VIETNAM GENERAL CONFEDERATION OF LABOR

**TON DUC THANG UNIVERSITY**

**FACULTY OF INFORMATION TECHNOLOGY**

****

**Nguyen Lam Duy - 521H0499**

**Ly Manh Phi - 521H0286**

**MIDTERM REPORT**

**FIREBASE FIRESTORE FOR DEVELOPING A REALTIME APP FOR STUDENT INFORMATION MANAGEMENT**

**COMPUTER SCIENCE**

**HO CHI MINH CITY, 2023**

VIETNAM GENERAL CONFEDERATION OF LABOR

**TON DUC THANG UNIVERSITY**

**FACULTY OF INFORMATION TECHNOLOGY**

****

**Nguyen Lam Duy - 521H0499**

**Ly Manh Phi - 521H0286**

**MIDTERM REPORT**

**FIREBASE FIRESTORE FOR DEVELOPING A REALTIME APP FOR STUDENT INFORMATION MANAGEMENT**

**COMPUTER SCIENCE**

Instructor

**TS. Mai Van Manh**

**HO CHI MINH CITY, 2023**

**Acknowledgement**

We want to express our sincere gratitude to Mr. Mai Van Manh for his exceptional teaching in Web Programming with Node.js and Mobile Application Development this semester. Despite the short duration, your clear explanations and passion for the subject greatly improved our understanding. Your availability for support created a positive learning environment, fostering collaboration among classmates.

As we approach the midterm essay, we're applying the foundation you've provided. We value your feedback for further improvement. Thank you, Mr. Mai Van Manh, for your dedication. We look forward to continuing our learning journey in your future courses.

*Ho Chi Minh City, day 25 month 11 year 2023*

*Author*

*(Sign and write full name)*



Ly Manh Phi

**INSTRUCTOR EVALUATION FORM**

Name of instructor:

Comments:

Total score according to rubrik assessment:

*Ho Chi Minh City, day month year 2023*

*Instructor guides*

*(Sign and write full name)*

**THE WORK IS COMPLETED AT TON DUC THANG UNIVERSITY**

Our team would like to assure that this is our own research project and is under the scientific guidance of MSc. Mai Van Manh. The research content and results in this topic are honest and have not been published in any form before. The data in the tables for analysis, comments, and evaluation were collected by the author from different sources and clearly stated in the reference section.

In addition, the report also uses several comments, assessments as well as data from other authors and other organizations, all with citations and source notes.

**If any fraud is detected, our team will take full responsibility for the content of our Essay Midterm.** Ton Duc Thang University is not involved in copyright violations caused by us during the implementation process (if any).

*Ho Chi Minh City, day 25 month 11 year 2023*

*Author*

*(Sign and write full name)*



Ly Manh Phi

**Tóm tắt/Abstract**

**Table of contents**

**List of images (if any)**

**List of symbols and abbreviations (if any)**

**CONTENT**

# I. INTRODUCTION

The Firebase platform provides two types of NoSQL databases, Realtime Database and Cloud Firestore, which let users store cloud databases and integrate synchronized data into mobile applications in real time.

## What is Firebase Realtime Database?

Firebase Realtime Database is a cloud-hosted NoSQL database that allows you to store and synchronize data. Data is stored in JSON format and is synchronized in real-time across all connected devices.

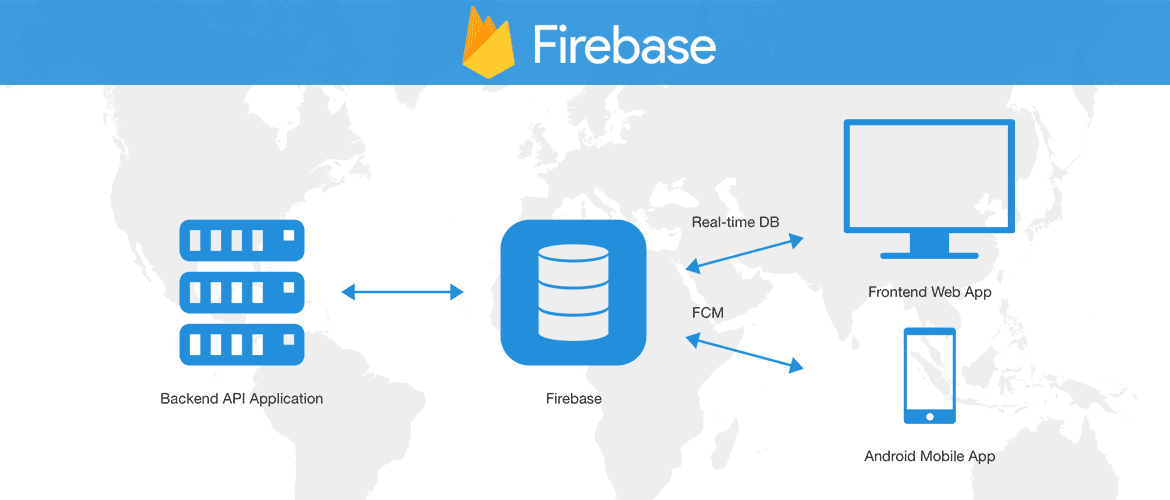
Using the iOS, Android, and JavaScript SDKs for cross-platform application development, all of your clients share a single Realtime Database instance and are automatically updated with the most recent information. Therefore, this is a low-latency and efficient option for mobile apps that need to synchronize data in real-time between devices.

Figure 1. Firebase Realtime Database

## What is Cloud Firestore?

Cloud Firestore is Firebase's new database designed for mobile applications. It inherits the legacy of Realtime Database with a new and more intuitive data model. Cloud Firestore is richer, faster, and highly scalable compared to Realtime Database.

Like Firebase Realtime Database, it helps synchronize your data across client applications through real-time listeners and provides offline support for mobile and web devices. Cloud Functions and other Firebase and Google Cloud Platform products are also integrated with Cloud Firestore.

Figure 2. Cloud Firestore

# II. KEY FEATURES AND HOW IT WORKS?

## 1) Key features

## 1.1) Firebase Realtime Database

* **Realtime**: Data syncs instantly across all connected devices.
* **Offline**: Firebase Realtime Database allows apps to function offline by storing data locally and syncing changes upon reconnection.
* **Multi-device usage**: Accessible from both mobile devices and web browsers.
* **Scalability through database combinations**: Blaze plan enables scaling with multiple databases within a single Firebase project.

## 1.2) Cloud Firestore

## Flexibility: Cloud Firestore offers versatile data structures and supports hierarchical data storage.

* **Abstract queries**: You can use Cloud Firestore to perform versatile queries, including filtering, sorting, and fetching data with ease.
* **Real-time updates**: loud Firestore ensures real-time data synchronization across all connected devices.
* **Offline support**: stores data locally, allowing applications to function offline and syncing data when online.
* **Scalability**: Cloud Firestore inherits scalability features from Google Cloud Platform, making it suitable for large-scale applications.

## 2) How does it works?

### **2.1) Firebase Realtime Database**

Firebase Realtime Database allows users to build easy, flexible, and secure applications with direct client-side access to the database.

Real-time events continue to happen even when the data is stored locally. Realtime Database automatically resolves conflicts and synchronizes changes in local data with server data when devices reconnect. When paired with Firebase Authentication, Firebase Realtime Database offers additional security rules for determining read/write rights and data structure.

Realtime Database is a NoSQL database designed for speed and performance, making it possible to create real-time experiences that can serve millions of users without worrying about responsiveness.

### **2.2) Cloud Firestore**

Cloud Firestore is a cloud-hosted NoSQL database accessible directly through SDKs for iOS, Android, Web, etc. Data is stored in documents that map to values, in accordance with the NoSQL data paradigm. To facilitate data structuring and querying, these papers are arranged into collections.

Protect your data access in Cloud Firestore with Firebase Authentication for Android, iOS, and JavaScript or Identity and Access Management (IAM) for server-based access control.

Cloud Firestore is a cloud-hosted NoSQL database accessible directly through SDKs for iOS, Android, Web, etc. Data is stored in documents that map to values, in accordance with the NoSQL data paradigm. To facilitate data structuring and querying, these papers are arranged into collections.

Protect your data access in Cloud Firestore with Firebase Authentication for Android, iOS, and JavaScript or Identity and Access Management (IAM) for server-based access control.

# III. DIFFERENCES BETWEEN REALTIME DATABASE AND CLOUD FIRESTORE

**Data Model:**

* Both are NoSQL databases.
* Realtime Database stores data as a JSON tree, which is simpler but challenging for organizing complex and hierarchical data when scaling.
* Cloud Firestore stores information as groups of documents. Simple data can be handled as JSON, and larger data can be organized into subcollections. Less data normalization is necessary for Cloud Firestore.

**Real-time Data and Offline Support**:

* Both were the first databases to support Realtime SDK and offline storage.
* Realtime Database only provides offline support for iOS and Android, whereas Cloud Firestore offers offline support for iOS, Android, and web.

**Queries:**

* Data can be fetched, sorted, and filtered from any database using queries.
* Realtime Database supports queries with sorting, filtering, and limiting but only one of these functions can be applied in a single query.
* Sorting, filtering, and combining filters in one query is supported by compound queries in Cloud Firestore. Query performance scales with the size of your result set and is independent of data volume when using the default index-based query type.

**Write and Transaction:**

* Firebase Realtime Database performs writes and transactions as individual operations, whereas transactions require completion callbacks.
* Cloud Firebase allows batch writes and retries until they succeed.

**Stability and Performance:**

* Realtime Database is a mature product with high stability and low-latency data synchronization.
* Cloud Firestore offers higher stability than Realtime Database as it distributes data across servers worldwide.

**Scalability:**

* Realtime Database can support 100,000 concurrent connections and 1,000 writes per second in a single database, so scaling may require distributing data across multiple databases.
* Cloud Firestore automatically scales to accommodate your needs.

**Security**:

* Realtime Database uses standard authentication-based security rules.
* Cloud Firestore uses Identity and Access Management (IAM) on servers, while Cloud Firestore protection Rules offer straightforward yet effective protection for mobile and web platforms.

**Pricing**:

* Both are available in Firebase pricing plans, including Spark (Free), Flame ($25/month), and Blaze (Pay as you go).
* Real-time database fees: Higher rates correspond to faster performance; fees are depending on bandwidth and stored data.
* Cloud Firestore fees are determined by database operations (read, write, and delete), taking into account bandwidth, low rate, and data stored.

# IV. RESULT

In this app, we are choosing authentication, realtime database and storage to use because of some reasons and this is our experience when using them:

**Authentication:**

**Advantages:**

* Security: Implementing authentication ensures that only authorized users can access your app and its features, protecting user data and maintaining privacy.
* Personalization: Authenticated users can have personalized experiences, such as customized settings, data, and preferences.
* Authorization: Enables fine-grained control over user permissions, allowing you to restrict access to certain parts of the app or data.

**Disadvantages:**

* Implementation Complexity: Depending on the authentication method chosen (email/password, social logins, etc.), there might be some complexity in integrating and managing the authentication flow.

**Realtime Database:**

**Advantages:**

* Real-time Updates: Changes made to the database are immediately reflected across all connected clients, providing a seamless real-time experience for users.
* Offline Support: Users can interact with the app even without an internet connection, and the changes are synchronized once the connection is restored.
* Rapid Prototyping: Well-suited for prototyping and quick development due to its simplicity and ease of use.

**Disadvantages:**

* Limited Query Capabilities: Compared to Cloud Firestore, the Realtime Database has limited querying capabilities, which might be a consideration for complex data retrieval needs.
* Scaling Challenges: In certain scenarios with complex data structures, scaling might become a concern compared to the more scalable Cloud Firestore.

**Storage:**

**Advantages:**

* Media Management: Suitable for storing and retrieving user-generated content such as images, videos, and other media files.
* Scalability: Firebase Storage can handle large amounts of data and is designed to scale with the growing needs of your app.
* Integration with Firebase Services: Seamless integration with other Firebase services like the Realtime Database and authentication.

**Disadvantages:**

* Costs: Storage costs may increase with the amount of data stored and the frequency of data retrieval. Understanding the pricing model is essential to manage costs effectively.
* Complexity with Large Files: While Firebase Storage is efficient for smaller files, handling large files might require additional considerations, and there are limits on file sizes.

In conclusion, the combination of authentication, the Realtime Database, and storage in your app offers a secure, real-time, and scalable solution. While there might be some complexities and considerations in the implementation, the advantages outweigh the potential disadvantages, especially in terms of user experience, data synchronization, and seamless integration of features.

And below is our final product:

**Login:**

In here, we are first need to be sign in. If you are user has role Admin then your account is already created. In here, I has initialized 3 accounts with 3 role different from each other included: Admin, Manager and Employee. We will use account has role Admin in this report.

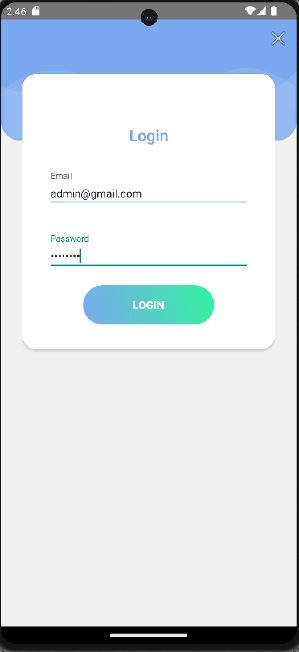
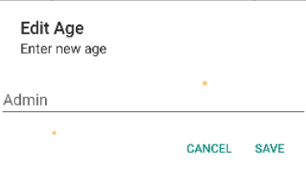
Our app is using firebase authentication because it will help a lots in access user data.

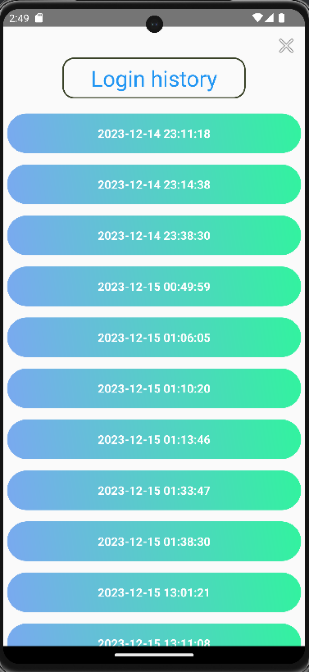
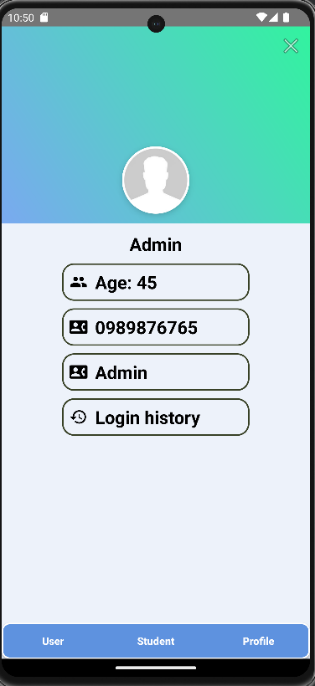
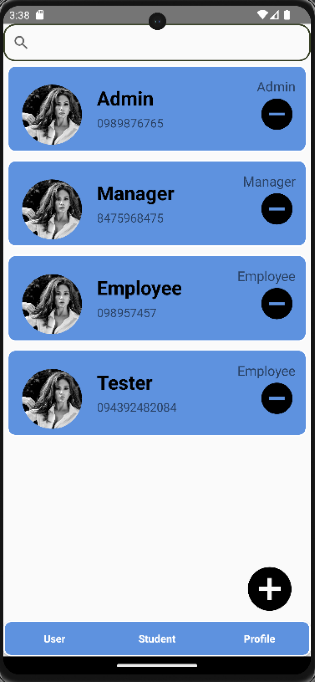
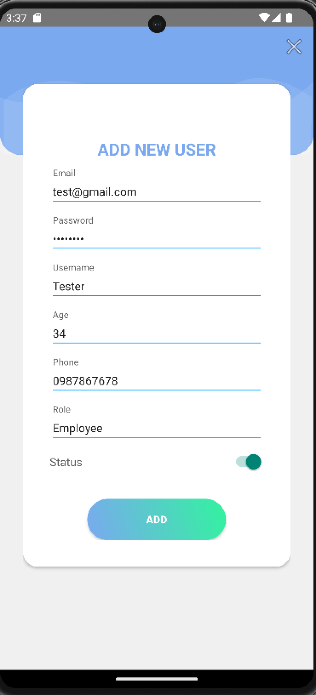
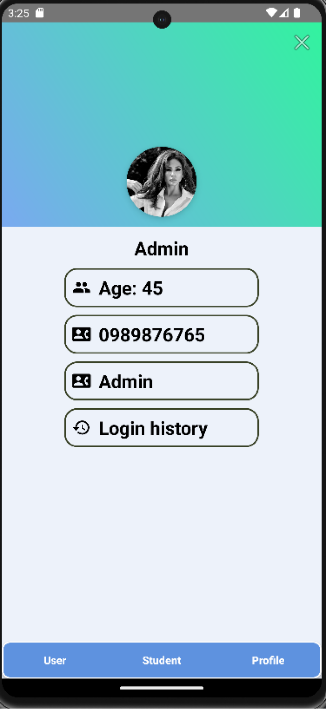
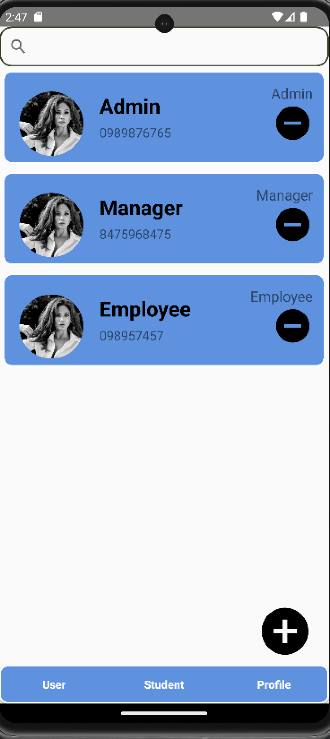
Figure 3. Login page

**User Account Management:**

After signed in, this is our first appearance of the app which is has the list of all users in database. From here we can access to others function included:

* Change profile picture.
* Add a new user: Name, Age, Phone Number, Status (Normal/Locked) by click in “+” icon in the bottom left of screen.
* Delete a user.
* Modify user information by accessing to the profile (click in user you want to modify or if you want to modify the current user, change to Profile).
* View login history of a user in their profile (only can check when you are current user).





Go to add user

Go to profile

View login history

After change avatar and age

After add user

Figure 4. User Account Management

**Student Management:**

After redirect to “Student” page, this is our appearance of the app which is has the list of 6 buttons corresponding with 6 function (but only 2 of them is working). From here we can access to others function included:

* View List of Students by click in Student List.
* Add a New Student by click in Add Student.
* Delete a Student is only can called in list of students with icon “-“ in the end of each student item.
* Sort the List of Students based on ID and name.
* Search for Students using ID and name.
* Access Student Details Page:
* View complete student information on the details screen.

+ Update Student Information by click in gender and birth.

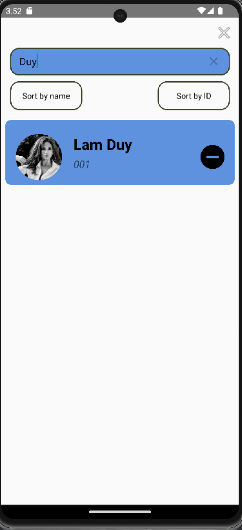
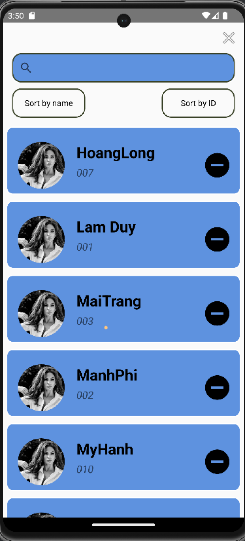
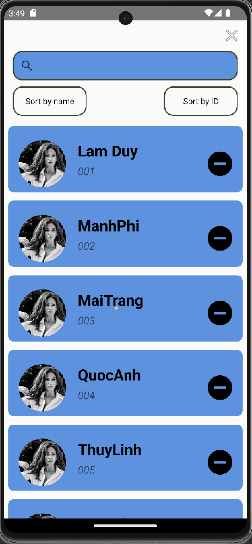
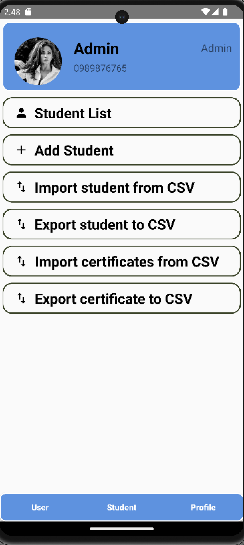
* Manage a list of certificates for students:
* View the list of certificates
* Add a new certificate
* Delete a certificate
* Update certificate information

And there is a part of 4 buttons in “Student” page which can not be used in this app right now included:

* Import a list of students from a file
* Export the list of students to Excel/CSV
* Import a list of certificates for a student from a file
* Export the list of certificates to Excel/CSV

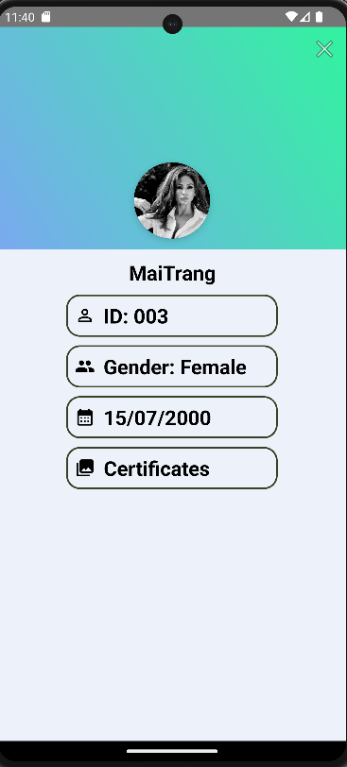
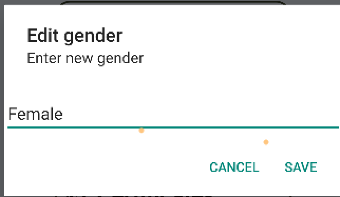
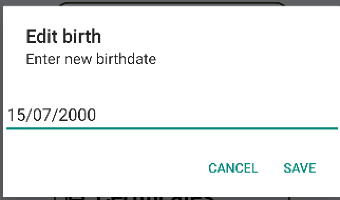
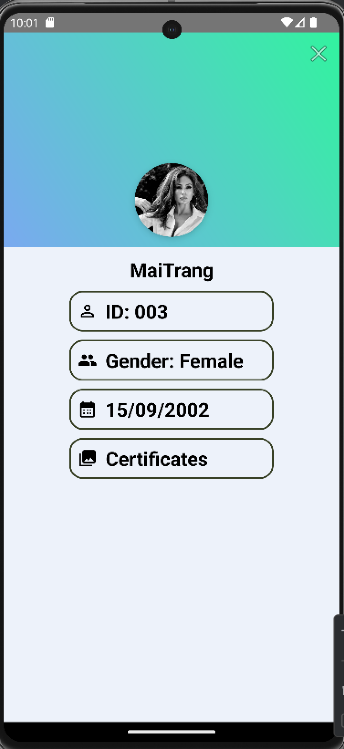
Below is our Student page and we will go to Student List:

Figure 5. Student page go to Student List has sort by ID, Name and search



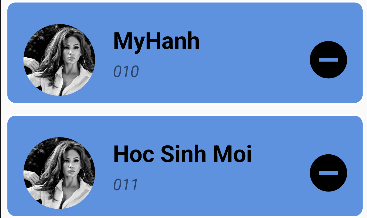
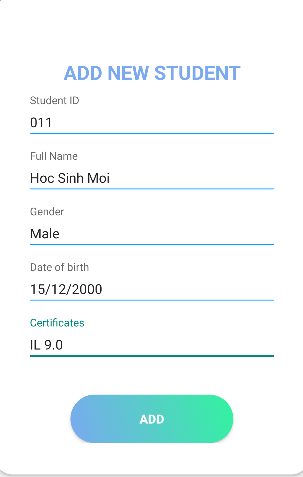
From Student List, we can access to each student details page, we choose go to MaiTrang. Then choose to edit gender and birth, after that we see there is some different from the old when new value is now update to database.

Figure 6. Details of student



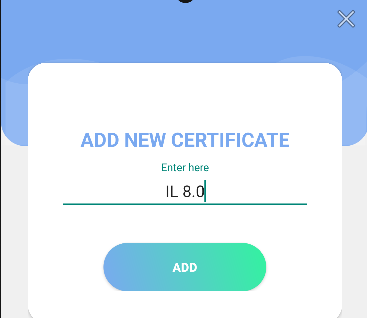
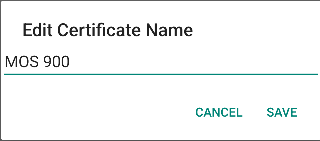
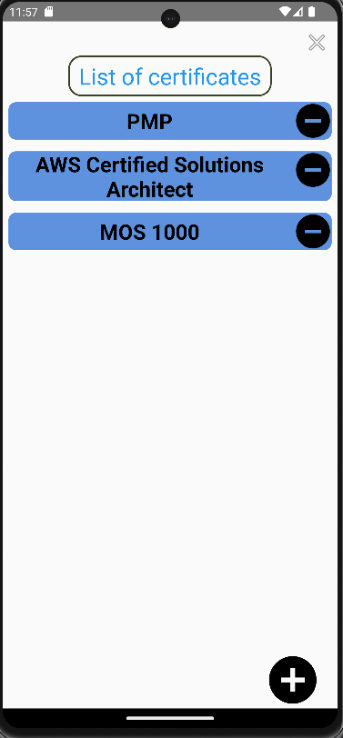
Then now we will go to Add Student to see how it works. From Student page, choose Add Student, enter some informations and ADD:

Figure 7 Adding new student



From details page of student, we can see their list of certificates. Let check what MaiTrang has in her profile and see can we modify them or add new one or delete

Figure 8. Modify, add list of certificates



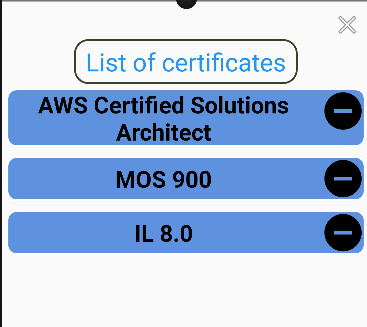
****

Figure 9. After delete a certificate

**References**

**Vietnamese**

**English**