ITAS 274 -Internet Programming III

Final Project

USE FIREBASE DATABASE TO STORE PHONE CONTACTS

Minh Trung. Duong

(April 2021)

I. Firebase Realtime Database

Firebase Realtime database is a cloud hosted database that supports multiple platforms Android, iOS, and Web. All the data is stored in JSON format and any changes in data, reflects immediately by performing a sync across all the platforms & devices. This allows us to build more flexible realtime apps easily with minimal effort.

CRUD is the 4 operations that are indispensable for every application with Database interaction. C (Create- add new), R (Retrieve - query data), U (Update- update data), D (Delete- delete data)

When there is a change of data on Firebase Database, immediately the interface of any device using this software will automatically update (called Realtime). Data on Firebase Database is in JSON format

The Firebase Realtime Database lets users build rich, collaborative applications by allowing secure access to the database directly from client-side code. Data is persisted locally, and even while offline, realtime events continue to fire, giving the end user a responsive experience. When the device regains connection, the Realtime Database synchronizes the local data changes with the remote updates that occurred while the client was offline, merging any conflicts automatically.

The Realtime Database provides a flexible, expression-based rules language, called Firebase Realtime Database Security Rules, to define how your data should be structured and when data can be read from or written to. When integrated with Firebase Authentication, developers can define who has access to what data, and how they can access it.

The Realtime Database is a NoSQL database and as such has different optimizations and functionality compared to a relational database. The Realtime Database API is designed to only allow operations that can be executed quickly. This enables you to build a great realtime experience that can serve millions of users without compromising on responsiveness. Because of this, it is important to think about how users need to access our data and then structure it accordingly.

II. Deploy the app

1. Requirements of the application

Phones using Android operating system version 7.0 or higher. It must have camera and free memory. The Application are allowed to use the camera and memory of the phone. In addition the phone must be connected to the internet. Because when creating, deleting or editing a contact, the application requests a connection to the Firebase.

Phones using Android operating system version 7.0 or higher. It must have camera and free memory. The Application are allowed to use the camera and memory of the phone. In addition the phone must be connected to the internet. Because when creating, deleting or editing a contact, the application requests a connection to the Firebase.

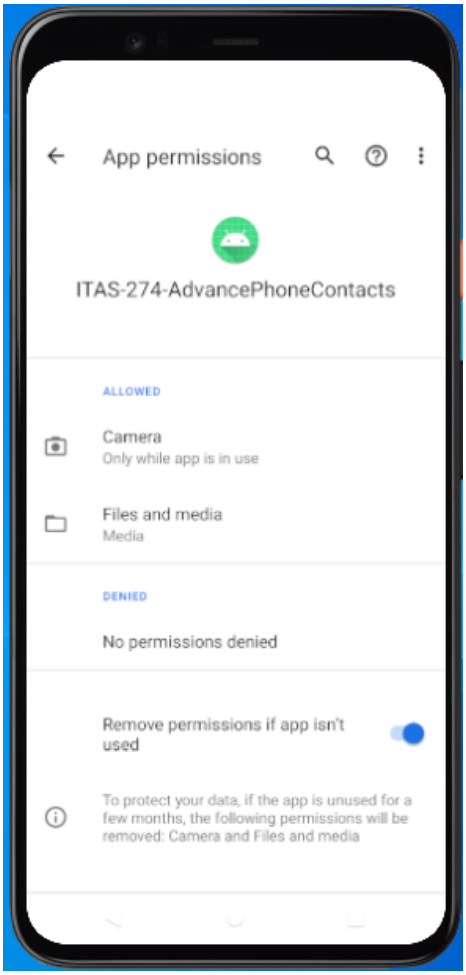

Figure. The Application are allowed to use the camera and memory

2. Create Realtime Database

I create a Realtime database (named ITAS247-phonecontacts), the contacts are stored in this database.

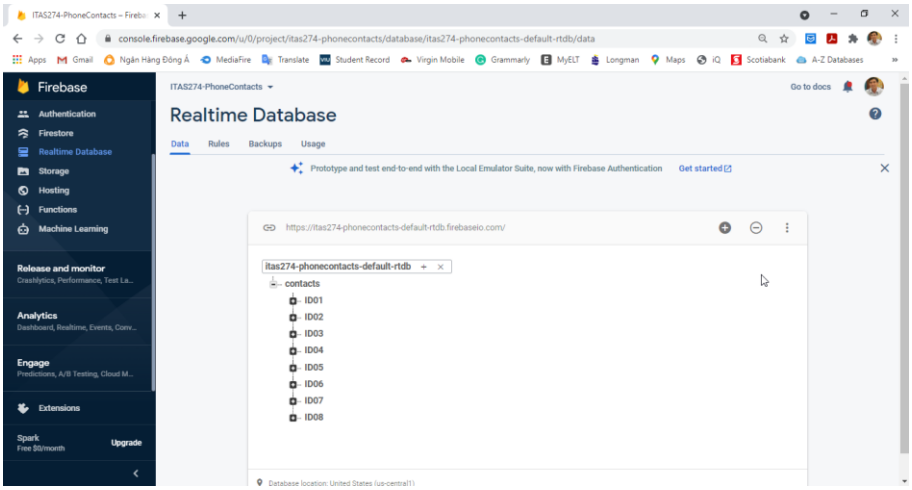

Figure. The Realtime database

3. Create New Project and Connect to the Realtime Database

I create a new Visual Studio Project and connect this project to the Realtime database *(ITAS247-PhoneContacts)* that I created earlier.

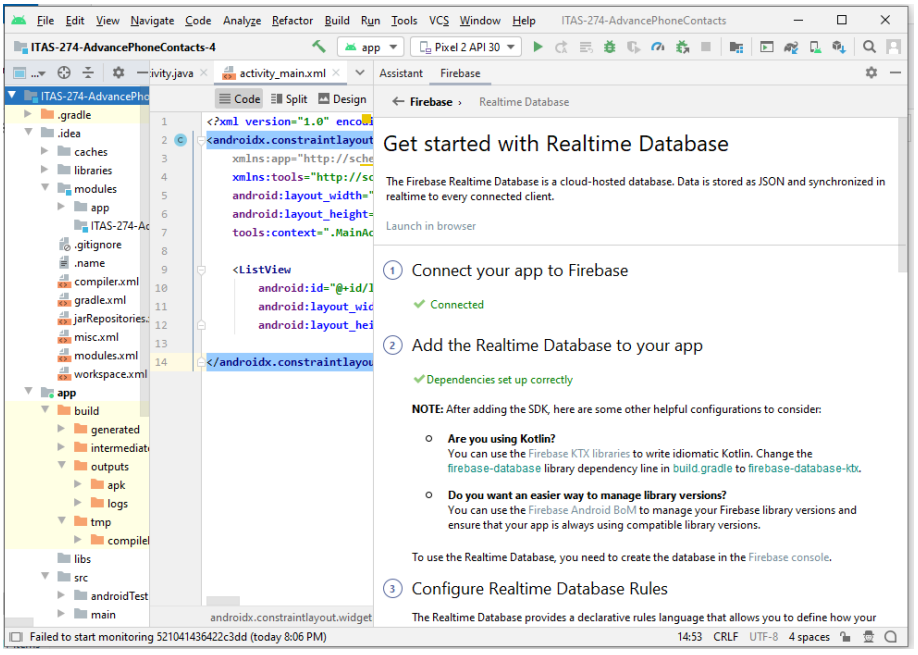

Figure. Connect to the Realtime database

4. Edit Database Rules

We access data on Firebase and allow to add, delete or modify records on it so we edit database rules to perform these operations.

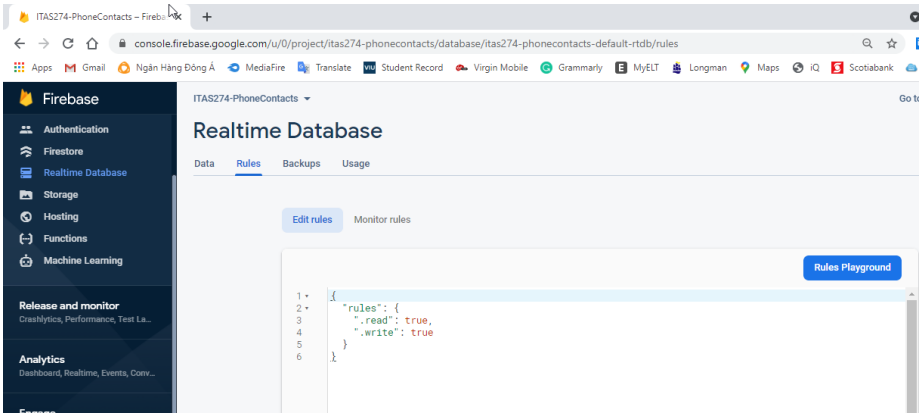

Figure. The Realtime database

5. Interface design

Main Activity screen

In the main screen, the application lists all the contacts that were previously saved. A contact includes: ID, name, email, phone number, and an image.

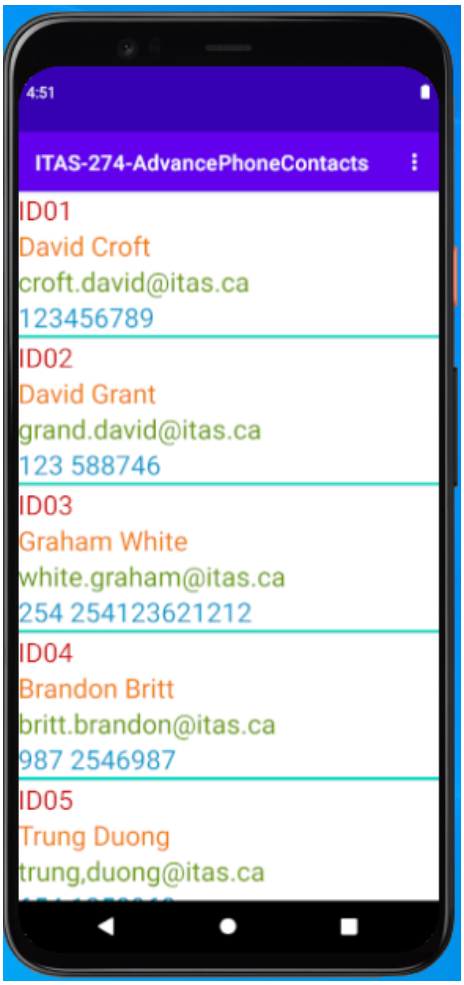

Figure. The main activity interface.

The application has 2 other interfaces, the interface of adding a new contact and editing an existing contact

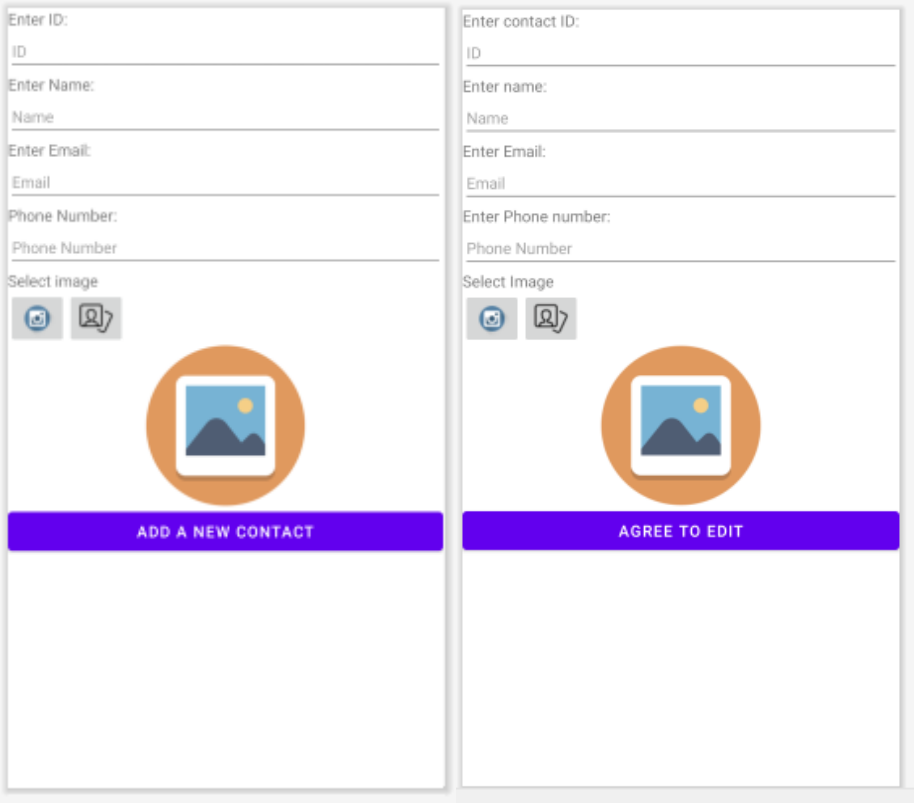

Figure. Add a new contact and edit a contact interface.

6. Add a new contact

In the upper right corner of the main menu, there is an “Add Contact” button. We can use it to add a new contact.

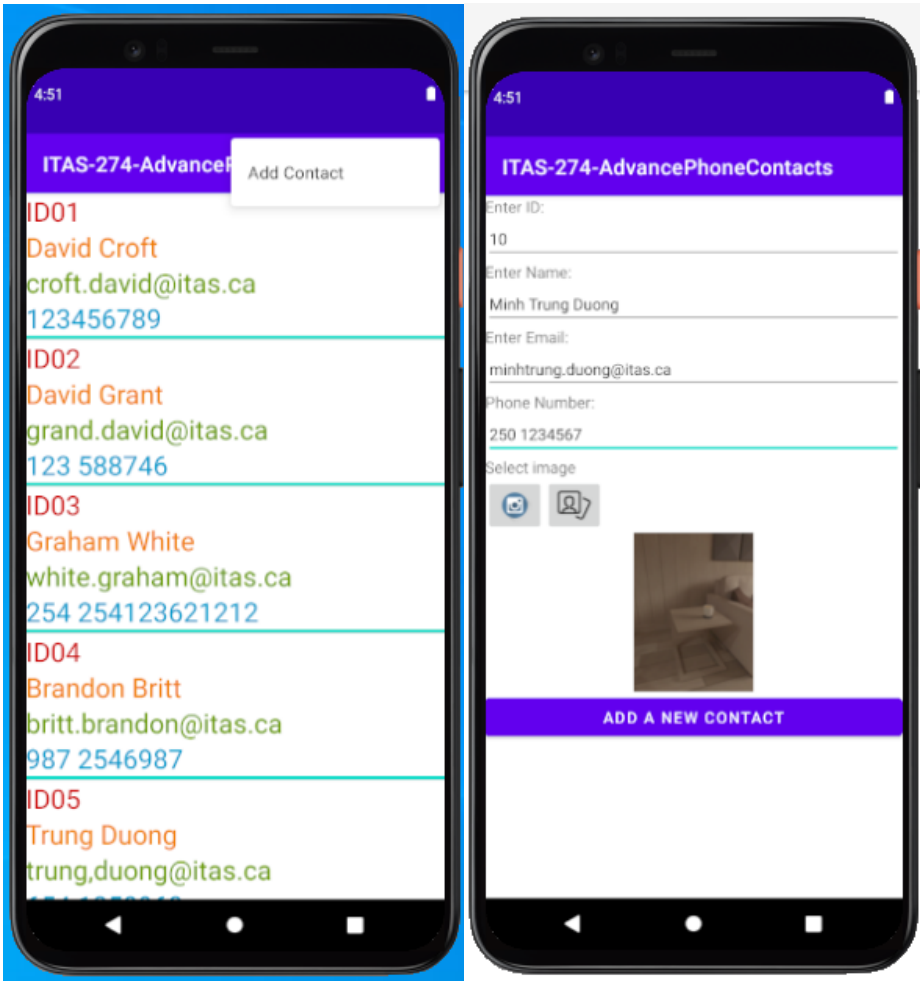

Figure. Add a new contact interface.

7. Edit a contact

To edit a contact, we can click on a contact we want to edit, the application will lead the user to the edit screen, after entering the information to edit, and we select the “AGREE TO EDIT” button to save those edits.

The editing function allows us to select an image previously saved on the phone or we can take a new photo from the camera as an avatar.

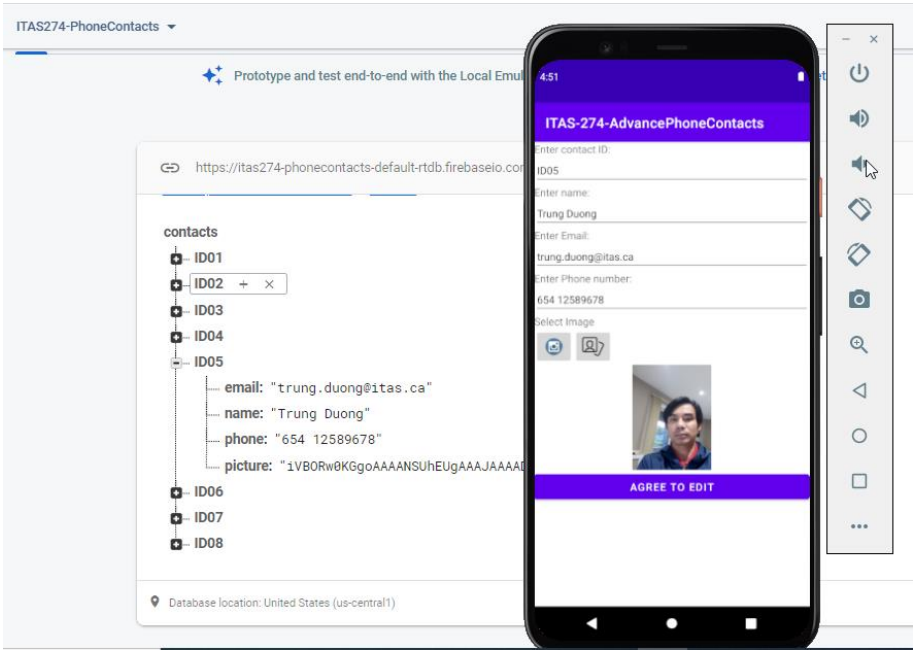

Figure. Edit a new contact

8. Delete a contact

To delete a contact click and hold for a short time (2-3 seconds), the contact will be deleted and displays a toast "*The contact has deleted successfully!"*

III. Conclusion

In this project, I got a chance to learn more about Android programming and Firebase Realtime Database, so I understand more about them. Besides that, I was able to write an application by myself, although it had many shortcomings that needed more work to be done. Overall, the project is very valuable as it further improves my knowledge of Android programming language and application connection to Realtime Database.

Reference

Use file metadata with Cloud Storage on Android, Firebase, 2021,

<https://firebase.google.com/docs/storage/android/upload-files?authuser=1>