

EASTERN INTERNATIONAL UNIVERSITY
SCHOOL OF COMPUTING AND INFORMATION TECHNOLOGY
DEPARTMENT OF SOFTWARE ENGINEERING



PROJECT B

**ELA ENGLISH LANGUAGE CENTER
MANAGEMENT**

Students

Nguyen Gia Bao - 1831200023

Supervisor

Tat Quang Phat, M.Sc.

Binh Duong, 13th June 2022

ABSTRACT

ELA English Language Center Management exists mainly for the purpose of supporting English Center's staffs and students with paperwork, simplifying the process of accessing students' information. It was made under the intention of catching up to the current technologically developing trend, which has proven to be quite useful and convenient for various situations. Upon using this app, the center's student body will have access to distinctive, functional features such as homework, schedules for upcoming tests, previous tests' scores and their class's information. For staff, the app serves as an effective management tool, as it allows them to easily find specific students' personal information, therefore speeding up their work process.

ACKNOWLEDGEMENT

The success and outcome of this project required a lot of guidance and assistance from many and I am extremely privileged to have got this all along with the completion of our project. All that I have done is only due to such supervision and assistance.

I would like to express our deepest appreciation to our teacher, Mr.Tat Quang Phat, whose contribution to stimulating suggestions and encouragement, helped us to coordinate my project and complete it duly. I am extremely thankful to him for providing nice support and guidance, although he had a busy schedule.

Besides, I would also like to thank our greatest university which supports me with office and equipment to implement our project. It is a source of extremely valuable assistance.

ABSTRACT	1
ACKNOWLEDGEMENT	2
CHAPTER I: INTRODUCTION	5
Motivation	5
Objectives	5
Project Overview	5
CHAPTER II: TECHNOLOGIES	6
Android studio	6
Firebase	6
Java	6
CHAPTER III: SUBJECT ANALYSIS	8
Topic Analyzation	8
Database Design	8
Connect firebase and Realtime database with android	8
Features	12
Login interface	12
CHAPTER IV: CONCLUSION AND FUTURE WORKS	23
Result	23
Drawback	23
Future works	23
REFERENCES	24

List of Figures

Figure 1. Connect firebase and Realtime database with android

Figure 2. Realtime database of question

Figure 3. Login interface

Figure 4. Create account interface

Figure 5. Forgot password interface

Figure 6. Main interface

Figure 7. Search interface - no input

Figure 8. Decimal - Search interface - with input

Figure 9. Add new student interface

Figure 10. Add new student interface - missing/wrong input

Figure 11. Create new class interface

Figure 12. Search for class interface - no input

Figure 13. Search for class interface - with input

Figure 14. Student profile interface

Figure 15. Class information interface

List of Tables

Table 1. Database of students

Table 2. Database of accounts

Table 3. Database of classes

CHAPTER I: INTRODUCTION

1.1. Motivation

In modern day, English has become the international language. Learning English is the most common activity that not only students but also adults take part in frequently. In Viet Nam, the number of English centers has grown rapidly over the last 10 years. However, the digitization of the management process of language centers is yet to be applied widely.

1.2. Objectives

With this project - ELA English Language Center Management app, I wish to help to simplify the management of these centers and create a way of interaction between students and English centers outside of study time. This will also help to reduce the use of papers and reach out to potential students since mobile applications are very popular.

1.3. Project Overview

ELA English Language Center Management is an android application written with Android Studio. This app uses Firebase realtime database to manage databases. The app will provide a way for staff members to register new students and create new classes, manage their information, upload assignments and send notifications to users. For normal users, the app offers methods for viewing classes' information and their courses' results, do assignments and view learning materials provided by the center. All of these are done online via the app.

CHAPTER II: TECHNOLOGIES

2.1 Android studio

Android Studio is the official IDE for Android application development, based on IntelliJ IDEA. On top of the capabilities you expect from IntelliJ, Android Studio offers:

- Flexible Gradle-based build system
- Build variants and multiple apk file generation
- Code templates to help you build common app features
- Rich layout editor with support for drag and drop theme editing
- Lint tools to catch performance, usability, version compatibility, and other problems
- ProGuard and app-signing capabilities
- Built-in support for Google Cloud Platform, making it easy to integrate Google Cloud Messaging and App Engine

2.2 Firebase

Firebase is a toolset to “build, improve, and grow your app”, and the tools it gives you cover a large portion of the services that developers would normally have to build themselves, but don’t really want to build, because they’d rather be focusing on the app experience itself. This includes things like analytics, authentication, databases, configuration, file storage, push messaging, and the list goes on. The services are hosted in the cloud, and scale with little to no effort on the part of the developer.[1]

2.3 Java

Java is a programming language and a platform. Java is a high level, robust, object-oriented, and secure programming language.

Java was developed by Sun Microsystems (which is now the subsidiary of Oracle) in the year 1995. James Gosling is known as the father of Java. Before Java, its name was Oak.

Platform: Any hardware or software environment in which a program runs, is known as a platform. Since Java has a runtime environment (JRE) and API, it is called a platform.[2]

CHAPTER III: SUBJECT ANALYSIS

3.1 Topic Analyzation

ELA Language Center Management app:

- Users need to log in with their accounts
- Build on the following purposes:
 - Create and manage accounts
 - Register students and manage students
 - Create and manage classes
 - Upload and manage assignment
 - Save account, student, assignment and class to database

3.2 Database Design

3.2.1 Connect firebase and Realtime database with android

Connect firebase: Open android studio -> Tools -> Firebase -> Realtime database -> Get started with Realtime Database -> Connect to Firebase.

Connect Realtime database: Open android studio -> Tools -> Firebase -> Realtime database -> Get started with Realtime Database -> Add the Real Time Database SDK to your app.

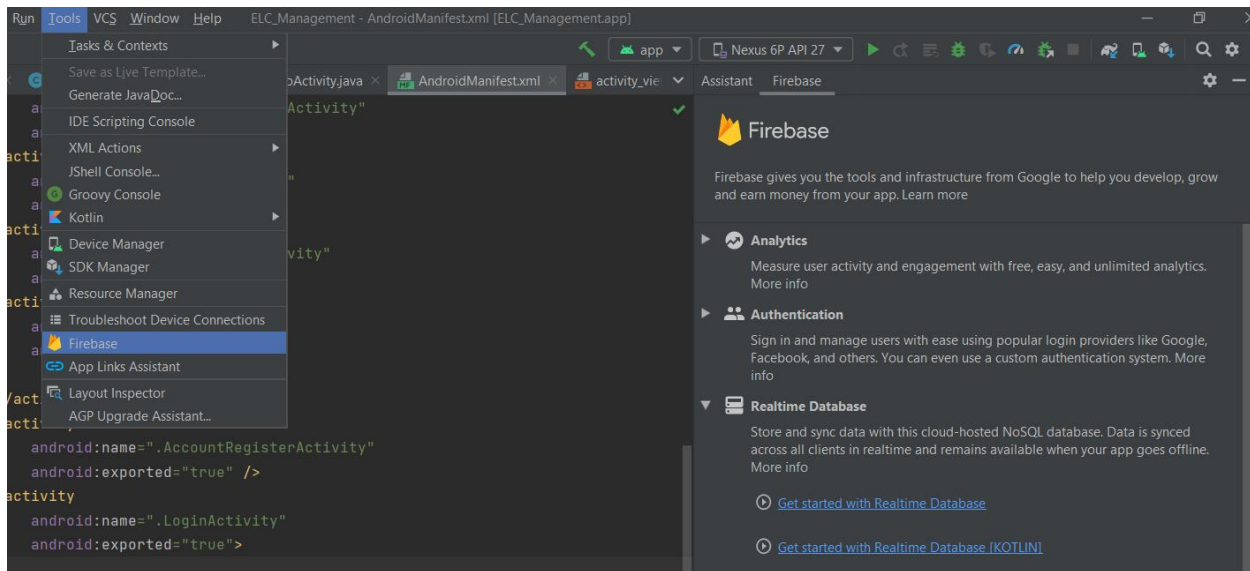


Figure 1. Connect firebase and Realtime database with android

3.2.2 Realtime database



Figure 2. Realtime database of question

3.2.3 Detailed database tables

Data Name	Data Type	Description
cId	STRING	The id of the class
schedule	STRING	The start/end date of the class
time	STRING	The time when the class takes place
status	STRING	The class is ongoing or finished
quantity	INT	The number of students in the class

Table 1. Database of class

Data Name	Data Type	Description
sId	STRING	Student's id
name	STRING	Student's name
gender	STRING	Student's gender
sClass	STRING	Student's class
phoneNumber	STRING	Student's registered phone number
email	STRING	Student's email

Table 2. Database of student

Data Name	Data Type	Description
aId	STRING	Account's id
aUsername	STRING	Account's username
aPassword	STRING	Account's password
aPhoneNumber	STRING	Account's registered phone number (must be the same as the student's registered phone number)
aEmail	STRING	Account's registered email
aNickname	STRING	Account's nickname

Table 3. Database of account

3.3 Features

3.3.1 Login interface

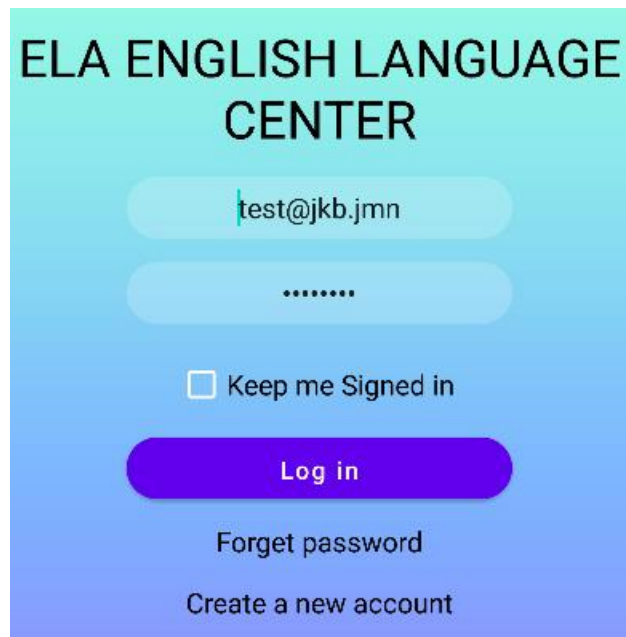
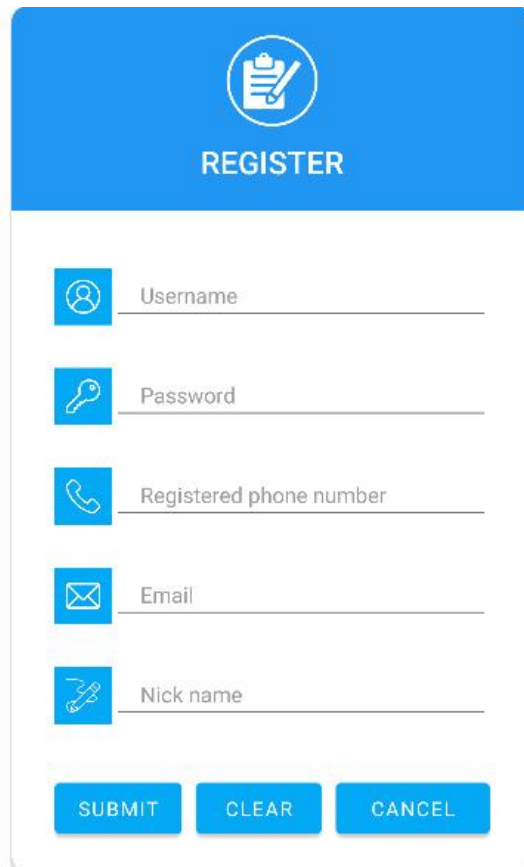


Figure 3. Login interface

When user clicks "Create a new account", the app will switch to "Create account interface"

3.3.2 Create account interface.

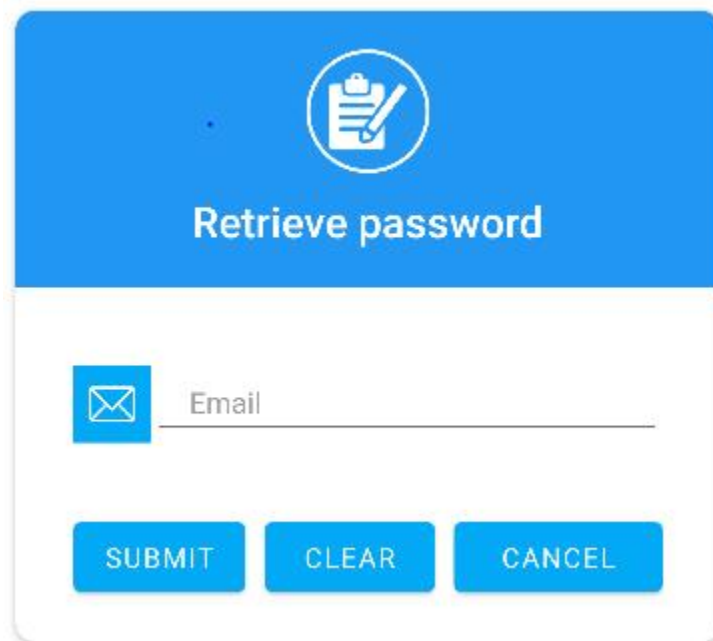


The image shows a mobile application interface for creating an account. It features a blue header with a white icon of a clipboard and a pencil, and the word "REGISTER" in white capital letters. Below the header, there are five input fields, each with a blue icon on the left and a text label to the right of the input line. The fields are: "Username" (person icon), "Password" (key icon), "Registered phone number" (phone icon), "Email" (envelope icon), and "Nick name" (pencil icon). At the bottom of the form, there are three blue buttons with white text: "SUBMIT", "CLEAR", and "CANCEL".

Figure 4. Create account interface

When the user clicks on the submit button, the app will validate the inputs. If all conditions are met, the account will be created and saved to the database.

3.3.3 Forgot password interface



The image shows a 'Retrieve password' interface. It has a blue header with a white icon of a clipboard and a pencil, and the text 'Retrieve password'. Below the header is a white form area. On the left of the form is a blue square icon with a white envelope. To its right is a text input field with the placeholder text 'Email'. Below the input field are three blue buttons with white text: 'SUBMIT', 'CLEAR', and 'CANCEL'.

Figure 5. Forgot password interface

Enter registered email address and an email containing reset password link will be sent to the provided email address.

3.3.3 Main interface

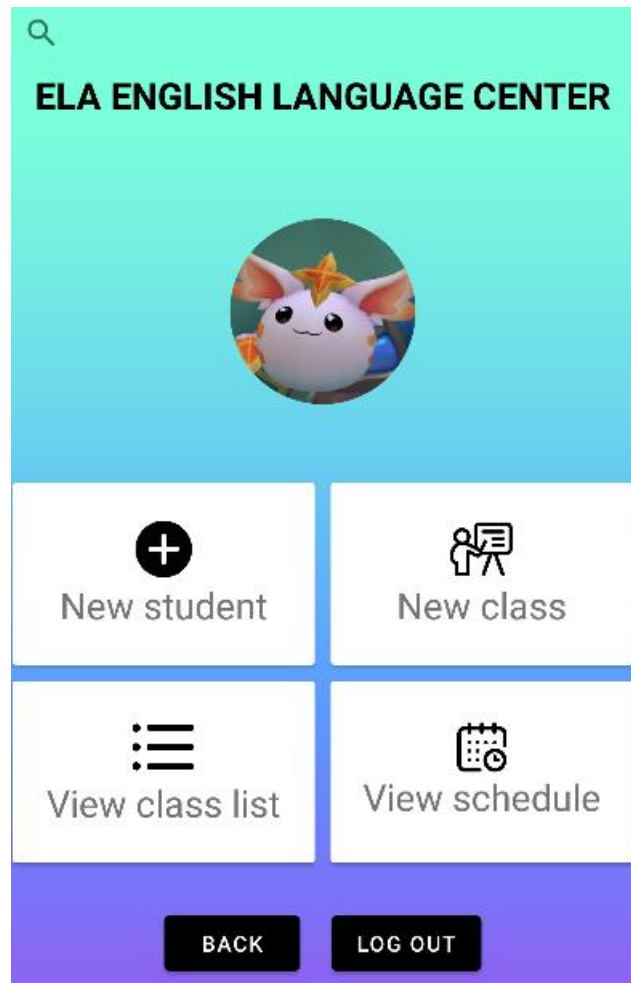



Figure 6. Main interface

From here users can access and use other functions of the app such as: add new student, create new class, view class lists, view schedules and search for specific students/classes.

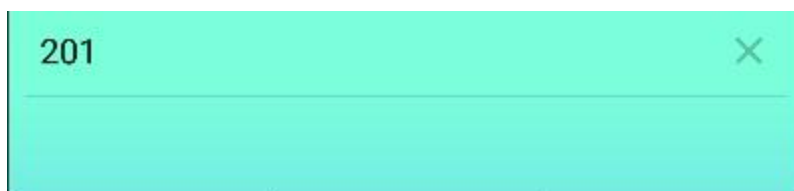
When users have not entered anything to the search bar, the interface shows all the students.



Class	Name	Phone number
Pre102	Bao	1478523690
11	hilde	1478523693
Beg201	Tu	7412589630
Beg201	Thao	7412589631
Beg201	Tra	5462871390
Beg301	test	1478523699
Beg102	Thi	1478523694

Figure 7. Search interface - no input

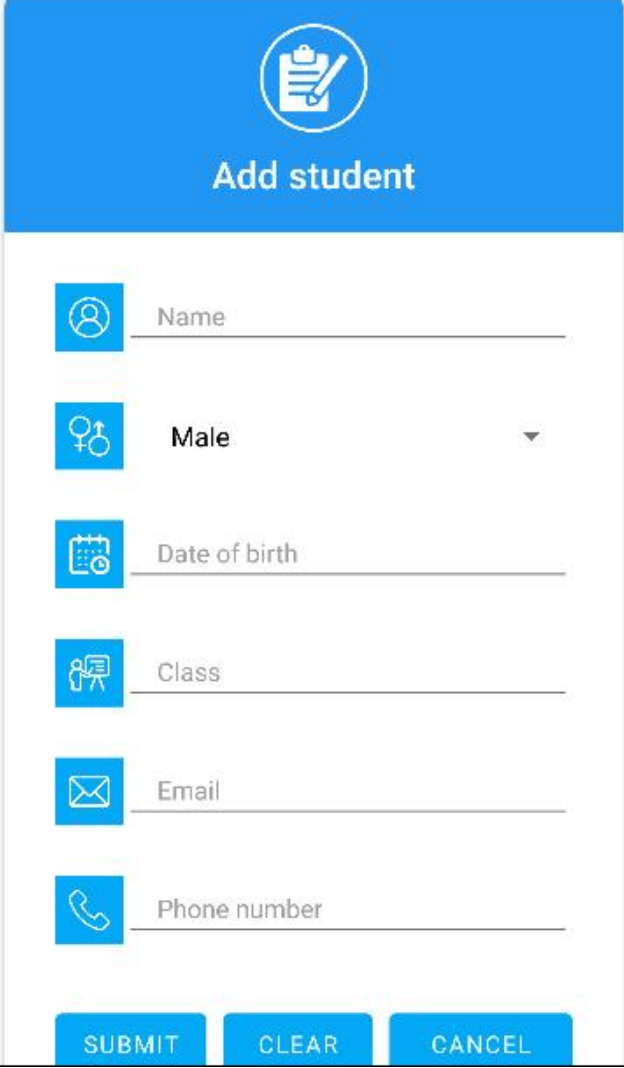
If users enter input the interface shows only the results which match the conditions. Currently, the conditions are based on matching a student's name and/or classes.



Class	Name	Phone number
Beg201	Tu	7412589630
Beg201	Thao	7412589631
Beg201	Tra	5462871390

Figure 8. Decimal - Search interface - with input

3.3.4 Add new student interface



The image shows a mobile application interface for adding a new student. It features a blue header with a white icon of a clipboard and pencil, and the text "Add student". Below the header, there are six input fields, each with a blue icon on the left and a text label on the right: "Name" (person icon), "Male" (gender icon, with a dropdown arrow), "Date of birth" (calendar icon), "Class" (classroom icon), "Email" (envelope icon), and "Phone number" (phone icon). At the bottom, there are three blue buttons labeled "SUBMIT", "CLEAR", and "CANCEL".

Figure 9. Add new student interface

Users must provide all the above information in order to create and add a student to the database.

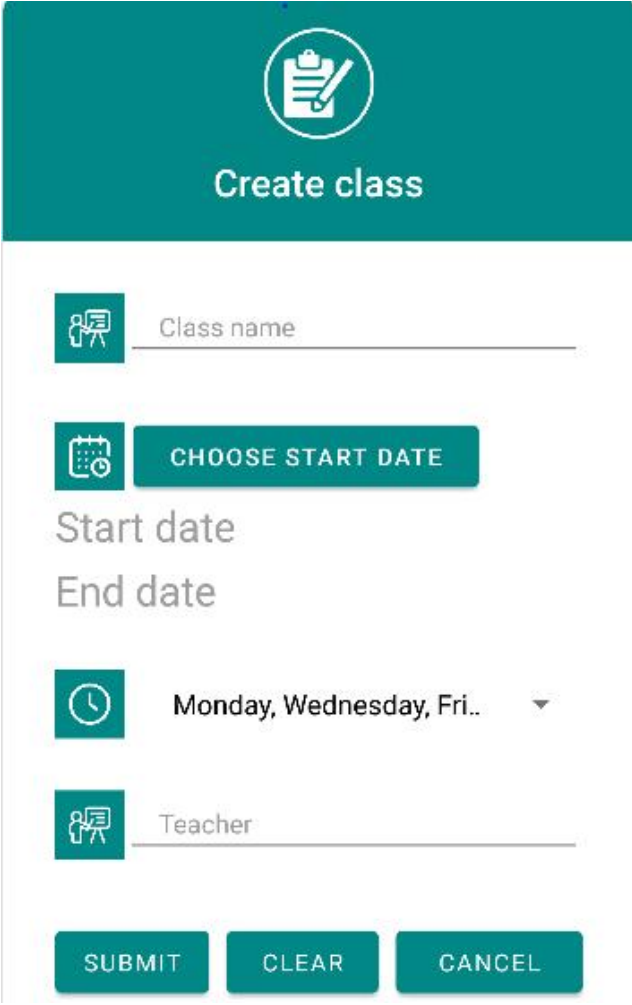
The screenshot shows a mobile application interface for adding a student. At the top is a blue header with the text "Add student". Below the header are six input fields, each with a blue icon on the left and a red exclamation mark on the right, indicating a validation error. The fields are: "Name" (with a person icon), "Male" (with a male symbol icon), "Date of birth" (with a calendar icon), "Class" (with a classroom icon), "Email" (with an envelope icon), and "Phone number" (with a phone icon). A black tooltip with the text "Required" is positioned over the "Male" field. The "Name" field has a red underline. The "Date of birth" field has a red underline. The "Class" field has a red underline. The "Email" field has a red underline. The "Phone number" field has a red underline.

Figure 10. Add new student interface - missing/wrong input

If users leave a field blank or fill in wrong input according to the conditions, the app will notify with the red warning and show the type of errors. Currently, the conditions are:

- All field are required
- Name contains only letters
- Phone number must be at least 10 numbers

3.3.5 Create new class interface

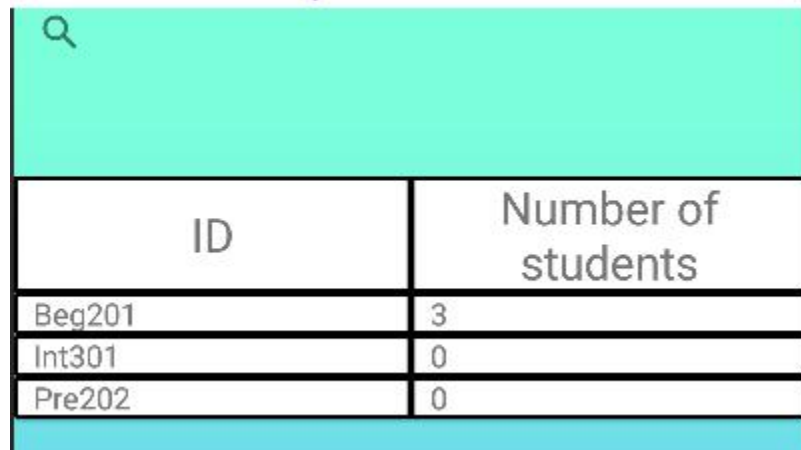


The interface for creating a new class is displayed within a teal header bar. The header bar contains a white circular icon with a clipboard and pencil, and the text "Create class" in white. Below the header bar, the form is organized into several sections. The first section is a text input field with a teal icon of two people and a speech bubble, followed by the placeholder text "Class name". The second section features a teal icon of a calendar, a teal button labeled "CHOOSE START DATE", and two text input fields labeled "Start date" and "End date". The third section includes a teal icon of a clock, a text input field with the placeholder "Monday, Wednesday, Fri..", and a small downward arrow. The fourth section has a teal icon of two people and a speech bubble, followed by a text input field labeled "Teacher". At the bottom of the form, there are three teal buttons labeled "SUBMIT", "CLEAR", and "CANCEL".

Figure 11. Create new class interface

In this, users must provide all the information except for the teacher field (can be assigned later) to create a new class. A class's duration is 3-month long by default.

3.3.6 Search for class interface

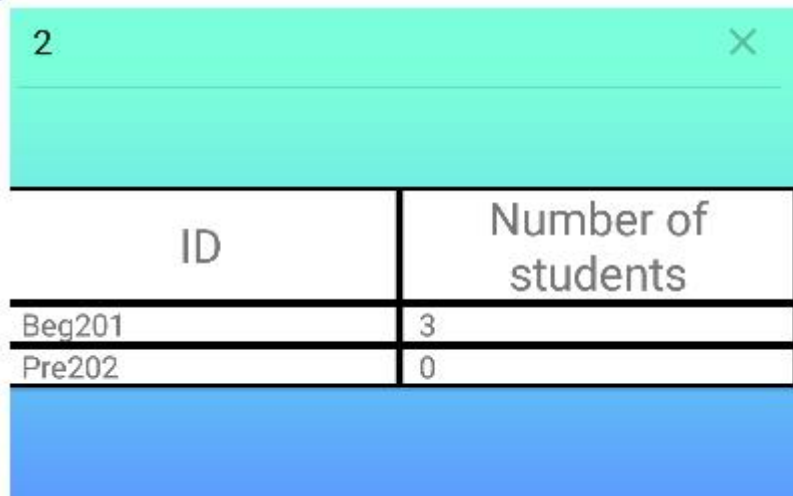


The screenshot shows a search interface with a light blue header bar containing a magnifying glass icon. Below the header is a table with two columns: 'ID' and 'Number of students'. The table lists three classes: Beg201, Int301, and Pre202. The 'Number of students' column shows values 3, 0, and 0 respectively. The table is set against a white background with black borders.

ID	Number of students
Beg201	3
Int301	0
Pre202	0

Figure 12. Search for class interface - no input

When the user clicks on “View class list” from the main interface, the app will direct the user to this interface. With no input, the interface will show all the existing classes.



The screenshot shows the same search interface as Figure 12, but with a search input field at the top. The input field is light blue and contains the number '2'. A close button (X) is visible in the top right corner of the input field. The table below shows the results of the search, displaying only the classes that match the input: Beg201 and Pre202. The 'Number of students' column shows values 3 and 0 respectively. The table is set against a white background with black borders.

ID	Number of students
Beg201	3
Pre202	0

Figure 13. Search for class interface - with input

When the user provides a key word, the interface will look up and show the appropriate results. Currently, the search condition is based on matching class's ID characters.

3.3.7 Student profile interface



The image shows a mobile application interface for a student profile. At the top, there is a black header with the text "Student Profile" in white. Below the header is a large light blue rectangular area. In the center of this area is a circular profile picture of a pink and white cartoon creature with orange star-shaped ears. Below the profile picture is a form with several rows, each with a label on the left and a text input field on the right. The rows are: Name (Tu), Gender (Female), Dob (11), Class (Beg201), Phone (7412589630), and Email (tu@glm.glm). At the bottom of the form are three buttons: SUBMIT, DELETE, and BACK.

Field	Value
Name	Tu
Gender	Female
Dob	11
Class	Beg201
Phone	7412589630
Email	tu@glm.glm

SUBMIT DELETE BACK

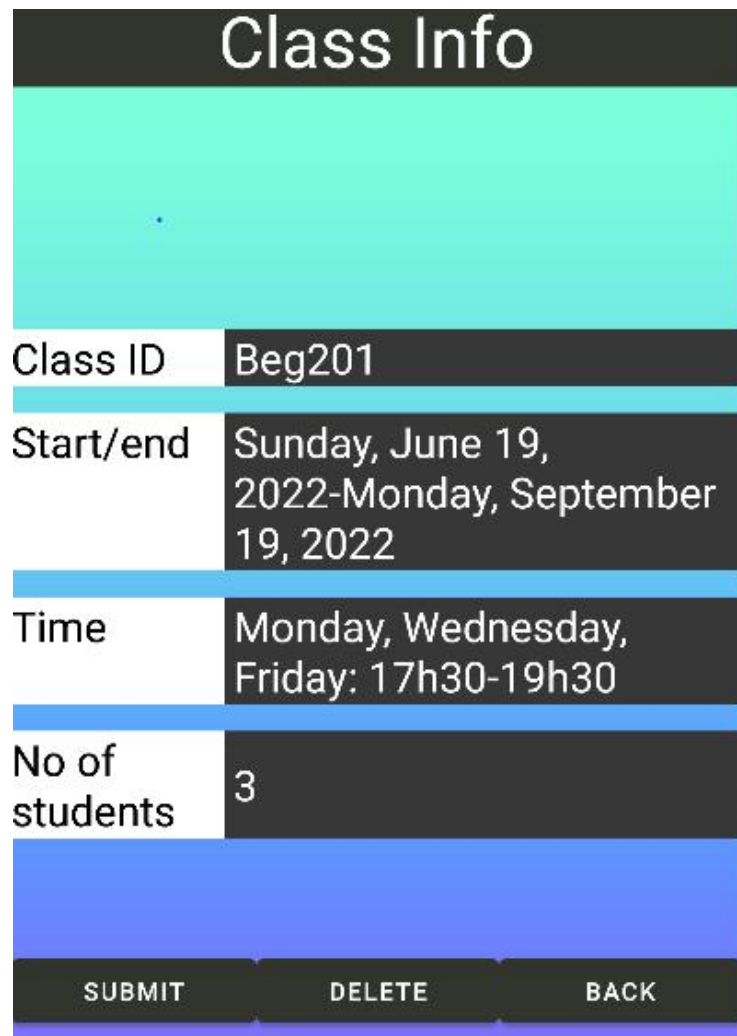
Figure 14. Student profile interface

When the user clicks on a student from the student list, the app will show the student profile interface, which contains the corresponding student profile such as: name, gender, date of birth, class, phone number, email.

Users can click on a field and edit it to change the student information and click on the “submit” button to commit the change.

Users can also delete a student by clicking on the “delete” button to remove a student from the database.

3.3.8 Class information interface



The image shows a web interface titled "Class Info". It features a large cyan rectangular area at the top. Below this, there are four rows of information, each with a label on the left and a value on the right, separated by a vertical line. The labels are "Class ID", "Start/end", "Time", and "No of students". The values are "Beg201", "Sunday, June 19, 2022-Monday, September 19, 2022", "Monday, Wednesday, Friday: 17h30-19h30", and "3" respectively. At the bottom of the interface, there is a blue bar containing three buttons: "SUBMIT", "DELETE", and "BACK".

Class Info	
Class ID	Beg201
Start/end	Sunday, June 19, 2022-Monday, September 19, 2022
Time	Monday, Wednesday, Friday: 17h30-19h30
No of students	3
<div>SUBMITDELETEBACK</div>	

Figure 15. Class information interface

When users click on a class from the “View class list” interface, the program shows the class information interface, which contains information about the class such as: class id, start/end date, class occupied time and the number of students in the class.

Same for student profiles, users can edit the class information (except for number of students) by clicking on the fields and clicking the submit button to commit the change.

Users can also delete a class in the same way as deleting a student from the database.

CHAPTER IV: CONCLUSION AND FUTURE WORKS

4.1 Result

After the implementation process, ELA English Language Center has applied the following technologies:

- Firebase: Database management system.
- Realtime database: Real-time tuning database system

The application helps users manage the students and classes online, simplifying the process of manual labor.

4.2 Drawback

The project is still not finished: some main functions are not yet finished, contains some bugs, and is facing a synchronization problem between the database and encryption. Furthermore, the project does not fully apply any main design pattern to make the code cleaner and easier to maintain. Besides, the UI of the app is still too simple, along with the lack of notification interfaces.

4.3 Future works

In the future, I will develop ELA English Language Center Management with more functions to meet the needs and improve the quality of user service, specifically I will focus on:

- Complete database.
- Add sound to the app.
- Add more notification interfaces.
- Develop function to upload assignments.
- Application user interface improvements.
- Security.
- Developed interfaces for non-staff users.
- Fully apply authentication and authorization.
- Fully apply appropriate design patterns.

REFERENCES

- [1] Firebase helps you build and run successful apps: <https://firebase.google.com/>
- [2] JavaTpoint, “Java”: <https://www.javatpoint.com/java-tutorial/>
- [3] Android studio documentation: <https://developer.android.com/docs>
- [4] Public platform to find and contribute answers to technical challenges:
<https://stackoverflow.com/>
- [5] Java Programming for Android Developers For Dummies, 2nd Edition [For Dummies] -
Barry Burd