

# Phase 2 App Architecture

Group 5, Team Fish By: Jin Cabia, Dan Choi, Winko Peng, Cathy Sun

## UI Tool kit

### Context

**What is the issue that we're seeing that is motivating this decision or change?**

- Now all Modern Mobile Applications have a Front-End which is responsible for the design of the application. The Front-End of an application is crucial as it is the First Impression to the customers. A good Front-End is a necessity nowadays since it plays a massive role towards whether or not an application is usable or meets a customers needs. A Front-End is to supply users with an enjoyable user experience it is responsible for the Applications design, user flow and user retention.
- From Phase 1 Project Selection, Team Fish has selected a Mobile Expense Tracking application. In order to create a quality application utilizing the teams skill set, the team requires a UI Tool Kit where the team can learn quickly due to the short timeline of the project and where it best serves the needs of the developers and the project.

### Status

- Accepted

### Decision

**What is the change that we're proposing and/or doing?**

- Selecting a UI Toolkit.
- Considering the timeline of this project is relatively short we decided React Native Elements will be a good choice as many concepts of React Native Elements are present in React, which all team members are currently studying. There are many concepts and logic that are familiar which can help speed up our development process. Considering the teams skill set, each team member seems to prefer Object Oriented Programming while the whole team is weaker in terms of Styling and UI Design. React Native Elements has a large community which offers documentation and support, we can use videos and tutorials to learn about any features that may be needed in the process, as well as if we need to trouble shoot any problems along the way. In all, the reasons why we chose React Native Elements is due to how familiar it is to React as they both share similar concepts, and how there is an abundant amount of information online that allows us to learn, get by bugs and issues.

### Consequences

**What becomes easier or more difficult to do because of this change?**

- The timeline for this project is fairly short, as well as considering the skill set of the team being relatively new developers having a familiar language that we have worked previously with and studying currently in other courses React

Native Elements is the teams best fit for a UI Tool kit. The concepts of both React Native Elements and React are similar, using components and state management. React Native Elements provides a large community that offers documentation and support when troubleshooting issues. It provides pre-made components which can help the speed of development in the project.

- The design, look and feel that React Native Element provides leans more towards a minimalistic style which is very versatile, as it can work with any type of app. Which is a positive as the team currently has no direction in the styling of the application.
- However due to the limited amount of components pre-made by React Native Elements if the components provided can not fulfill the applications UI Design it can prove as a constraint.

## **Navigation Strategy for Fishy Bank Expense Tracking App**

### **Context**

A good navigation strategy is critical for the success of a mobile application. To provide user with best navigation experience, it is necessary to decide the Navigation Strategy of our Fishy Bank Expense Tracking App before starting the next User Interface Design phase.

### **Decision**

The design of the navigation menu items will follow the C.R.A.P. design principles:

- Contrast: The navigation menu icons will have sufficient color contrast between the text and background, so that they can be easy to read, especially for those users with accessibility needs.
- Repetition: The navigation menu will use the same color tone, font style, and font size. This could help the menu as well as the app looks consistent, professional, and easier for user to identify the theme.
- Alignment: The menu items will have the same alignment with sufficient white spaces in between, which could provide a good visual impression and increase readability.
- Proximity: The menu items will be grouped by the app's functionalities, which enable user to better understand the app features and structure.

After initial analysis, the Fishy Bank Expense Tacking app will have 4 key functional layouts, which can be accessed from 4 main menu icons. The app will use the bottom tab navigation method, there will have 4 fixed icons located at the bottom of the screen. This strategy enables users to access to our app functions quickly and easily, which

best fit to our app and matches with our goal. The following is the sample layout:



**Status**

Accepted

## Consequences

By using above navigation strategy, user will find our app interface clean, efficient, and easy to navigate. The navigation menu bar and items will also be responsive to users' mobile devices.

This ADR describes the navigation strategy for our Expense Tracking App project which will be developed within relative short timeline, but not suitable for app project with complex and huge structure.

## Hardware

### Context

Team Fish, we are developing a mobile application for expense and income tracking. As part of the application's functionality, we need to consider the hardware components necessary for certain features such as location tracking, audio output, and facial recognition.

### Decision

We have decided to integrate the following hardware components into our application:

- **GPS:** To track the user's location for expense logging and analysis purposes.
- **Speaker:** To provide audio feedback and notifications to the user.
- **Camera (with Face ID capability):** To enable facial recognition for secure authentication.

### Status

Accepted

### Rationale

- **GPS:** Utilizing GPS allows for accurate tracking of the user's location, enabling location-based consumption automatically selects the local currency and analyzes the frequency of user spending or income in a specific area.
- **Speaker:** Providing audio notifications through the speaker enhances accessibility for users who may have visual impairments.
- **Camera (with Face ID capability):** Integrating a camera with Face ID capability enables secure authentication, ensuring that only authorized users can access sensitive financial data within the application.

## Consequences

- **Increased Development Complexity:** Integrating hardware components like GPS, speaker, and camera with Face ID adds complexity to the development process.
- **Potential Privacy Concerns:** The use of GPS and facial recognition technology may raise privacy concerns among users.
- **Hardware Compatibility:** Compatibility issues may arise with certain devices that lack specific hardware features.
- **Potential Battery Drain:** Continuous use of GPS functionality may impact device battery life, requiring optimization strategies to minimize power consumption.

## Database Architecture

### Context

**Issue: What is the issue that we're seeing that is motivating this decision or change?**

- As "Fishy Bank" is an expense tracking application, it necessitates a robust and secure system for managing and storing financial data, including transactions, budgets, and user information. A well-structured database is essential for facilitating efficient data retrieval, updates, and ensuring data integrity. Additionally, considering the app's requirement to run on Android devices and be developed using React Native, the database must be compatible and performant on mobile platforms. The decision surrounding the database architecture is crucial as it directly impacts the app's scalability, security, and overall user experience.

## **Status**

- Accepted

## **Decision**

**What is the change that we're proposing and/or doing?**

- Implementing a local, encrypted database using Realm Database.

## **Group**

- Data

## **Assumptions**

- The application will be used on Android devices.
- Users expect their financial data to be stored securely.

## **Constraints**

- The database must operate efficiently on mobile devices.
- Data encryption is necessary to protect sensitive user information.

## **Positions**

- **SQLite:** A lightweight, local database option but lacks built-in encryption.
- **Firebase Realtime Database:** Offers real-time capabilities but requires internet connectivity and does not store data locally.
- **Realm Database:** Provides a local storage solution with encryption capabilities and is optimized for mobile platforms.

## **Argument**

- Realm Database is chosen for its performance on mobile devices and its support for encryption, which is crucial for securing financial data. Its React Native support aligns well with the team's skills and the app's development framework. While SQLite is a viable lightweight option, its lack of built-in encryption makes it less suitable for an application handling sensitive financial data. Firebase Realtime Database offers real-time capabilities but relies on internet connectivity and does not align with the requirement for local data storage.

## **Implications**

- Choosing Realm Database introduces the need for implementing encryption mechanisms, potentially requiring additional development time to ensure data security. This decision may also necessitate further research and learning for

the team, particularly in integrating Realm with React Native and managing encrypted data.

### **Related decisions**

- UI Toolkit selection
- Navigation strategy
- Integration of hardware features like fingerprint scanners for authentication

### **Related requirements**

- Secure storage of financial data
- Efficient data retrieval and storage on Android devices
- Compatibility with React Native

### **Related artifacts**

- N/A

### **Related principles**

- Security by design: Ensuring the app's architecture incorporates security measures from the outset.
- Scalability: Choosing a database solution that can accommodate growing data needs.
- User-centric design: Ensuring the database architecture supports a seamless and responsive user experience.

### **Notes**

- Concerns were raised about the learning curve associated with Realm and encryption. Further training and research will be allocated to ensure my team is equipped to implement this solution effectively.