UNIVERSITI MALAYSIA TERENGGANU

FACULTY OF OCEAN ENGINEERING TECHNOLOGY & INFORMATICS

CSM3123 : Native Mobile Programming

Group Project : Revolutionizing Business Connectivity and Innovative Business Card App

---

Prepared By Group 10 :

S62091	DANIAL SOLEHIN SAFIE
S62993	MUHAMMAD AFIQ HANIF SUHAIMI
S62250	MUAZ ZAINAL

Lecturer :

DR. RABIEI BIN MAMAT

BACHELOR OF COMPUTER SCIENCE (MOBILE COMPUTING) WITH HONOURS
SEMESTER I 2023/24

Table Of Contents

1.0 Introduction

Through a unique business card application, this project intends to revolutionise corporate connection. This app aims to transform the traditional business card into a dynamic and interactive tool, allowing users to effortlessly manage their professional connections by seamlessly adding, deleting, and editing business details. Otherwise, this solution will redefine how individuals network and engage within professional spheres by combining efficiency with modern functionality. The envisioned business card application seeks to revolutionize professional connections by offering dynamic and interactive digital business cards. Users can effortlessly create personalized cards with clickable links for enhancing the digital interactivity of their professional presence.

Additionally, this application provides a user-friendly interface, allowing seamless management of connections in real-time. Users can add, delete, and edit business details effortlessly, ensuring that their contact information remains up-to-date. This combination of digital interactivity and effortless management aims to redefine how individuals engage within professional spheres, blending modern functionality with efficient networking practices. By incorporating these features, the business card application can offer a comprehensive and modern solution for professionals looking to streamline and enhance their networking experience.

2.0 Problem Statements And Solutions

2.1 Inefficient Business Card Management

Problem : Conventional business cards lack dynamism and struggle to accommodate evolving business details beyond basic contact information, leading to inefficiencies in maintaining up-to-date professional connections.

Solution : This app will reimagine the traditional business card by providing a digital platform where users can easily add, edit, and delete business details. This includes phone numbers, company names, business descriptions, and locations, ensuring a comprehensive and adaptable representation of their professional identity.

2.2 Limited Flexibility in Business Representation

Problem : Static business cards restrict the representation of diverse business offerings and fail to reflect the dynamic nature of modern enterprises.

Solution : Through our app, users can modify and update business details effortlessly. This flexibility allows for comprehensive representation, showcasing various services or products offered by a company, thereby enhancing their networking capabilities.

2.3 Challenges in Keeping Business Information Current

Problem : Traditional business cards struggle to maintain accuracy in business details as they lack mechanisms for easy updates or changes.

Solution : This app facilitates easy deletion and addition of business information. Users can swiftly update phone numbers, company names, business descriptions, and physical locations, ensuring that their professional network always has access to the latest and most accurate details.

3.0 Main Objectives

3.1 Streamlined Business Connectivity

Develop an intuitive interface that simplifies the process of adding, deleting, and editing business details on a digital business card, ensuring efficient networking and professional connectivity.

3.2 Comprehensive Business Representation

Create a platform that accommodates diverse business information, including phone numbers, company names, business descriptions, and physical locations, providing users with a versatile tool to showcase their professional identity.

3.3 Effortless Information Management

Ensure that users can easily update and modify business details, fostering an environment where maintaining accurate and current business information is convenient and accessible.

4.0 Task Distribution

Team Members	Role	Description
Afiq & Danial	Frontend Developer	We will focus on designing the user interface and implementing frontend functionalities for adding, editing, and deleting contact cards.Additionally,we will focus on ensuring smooth integration between the frontend and backend components
Muaz	Backend Developer	Muaz will handle the backend development tasks, including setting up Firebase for data storage and implementing backend functions for CRUD operations.

5.0 Explanation by activity, coding, and resources

5.1 MainActivity.kt

- The **MainActivity** class extends **AppCompatActivity**, which is the base class for activities in Android with backward-compatible support.
- The **binding** variable is an instance of **ActivityMainBinding**, which is automatically generated based on the XML layout file (presumably **activity_main.xml**) using View Binding.
- In the **onCreate** method, the layout is inflated using **ActivityMainBinding** and set as the content view.
- Three buttons (**mainUpload**, **mainUpdate**, and **mainDelete**) are defined in the XML layout file.
- Click listeners are set up for each button, and when a button is clicked, it triggers an **Intent** to start a new activity (**UploadActivity**, **UpdateActivity**, or **DeleteActivity**).
- **finish()** is called after starting **UploadActivity** to close the current activity, preventing it from being in the back stack.

5.2 UpdateActivity.kt

- The **UpdateActivity** class extends **AppCompatActivity**.
- The **binding** variable is an instance of **ActivityUpdateBinding**, generated using View Binding.
- In the **onCreate** method, a click listener is set for the **updateButton**.
- When the button is clicked, data is retrieved from the input fields, and the **updateData** function is called.
- The **updateData** function initializes the Firebase database reference, creates a map containing the updated user data, and then updates the user data in the Firebase database.
- If the update is successful, the input fields are cleared, a success message is displayed, and the user is redirected to the main activity.
- If the update fails, a failure message is displayed.

5.3 DeleteActivity.kt

- The **DeleteActivity** class extends **AppCompatActivity**.
- The **binding** variable is an instance of **ActivityDeleteBinding**, generated using View Binding.
- In the **onCreate** method, a click listener is set for the **deleteButton**.
- When the button is clicked, the phone number is retrieved from the input field, and if it's not empty, the **deleteData** function is called.
- The **deleteData** function initializes the Firebase database reference, removes the user data from the database based on the provided phone number, and handles success and failure cases.
- If the deletion is successful, the input field is cleared, a success message is displayed, and the user is redirected to the main activity.
- If the deletion fails, a failure message is displayed, and the user is still redirected to the main activity.

5.4 UploadActivity.kt

- The UploadActivity class extends AppCompatActivity.
- The **binding** variable is an instance of **ActivityUploadBinding**, generated using View Binding.
- In the **onCreate** method, a click listener is set for the **saveButton**.
- When the button is clicked, data is retrieved from the input fields, and a **User** object is created with the retrieved data.
- The Firebase database reference is initialized, and the user data is saved to the database under the specified phone number.
- If the save operation is successful, the input fields are cleared, a success message is displayed, and the user is redirected to the main activity.
- If the save operation fails, a failure message is displayed

5.5 User.kt

- The class **User** is a data class, which is a concise way to create classes in Kotlin that are primarily used to hold data.
- The primary constructor is defined with four parameters: **name**, **business**, **location**, and **phone**.
- Each parameter has a default value of **null**, making them optional when creating an instance of the **User** class. This allows you to create instances with only the properties you want to set.
- The properties correspond to the user details you mentioned (name, business, location, and phone).
- By using a data class, you automatically get several helpful functions generated by the Kotlin compiler, such as **toString()**, **equals()**, and **hashCode()**.

5.0.1 User Interface Explanation

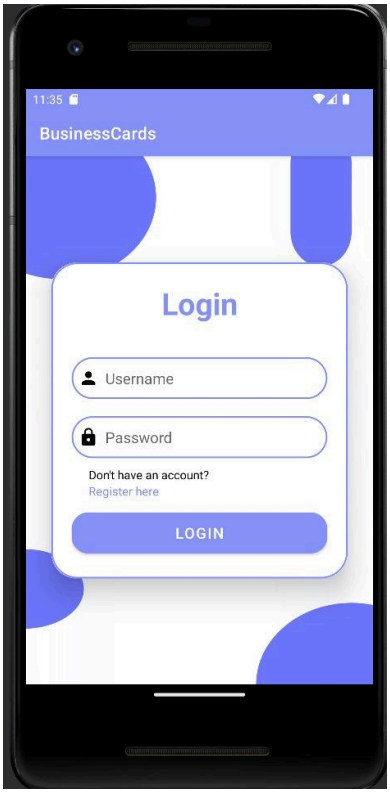

Figure 5.0 : *Login Page*

Figure 5.0 shows the Login Page that prompts users to enter their registered email address and the password they set during the sign-up process.This app will display the login first to verify the existing account. If the user's data is not available in the database, the user needs to go to the sign up page to verify the user's data.

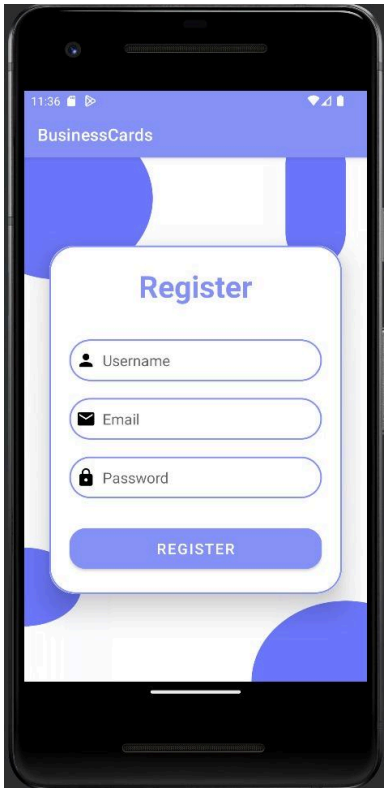

Figure 5.1 : *Sign Up Page*

Figure 5.1 shows the sign up page to access the innovative features and benefits of our business card application, users will need to complete a straightforward sign-up process. The sign-up process is designed to be user-friendly and efficient, ensuring a seamless onboarding experience.Enhance security and ensure the accuracy of user information, a verification step is included.

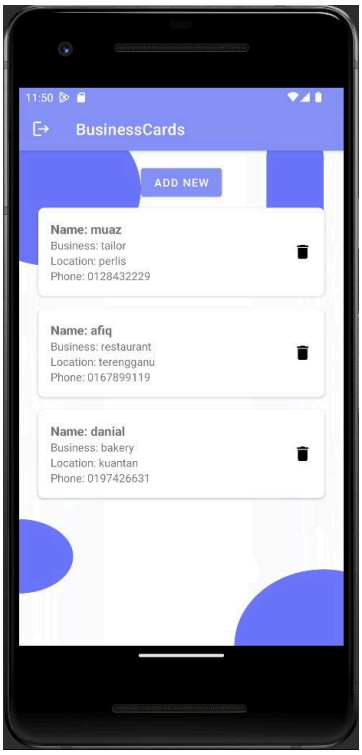

Figure 5.3 : *Home Page*

Figure 5.3 is the home page for our design. It will display 4 buttons which lead to its functionality. It has upload, view data, update and delete functionality.Users can create a new business card. Here they can submit their details such as name,business,location and phone number. Please take note that we use phone no as key.This is for delete functionality users need to put their number for deleting their details.

6.0 Link For The Github :

https://github.com/DanialSolehin/NativeProject