of Firebase is building user authentication steps by Email, Facebook, Twitter, GitHub, Google. Also anonymous authentication for applications. Authentication can help keep users' personal information safe and ensure that their accounts are not stolen.

* Firebase Hosting?



Figure 18: Firebase Hosting

The final way Firebase works covered in this article is to provide hosting. Hosting is delivered via SSL security technology standard from CDN network. So what is a CDN?

CDN stands for Content Delivery Network, which is a network of servers that keep a copy of the static content inside the website and distribute it to many PoP servers. The network of CDN servers is located all over the globe. From PoP (Points of Presence), data will continue to be sent to the end user. Through a CDN, a copy of the content on the nearest server is returned to the user when they visit the website.

2.3.5.3 Advantages of Firebase

* Create an account and use it easily: Firebase allows users to sign in with a simple Google account. Firebase's Spark plan is free and offers plenty of features to help developers get started. As the requirement increases, I choose the paid Blaze plan with more advanced features.
* Fast development speed: Firebase is a suitable app development option that can save developers time and reduce time to market for apps. Normally, every Developer needs to have access to Server and Host to create and maintain databases and backend services. Therefore, it is necessary to have a Backend Developer and a Frontend Developer to build applications. However, this can often lead to bugs and issues that can cause app crashes and increase development costs. Through the use of Firebase and Firestore, Frontend Developers can manage and reduce the time it takes to get all the work done.

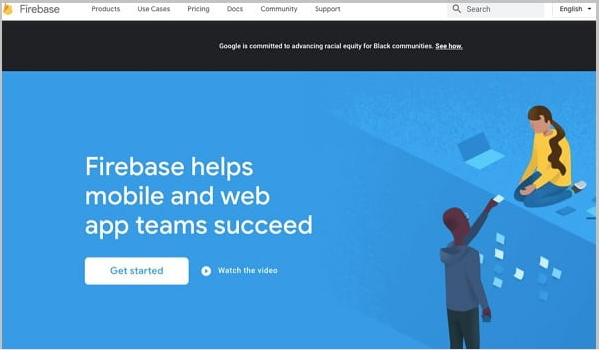


Figure 19: Fast development speed

* Multiple services in one platform: Firebase also provides a complete list of products to assist Developers in their development. The two database options are Firestore and Firebase's Realtime Database. Likewise, Firebase lets you make Cloud Media storage easy. It also enables serverless application development through the integration of Cloud Functions. Firebase covers the entire application development cycle. The platform contains features for building, releasing, and monitoring applications. Also, as the last step of the application development cycle, it provides tools to attract users and keep them using it. This is also an advantage of Firebase that I appreciate quite a bit.
* Powered by Google: Firebase is powered by Google, one of the most prominent and trusted names in the tech world. As far as I know, since it was acquired, Firebase has gone through a series of changes, evolving and becoming the trusted platform it is today. It harnesses the power of Google Cloud and many Google services. Firebase is now part of the Google Cloud Platform. It works well with other Google Cloud services and integrates with many third-party services.

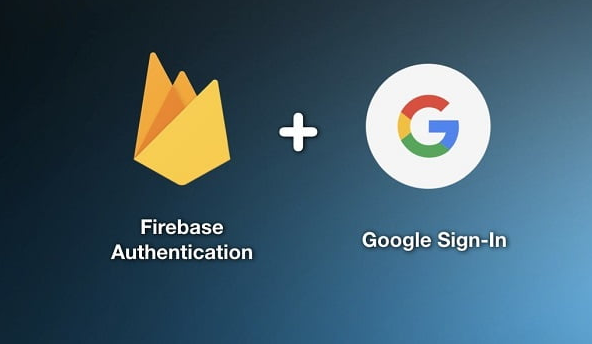


Figure 20: Firebase + Google

* Focus on UI development: Another advantage that makes me love Firebase is that it allows Developers to focus on creating Frontend code for mobile apps. This platform makes application development convenient and reduces costs significantly. Using Firebase also allows companies, developers to standardize the Backend environment under a single and easy to learn technology. The Backend Pattern reduces the amount of training required to support it and allows the Developer to focus on UI development.
* Serverless Firebase: I don't think it's an easy task to scale up or down. In particular, scaling a database cluster is challenging, and optimizing performance for massive workloads requires experienced professionals. Firebase solves this problem and provides a completely serverless environment. Firebase comes with a Serverless Architecture. Therefore, you won't need to worry about server infrastructure.
* Machine Learning: According to Gartner, 30% of businesses will use Machine Learning (ML) in part of their process. Firebase is also beneficial because it gives developers the option to rely on Machine Learning. Firebase comes with ML suite with inbuilt APIs for various mobile platform features like text recognition, face recognition, image labeling, barcode scanning, etc.



Figure 21: Firebase ML

* Generate Traffic: Firebase supports app indexing to allow users to re-engage customers searching on Google. Specifically, it provides app links on Google search. App ratings can also be enhanced on Single Search by app indexing. This exposes your app to new users who can install it.
* Bug Tracking: Next, I'd like to introduce you to Firebase's Crashlytics feature. This is a great tool to find and fix problems quickly. Firebase can track both non-fatal and fatal errors. Also, reports are generated based on how the bug affects the user experience.
* Backups: Firebase ensures optimal security and data availability through regular backups. Applications are protected from any possibility of data loss by relying on the platform's automatic backup feature. Using the Blaze plan, I was also able to easily configure Firebase Realtime Database to perform automatic backups.

2.3.5.4 Limitations of Firebase

* Not Open Source: Firebase is not an open source option for mobile app development. This makes it a suboptimal choice for many developers. Firebase source code cannot be modified by users. In my opinion, this is the biggest limitation of Firebase and prevents the community from improving the product.



Figure 22: Not open source

* Users do not have access to source code: This is another major limitation of Firebase for users. Not having access to the source code can be quite difficult for some developers. Especially for large applications, switching to other vendors is not an easy task and to do this the entire Backend will need to be rebuilt from the ground up.
* Firebase does not work in many countries: Firebase is a Subdomain of Google. Its official Web site is https://firebase.google.com and is blocked in many countries. As far as I know, Google along with other Google services are blocked and inaccessible in many countries, including China.
* Works only with NoSQL Databases: Both Firestore and Firebase Realtime Database provide a NoSQL structure. They have no option to use relational databases. Although Firebase has made significant improvements on Cloud Firestore compared to RTDB, running complex queries is still a challenge for certain users. With Firebase, users can't handle data migration as quickly as a plain SQL database. Firebase uses JSON and has almost no SQL features. So migrating from the database won't be easy.
* Slow queries: Not all databases are ideal for all use cases and Cloud Firestore is no exception. While it's a great solution for scaling your application, providing offline query support and real-time updates, it also has limitations like:
* Maximum API request size 10 MiB.
* No native aggregate queries.
* Document size limit is 1 MiB.
* Up to 1M concurrent connection.
* Query performance is slow.
* Not all Firebase services are free: Cloud functions are only available on the Blaze plan and users cannot try the service on the Spark plan. The Cloud Vision API in machine learning is also not available in the platform's free Spark plan. The service is only available to users of the Blaze plan and charges $1.5/k API.
* Firebase is expensive and volatile: Firebase is a feature-rich mobile app development platform but also an expensive platform for some people. It offers a free plan but has certain limitations. Firebase offers more useful features but will also cost more than a pure IaaS provider. All in all, I think it's cheaper to host an app yourself than using Firebase. Another aspect that makes Firebase expensive is that it is a proprietary technology. Firebase must offset the technical investment of its user base. Firebase offers services under a usage-based pricing model, and there is no way to limit pricing. There is no fixed price list for the platform, and tracking resource usage can be a tricky affair. Estimating the cost of Firebase can be difficult for some users, especially when scalability requirements cannot be predicted. Pricing is complex, and Firebase has specific pricing for every feature. As far as I know, Firebase used to provide the Flame package. It's a fixed price plan of 25 USD/month. But it was removed in January 2020.
* Runs on Google Cloud only: Firebase is now part of Google and its infrastructure runs entirely on Google Cloud. You won't have the option to run Firebase on other cloud providers like AWS, Azure, or Digital Ocean.



Figure 23: Firebase + Google cloud

* Lack of Dedicated Servers and enterprise contracts: Firebase does not have Dedicated Servers options or enterprise contracts. The only way to use Firebase is to use a Serverless architecture with less flexibility. One limitation that I find quite inconvenient when using Firebase is that it doesn't offer pricing plans, contracts, or enterprise support. However, the Blaze package can be considered an acceptable alternative. But I'm still hoping for more options and support.
* Does not provide GraphQL APIs: Firebase does not provide GraphQL APIs as part of the standard setup. While there are alternatives to implementing GraphQL with Firebase, REST is still the default option of the platform.

2.4 Summary

With the above solution, we have selected programming tools and data to support the process of building ILearn app applications. Due to the lack of solid knowledge, but the theory presented above has not been completed and there are references on the internet, books and so on.

Next chapter we will proceed to build App ILearn on Xamarin platform with MVVM model.

CHAPTER 3: BUILD APPLICATION ILEARN ON XAMARIN

Chapter 3 will proceed to implement the MVVM model on Xamarin to create an overall template. Creating a premise for the development of ILearn applications on two technology platforms

3.1 Pattern application

Building a remote learning and working application requires an intermediary data store for all users and executes the necessary queries for each user request.

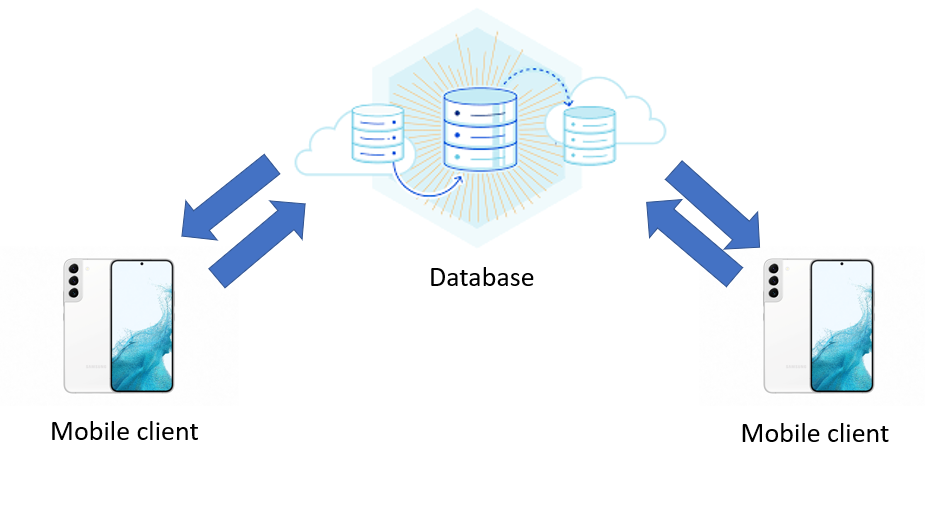


Figure 24: Pattern application Ilearn

The application is organized in the form of Database - Client. The phone with the Internet-connected app installed will send the Request to the Database (NoSQL), the Database will perform the query and return the results to the device and display it to the user. The application can provide a wide range of users to access and conduct learning and teamwork remotely.

This application builds on phones with Android 4.0 (Ice Cream Sandwich) or higher, iOS 8 and up. The application includes the following functions:

* Login: Allows users to access and applications to conduct work and study remotely.
* Account information: users can view user information, avanta.
* Change password, logout: Change user password, logout and select logged in accounts.
* Classroom: Users can Join existing classes with Code or create their own class and send the code to others to join the class.
* Homework: In each class, the class leader will assign assignments and assignment times for each task and students in the class will have to do and submit assignments on time in order to be assessed and graded.
* Store: Points when doing exercises can be exchanged for some items from Vendors who collaborate with the application and can be exchanged directly when going to the store with a new code.

3.2 Application analysis

3.2.1 Usercase of application

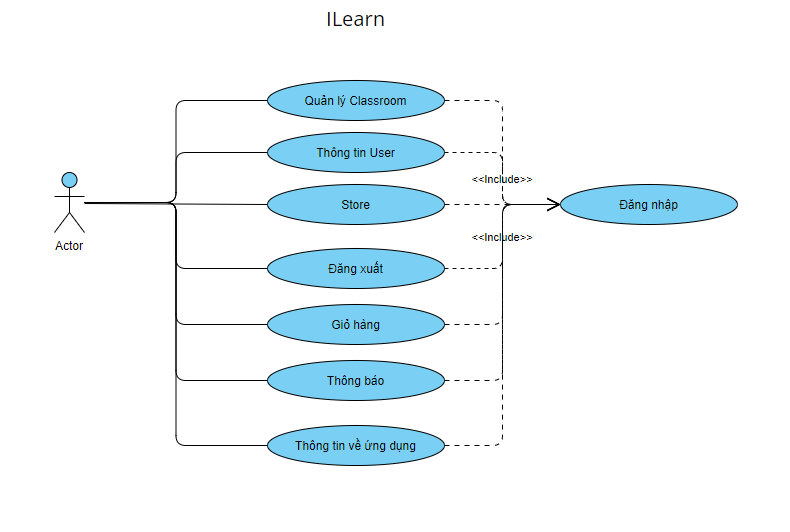


Figure 25: Main function in the system

Actors participating in the system:

* Teacher: has full functions such as: create task, upload file, mark task, etc
* Tutor: Some functions are limited compared to teachers: view graded tasks, etc
* Student: limited functionality: do tasks, join classes, etc

Application features:

* Login, Register: allow user access to application, create user account.
* Classroom: View all classes of users participating, or users owning. Manage tasks, files, etc
* Notification: allow the user to see all the notifications that have occurred.
* Store: where users proceed to use their points to exchange items.
* About: application information, application orientation and goals.
* Cart: contains a list of products purchased from the Store and details of the product.
* User information: User can edit basic information, change password, upload avanta.
* Logout: exit the app.

3.2.2 Login usercase

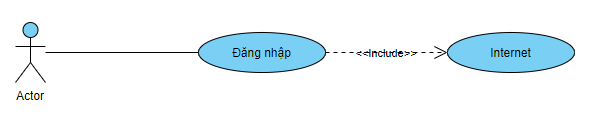


Figure 26: Login Usercase

Description: to log in to the application, users are required to have an internet connection to be able to perform login or other operations in the application. In addition, you can also reset your password or create a login account.

3.2.3 Classroom usercase

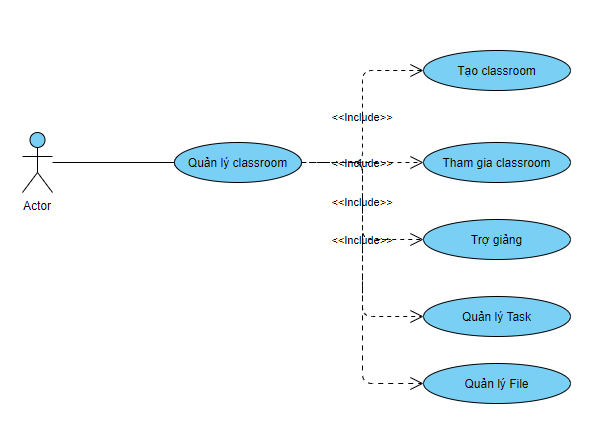


Figure 27: Classroom usercase

Description: Classroom can be created by any user as they can create their own study group. The person at the top of the class can create tasks for students who participate in the task and complete the assignment on time. You can also upload files for others to refer to and learn from. In addition, completing the task with high scores can be Points.

3.2.4 Information user usercase

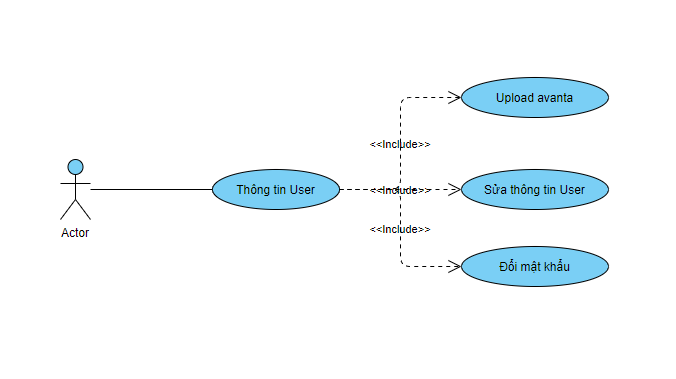


Figure 28: Information user usercase

Description: User can change the basic information here. Changing the password is also done by the user here and can also be uploaded to the user avanta.

3.2.5 Assignment manager usercase

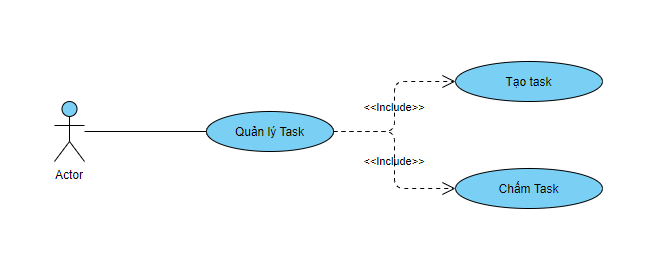


Figure 29: Assignment manager usercase

Description: This function is only available to classroom creators. They can create daily or weekly tasks with pre-set deadlines. When the time is up, students cannot submit or correct their work, then the teacher will conduct grading and give feedback to students. The answers to students' questions will be provided by the teaching assistant who can see the student answers and the teacher's feedback.

3.2.6 File manager usercase

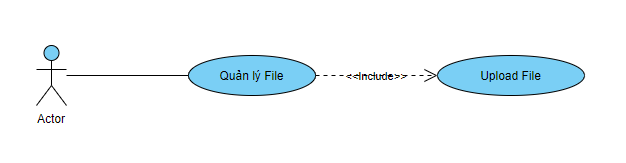


Figure 30: File manager usercase

Description: This function is only available to classroom creators. They can upload lectures or task-related documents for students to download and study.

3.2.7 Store usercase

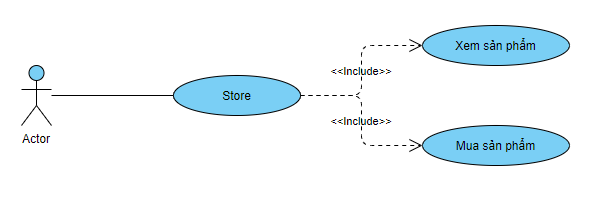


Figure 31: Store usercase

Description: This store will be an important function so that users can use their points to exchange items. All items in the Store will be provided and exchanged by the Vendors collaborating.

3.2.8 Cart usercase

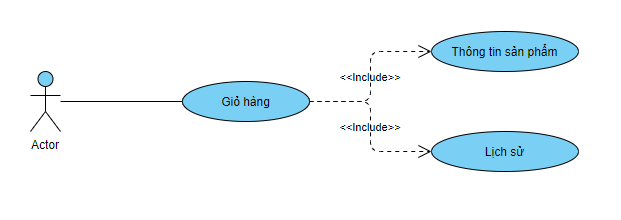


Figure 32: Cart usercase

Description: User can see the list of products that have been purchased from the shopping cart. Can see the code that will be used to redeem products from Vendors upon arrival at that store.

3.2.9 Notification usercase

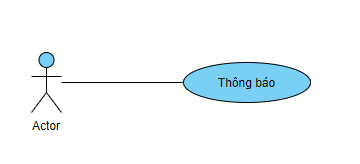


Figure 33: Notification usercase

Description: all user related notifications will be displayed here.

3.2.10 About application usercase

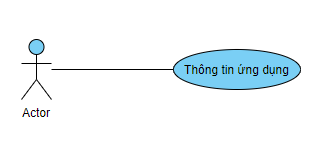


Figure 34: About application usercase

Description: information about the application or standards, orientation will be introduced here.

3.3 system operation

3.3.1 Login and register activity

Graph description: To use the system, users need to log in to the account that the admin has created for them. The system must be connected to the internet. When there is a connection, the user goes to the Cloud section to log in. The login section has two parts, Username and Password.

When entering Username and Password, the system will check whether this account exists, if not, there will be a message "User invalid", if the system continues to check the password, if it is not correct, it will report "Invalid password" password" and ask to re-enter, when the password is correct, the user logs in successfully.

In addition, when the user forgets their password they can select “forgot password” and enter their email to confirm and a password reset email will be sent to their email to be able to enter the new password.

The app will automatically save the email the user enters.

With the registration function, users will enter basic information to register an account

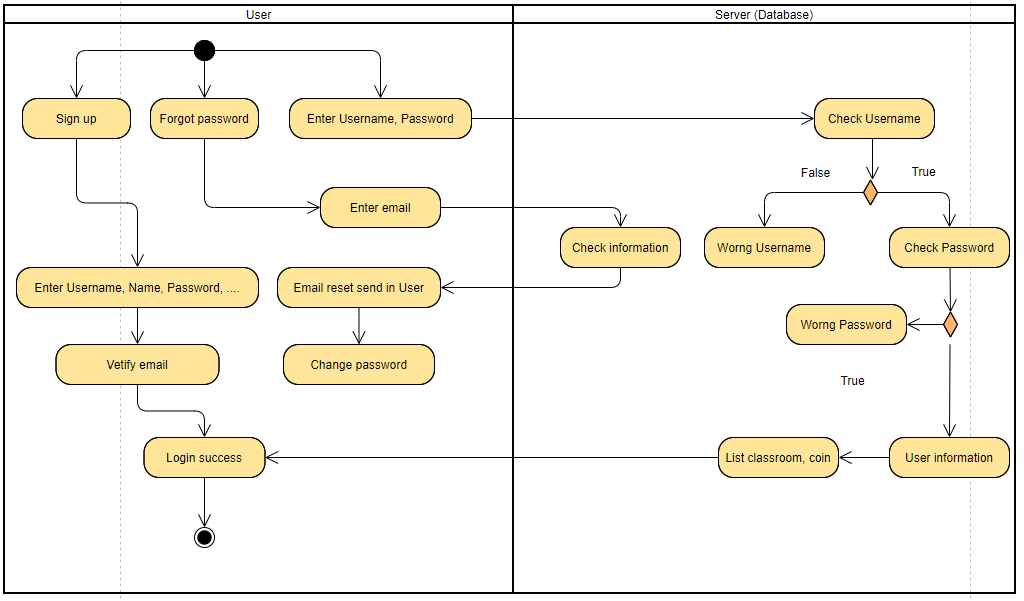


Figure 35: Login and register

3.3.2 Classroom activity

Description of the chart: To have a classroom, the user can join the classroom by entering the code of that class. In addition, they can also create their own class with basic information such as: name, class description. Then a new class will be created and will have a unique code.

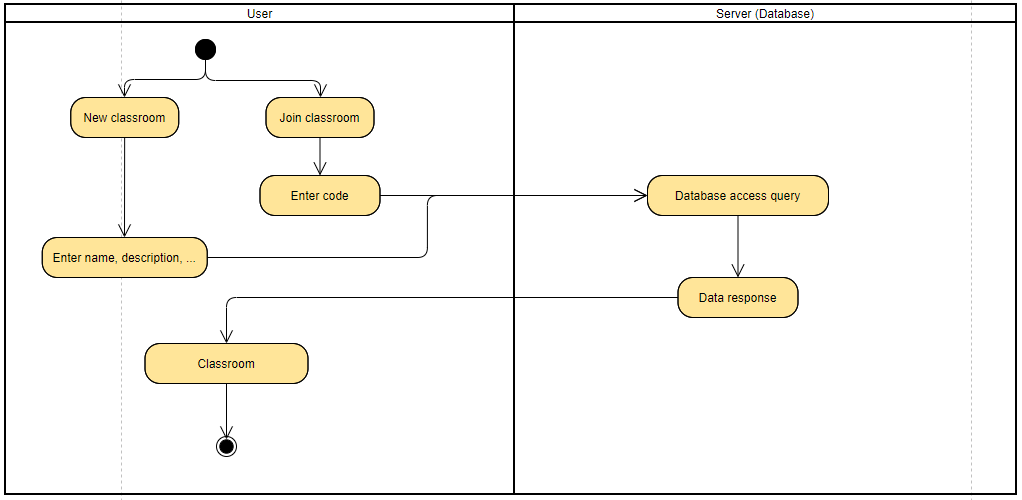


Figure 36: Classroom activity

Function Description:

* The teacher is the host of the classroom. Fully functional person
* Teaching assistant: can appoint any student to be his/her teaching assistant. Teaching assistants can see the student's grades and work and the teacher's comments on the student's work.
* Assignment in date: The teacher will create tasks with full content and time. When the assignment is posted, the student will proceed with the task and submit it on time. Past the time when the student cannot submit the work, once the student has submitted the work, he or she cannot edit or cancel the submission. Posts that have been graded in the student's interface will have points displayed and students will be given coins if the student's score exceeds the basic score threshold, the number of coins corresponding to the information contained in the assignment that the teacher has. have issued.
* Assignment due date: when the assignment has expired, the teacher will enter that assignment and conduct grading. When all student work is terminated the status of the assignment will be changed.
* File: teachers can upload the cast files or solutions to the classroom for students to download and refer to..

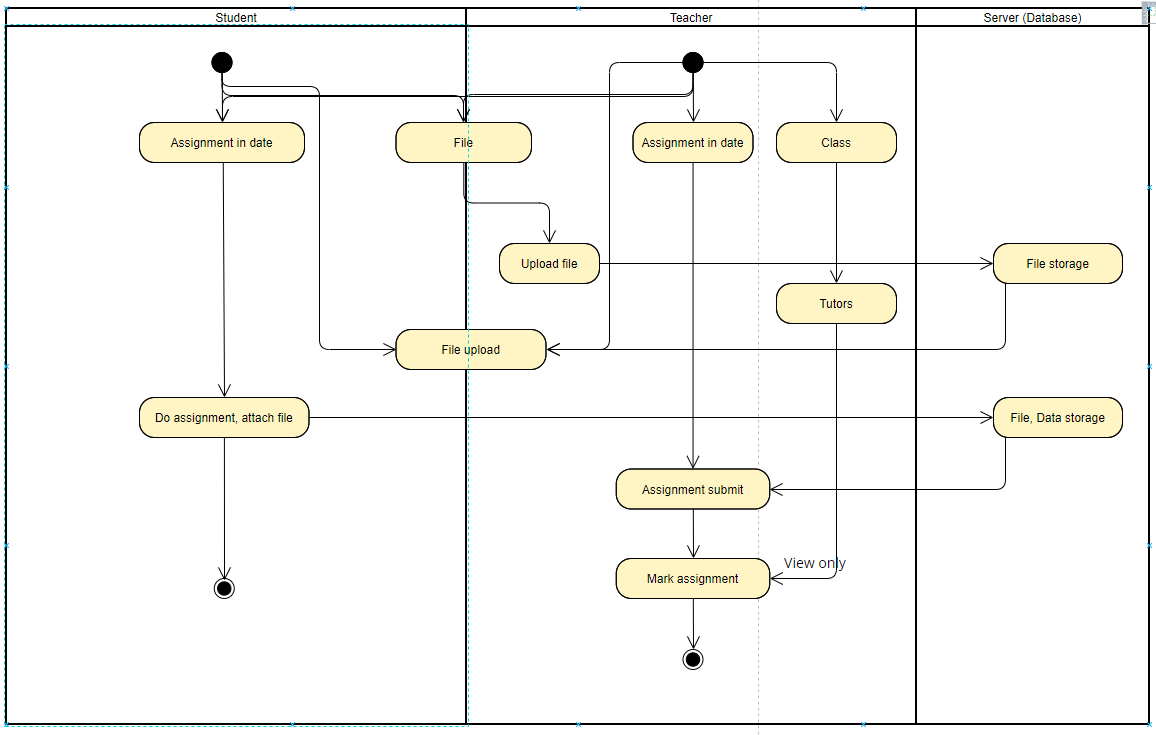


Figure 37: Detail classroom activity

3.3.3 Store activity

Function description: When users have points, they can proceed to exchange some items in the store. When items are purchased, they will be added to that person's shopping cart. In the shopping cart will contain all the information of the product with a code to be redeemed at the store. All products in the store are the products of the vendor provided and sponsored.

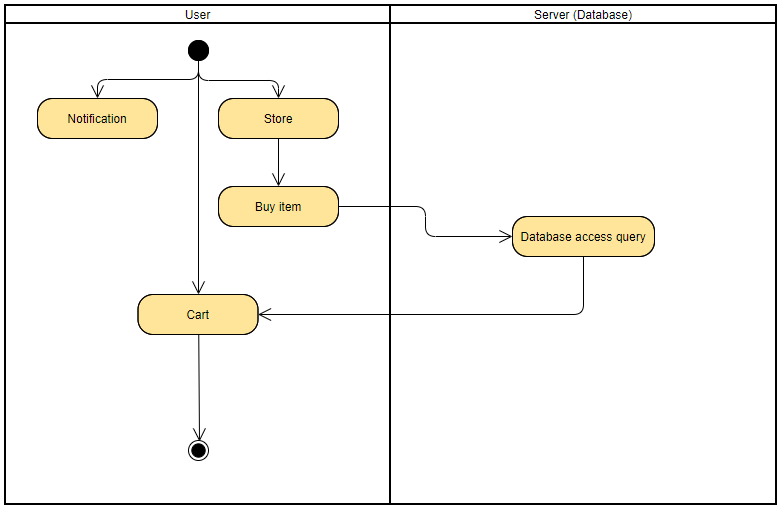


Figure 38: Store activity

3.4 Deploy in Xamarin

3.4.1 Create project Xamarin on Visual Studio 2019 (Windows/Mac)

Step 1: mở visual studio 2019 chọn phần “Create a new project”

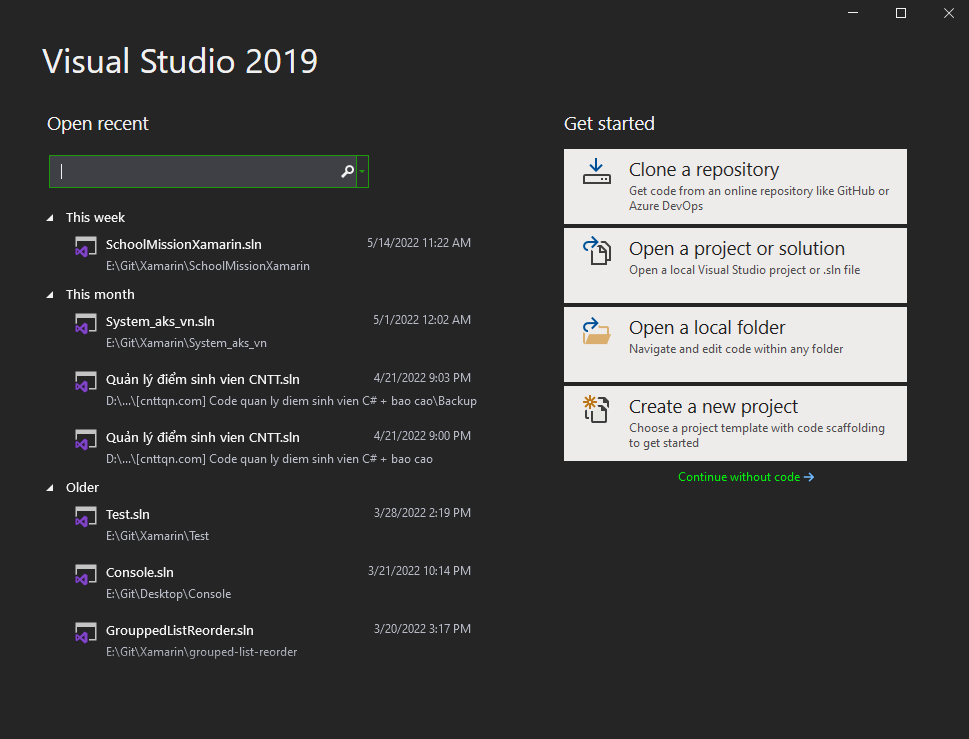


Figure 39: Visual Studio 2019

Step 2: Search Xamairn and choose Mobile App (Xamarin.Form)

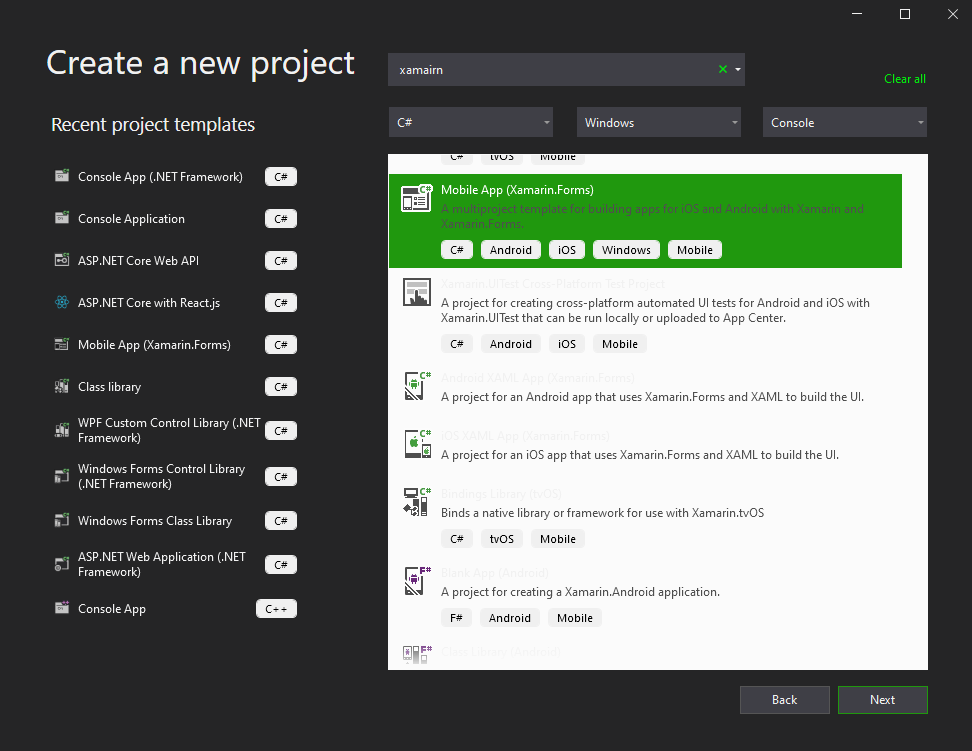


Figure 40: Search Xamarin

Step 3: Write Project Name and location save Project then click Create

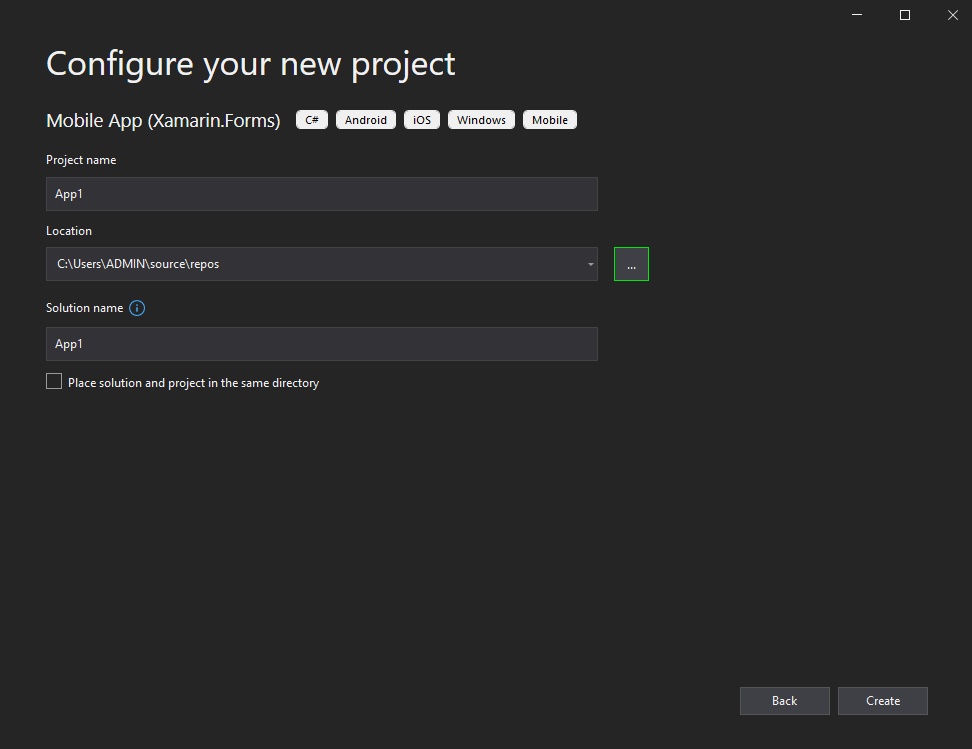


Figure 41: Write information of Project

Step 4: Choose type Template Project (Blank project)

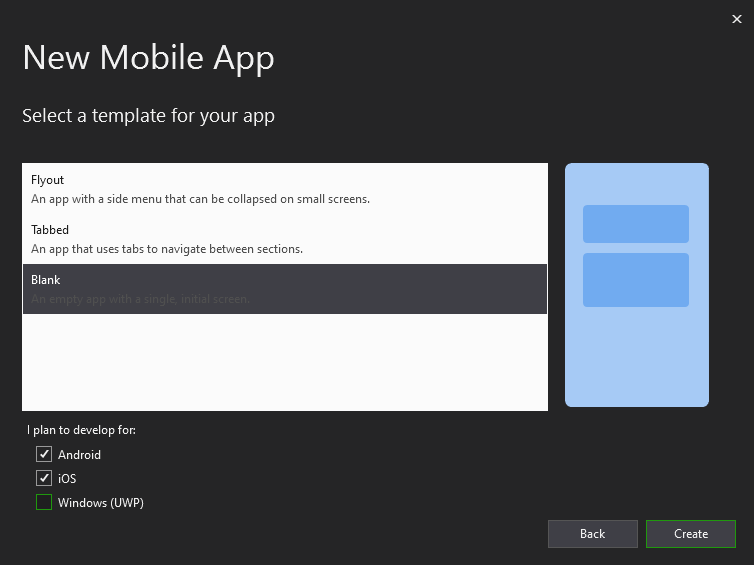


Figure 42: Template project

Step 5: Finish

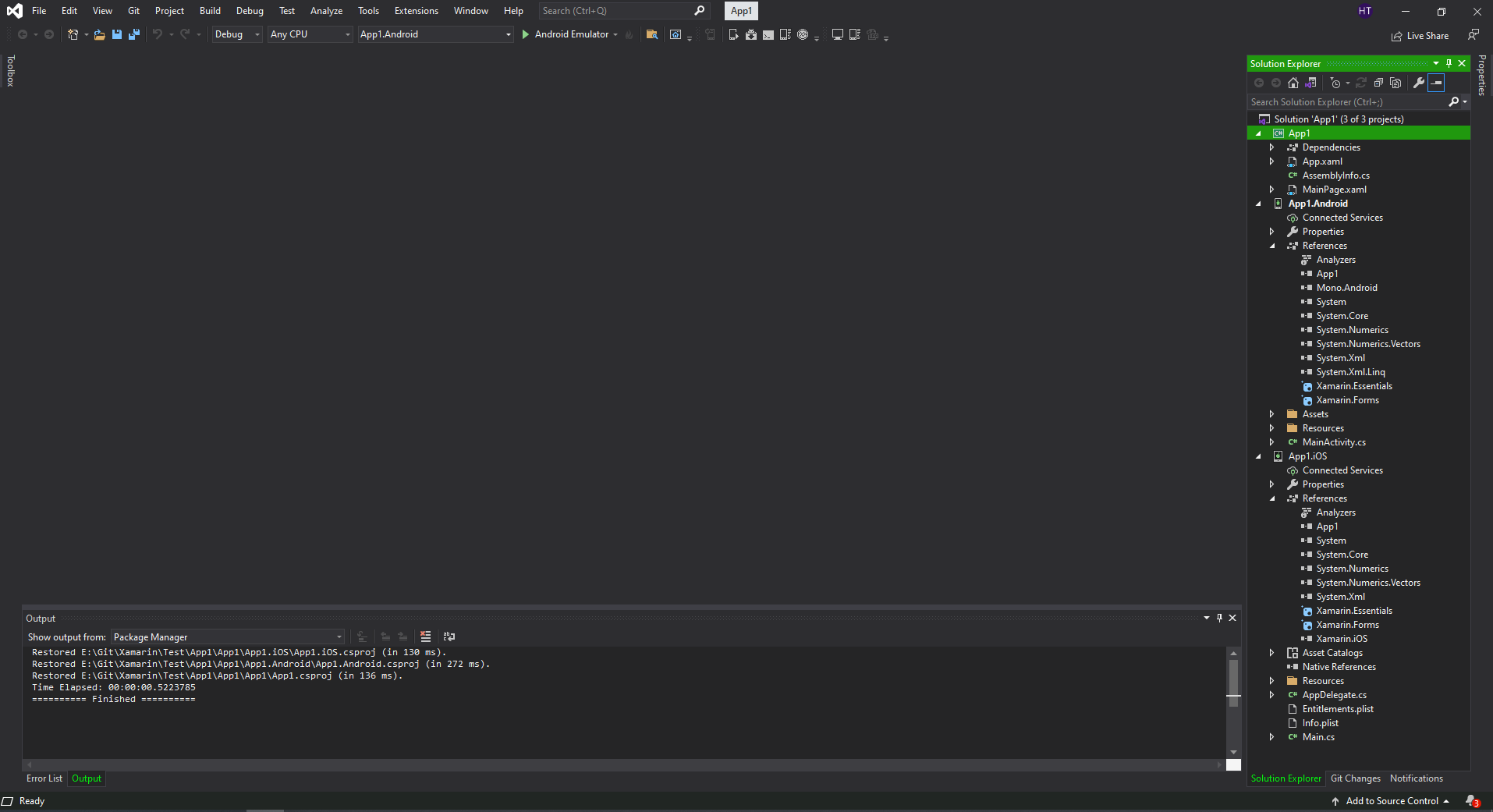


Figure 43: Create project success

3.4.2 Build pattern MVVM

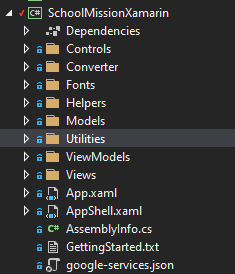


Figure 44: Pattern MVVM

Creating these folders is the Model - View - ViewModel architectural pattern that is widely used today. This makes it possible for application creation and processing to separate the application aspects (input logic, task logic, and interface logic). The MVVM model is used in any project because of its versatility.

In which the interface (UI logic) belongs to the View, the part that handles the user's operations and interactions is the ViewModel. The task logic ((Business logic – is information processing logic, the main purpose of the application) belongs to the model. This division helps you to reduce the complexity of the application and focus only on each aspect. need to be installed at each time.

3.4.3 Implementation of MVVM

After creating the MVVM model template, we need to make the program understand and it execute correctly according to the steps that have been performed according to the MVVM model.

First, we have to create a HomePage.xaml file in the Views folder. This file will contain the display interface so the user's screen. Then, in the ViewModel folder, we create a file with the same name as HomeViewModel.cs, this file has the effect of handling user operations to process data..

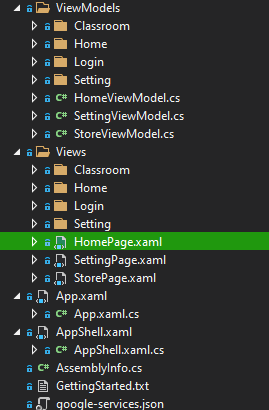


Figure 45: Create file in pattern MVVM

Next, in the HomePage.xaml file, we add the code below to the interface. Intent llafBindingContext from View down to ViewModel. As a bridge between View and ViewModel.

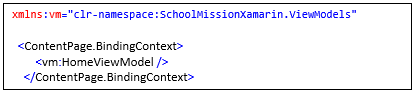


Figure 46: Binding context code

Next, in the file HomeViewModel.cs will inherit a class with specific content as below. The goal is that all ViewModels will have the same managed structure and more convenient data handling, time saving, and shorter code..

Contents, BaseViewModel.cs will have:

* Each page will have a title and Busy to load
* Every page will have to CRUD to the database (firebase)
* Each page will have fields that will change data continuously, so Notification will be required to update the interface immediately.
* Each page is the child Navigation of Page Parent

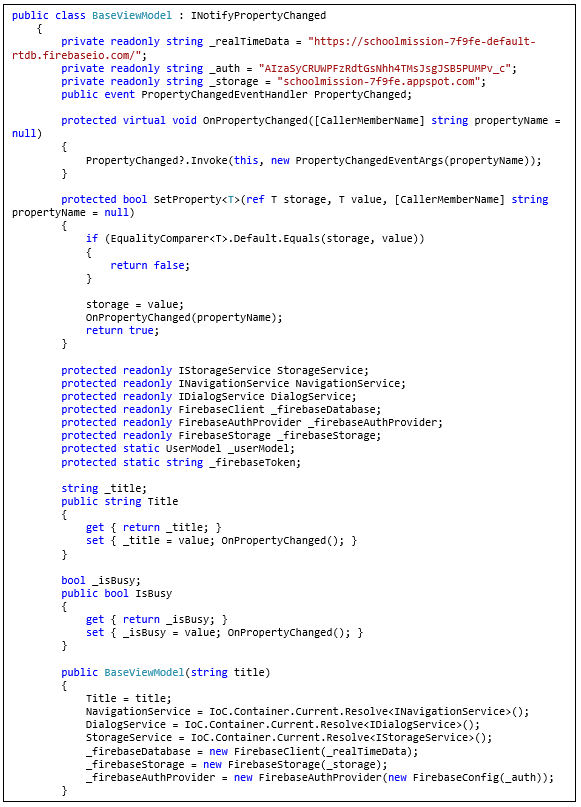


Figure 47: BaseViewModel codebehind

Below is the general code of the files in the ViewModel



Figure 48: Example HomeViewModel

So how does data Binding happen?

This is the interface code that displays the number of Points the user owns. Attention to the Text property. It is Binding to the variable CurrentUser.Point

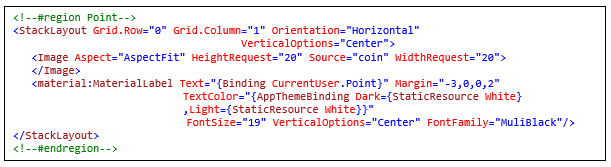


Figure 49: Binding Text property

So inside the ViewModel ie the file HomeViewModel.cs will have to have a variable named CurrentUser. This variable has a data type of UserModel and in the property of this class there is a field of Point.

In addition, in order for the data to be updated immediately, the interface must have a function Setproperty.

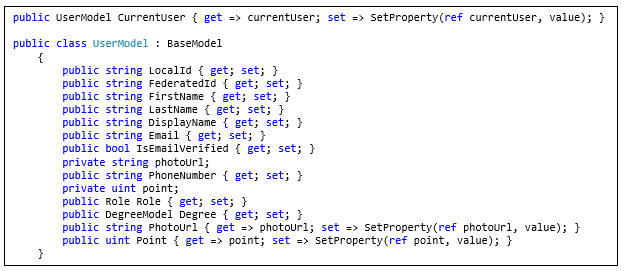


Figure 50: Usermodel and Notification property data change

3.5 Examples of some functions

3.5.1 Function login

Step 1: Create login screen interface

Login screen interface with some main fields: Username, password, forgot password, login

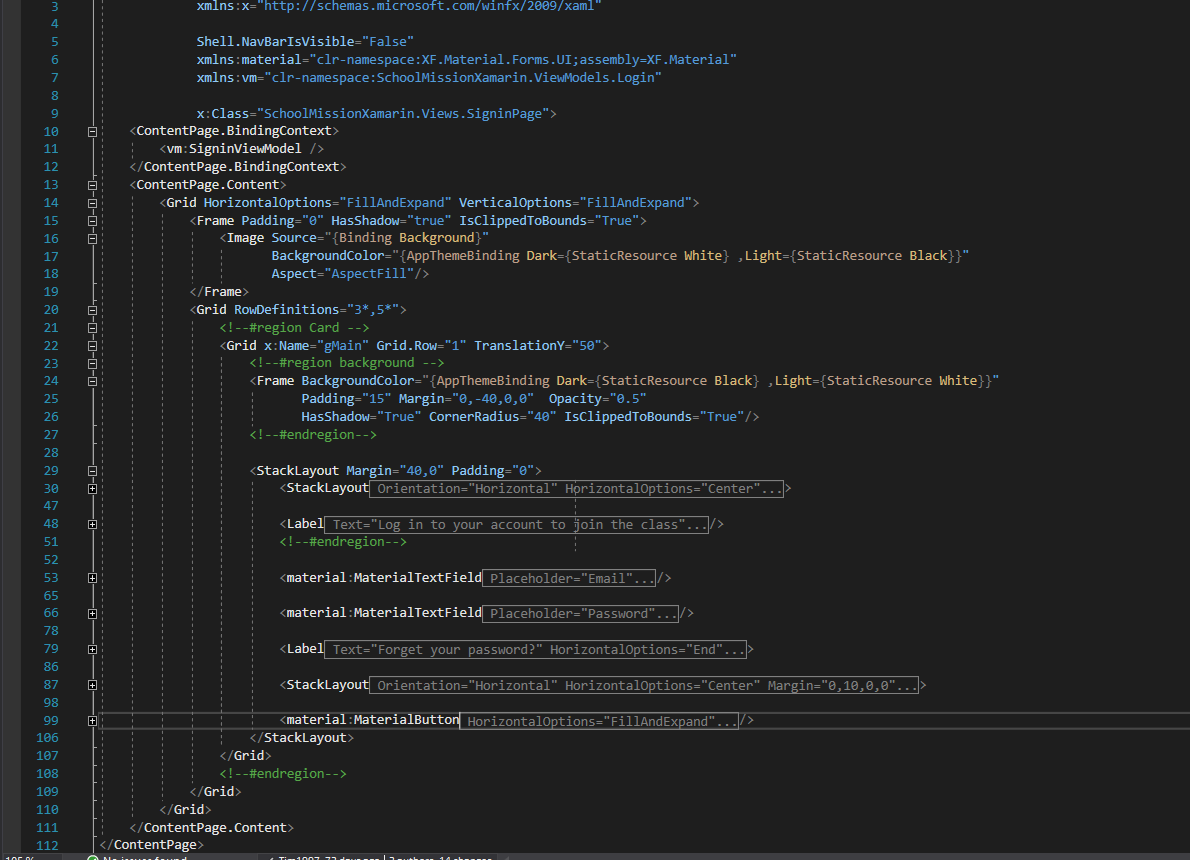


Figure 51: Code xaml UI Login

Step 2: Next comes the data processing part. In SignInViewModel.cs will include all the Properties that need Binding data, so Views are as below.

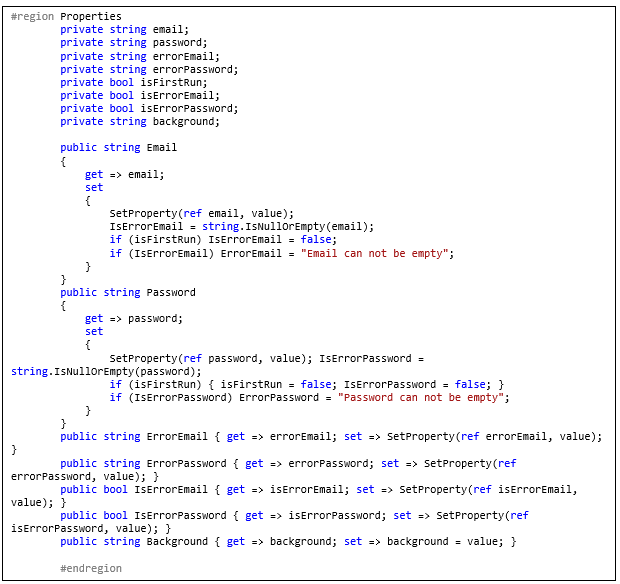


Figure 52: LoginViewModel code behind

Step 3: Next, we need to initialize basic values for some properties that need to display data like Email will be loaded from data stored in the App or have to set the background for the app through checking each time frame..

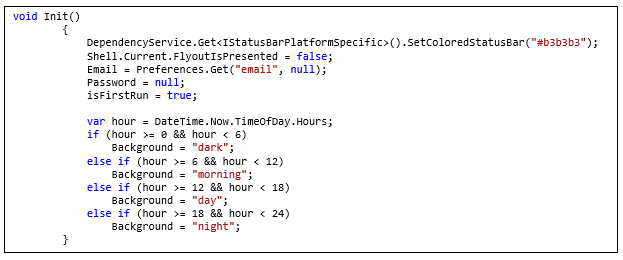


Figure 53: Init method in contructor

Step 4: Next, when the user has entered the Username and Password, they will click the Login button. Then the ViewModel will call the Command LoginCommand. This function will check the necessary conditions before sending the query to the database, if false, it will notify the user. If all checks are done, we will query the server and return the data for processing.

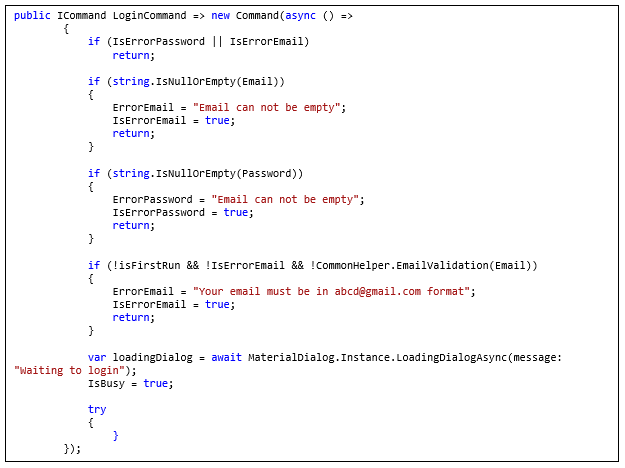


Figure 54: handle command login

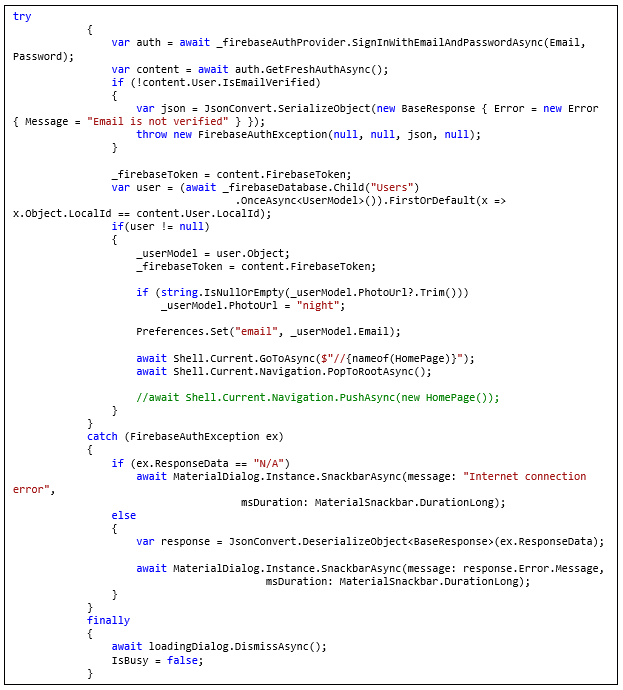


Figure 55: Request data to firebase server and get response

Step 5: As a result, we will have the following interface

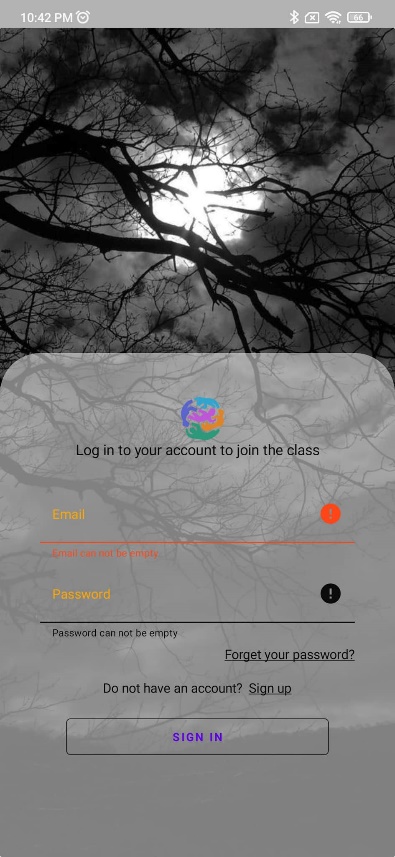
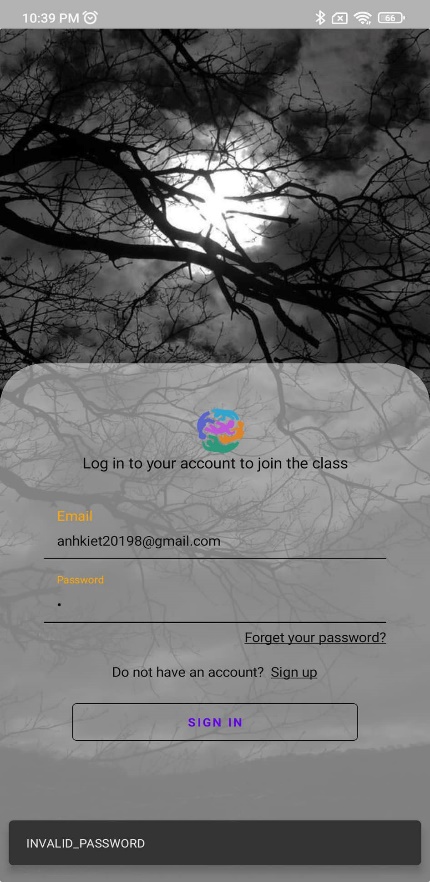
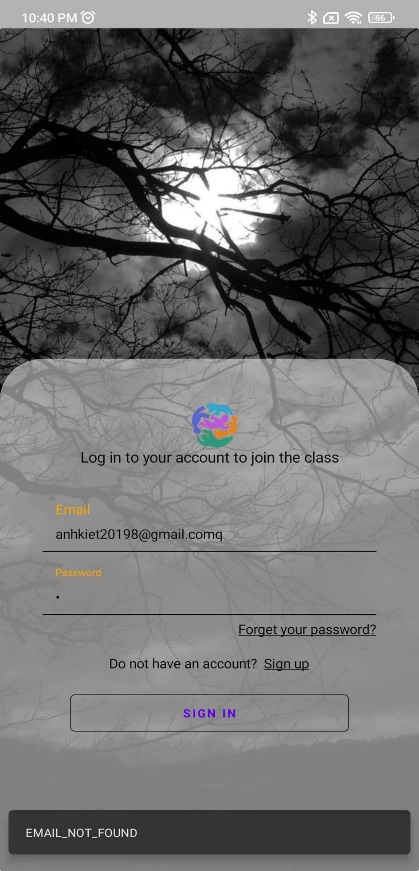
  

Figure 56: Login UI Figure 57: Wrong password Figure 58: Wrong email

*Hình STYLEREF 1 \s 3. SEQ Hình \\* ARABIC \s 1 43: Màn hình Login trên IOS*

*Hình STYLEREF 1 \s 3. SEQ Hình \\* ARABIC \s 1 42:Màn hình Login trên Android*

# BIBLIOGRAPHY

|  |  |
| --- | --- |
| [1] | T. Contributor, "techtarget," [Online]. Available: https://www.techtarget.com/whatis/definition/Model-View-ViewModel. |
| [2] | "w3schools," [Online]. Available: https://www.w3schools.com/cs/cs\_oop.php. |
| [3] | E. Team, "educative," [Online]. Available: https://www.educative.io/edpresso/what-is-firebase. |
| [4] | "britannica," [Online]. Available: https://www.britannica.com/technology/Internet. |
| [5] | "microsoft," [Online]. Available: https://docs.microsoft.com/en-us/xamarin. |
| [6] | "mongodb," [Online]. Available: https://www.mongodb.com/nosql-explained. |
| [7] | "sharpcorner," [Online]. Available: https://www.c-sharpcorner.com/article/xamarin-forms-working-with-firebase-realtime-database-crud-operations/. |

# SUMMARY

**General conclusions**

After the research, analysis, design and guidance of Ms. XXX. Online learning application (ILearn) has been completed on both Android and iOS platforms and can be Public on the Store.

The application has fully achieved the requirements of students, students and teachers from interface to function. Stable operation (Depends on network speed).

Improve teamwork ability, the connection between members in the project. Exposure to new technology, new background has brought a lot of expertise.

Get direct exposure to real projects. How to think systems, how to develop systems in stages. Working Tester improves work performance and personal skills.

However, due to limited knowledge and unfamiliarity with new technologies, it takes more time to learn and practice. Need a lot of help Miss XXX. Code layout is still ugly and unsystematic.

**Orientation**

It is very easy to extend and develop other functions of the system.

This application is just a project. It is possible to develop other projects.

Improve your coding skills and more efficient and scalable systems thinking.