

Team name: PersonaliFLY (product and innovation)

Proposed sector: IFE Product Innovation

Motivation

Oftentimes, in flight experiences, passengers are drawn to entertainment sources which primarily encompass watching movies, or other shows to keep their flight time occupied. Passengers are turned away by merchandise and in-flight sales, viewing it to be stagnant and that it can be substituted by any other product found in ordinary stores.

Here is where our Application proposal comes in. By keeping merchandising and retailing at the heart of our offer, we want to incorporate innovation in the way products are distributed, and personalise the offers instead. As we brainstormed through various marketing strategies, various studies show that profit margins can be greatly increased from personalising offers.

This way, consumers can benefit by getting the in-flight experience that they want, sold to them in a customised manner which appeals to their tastes and interests, and at prices which are considered as value for money by adding features which increase brand loyalty. Through personalisation, a win-win situation can be offered to both Consumers who are wary in decision making and Business who are focused on profit making and expansion.

User Stories

1. As a passenger who is particular about personalisation, I want to explore features which would cater to my specific needs and liking without having to browse too much.
2. As a passenger who is not interested in purchasing products but wants to explore apps for sportiveness, I want a feature or game to be interesting enough to appeal to my interests.
3. As a passenger who is savvy with money, I want to discover ways in which I can have control to maximise my savings as much as possible through discounts and referrals.
4. As a businessman who is concerned about profit making, I want my solution or idea to be as deployable as possible without much technicalities or high costs

5. As a businessman who is concerned about attracting the market (people) I want my solution to cater to all kinds of age groups and their different needs

Aim

The software we aim to develop consists of two main core features which we are willing to propose and explore.

The app will have a personality test which is introduced to consumers upon first opening the application, prompting them to discover more about their personality traits. This will then be guided to match them with products to their liking. For example, a person who focuses more on relaxing and peace and finds comfort in solitude would be more likely to purchase fragrance items such as diffusers for homes and customisable perfumes. Even if someone is unwilling to purchase any products, they would be willing to explore their personality for curiosity sake since personality tests provide a personalised result based on a set of options, which could further prompt them to explore products which they were unwilling to explore previously.

Should passengers be uninterested in personality tests, they can have the option to explore the in-app game.

Business Perspective

Introducing competitive elements via gamification could induce the idea that passengers have something to gain from their interaction with the IFE.

Recommending travel locations and products in tandem with a personality test could imbue a sense of personal sentiment between the product and passengers, which could incite a motivating factor to purchase the item.

Games within the app initiate passengers to explore the IFE hence, any interaction between the user and the IFE is intentional browsing.

adding missions or challenges to scour Kris SHOP or various IFE sources could allow the passengers to carefully navigate through the features and understand them better

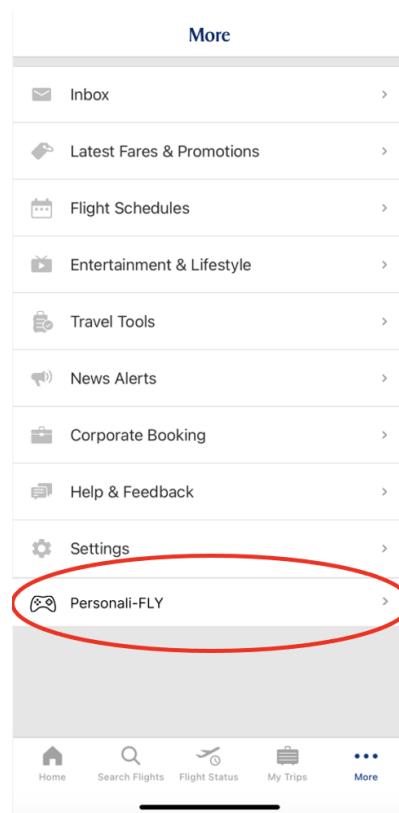
Posing various quizzes about the features of IFE could be a method to improve passengers' awareness of various deals being offered or the functionalities of the newly available features.

Tech stack:

1. ReactNative
2. Git
3. Firebase
4. Javascript

Features and Functionalities

How to access the Application:



Feature 1: Login Screen

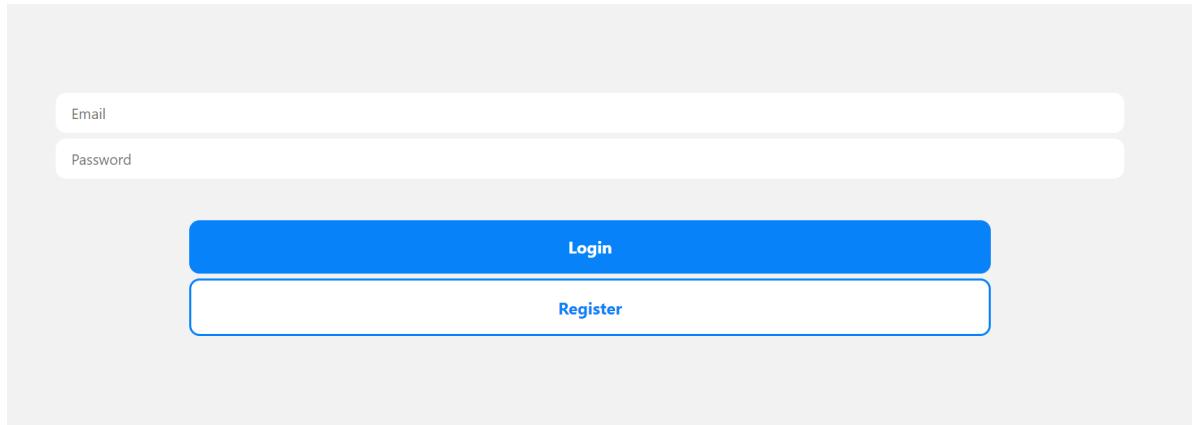


Figure 1: Login/Register Screen

```
Personalised-App-Manushri > MyApp > screens > JS LoginScreen.js > ...
1 import React, { useEffect, useReducer, useState } from 'react'
2 import { useNavigation } from '@react-navigation/core'
3 import { KeyboardAvoidingView, StyleSheet, Text, TextInput, TouchableOpacity, View } from 'react-native'
4 import { authentication } from '../firebase'
5 import { createUserWithEmailAndPassword } from "firebase/auth";
6 import { signInWithEmailAndPassword } from "firebase/auth";
7 import { getAuth, onAuthStateChanged } from "firebase/auth";
8
9 const LoginScreen = () => {
10   const [email, setEmail] = useState('')
11   const [password, setPassword] = useState('')
12
13 //add firebase-listener to navigate to home screen
14 const navigation = useNavigation();
15
16 useEffect(() => {
```

Figure 2: LoginScreen.js file

Explanation: **Firebase** was used to **authenticate users** upon a Login Screen which is greeted to users upon opening the application. With the use of Firebase, we used the authentication methods and the use of React Native's **useState** feature to enable users to log in using their email and password.

Two functions, **const handleSignUp** and **const handleLogin** is primarily used to record the **user credentials** and the “**catch**” command is to **ensure users do not register with an existing email which has been logged onto the system**.

```

<View style = {styles.buttonContainer}>
    <TouchableOpacity
        onPress={() => {handleLogin}}
        style={styles.button}
    >
        <Text style={styles.buttonText}>Login</Text>
    </TouchableOpacity>
    <TouchableOpacity
        onPress={handleSignUp}
        style={[styles.button, styles.buttonOutline]}
    >
        <Text style={styles.buttonOutlineText}>Register</Text>
    </TouchableOpacity>
</View>
</KeyboardAvoidingView>

```

Figure 3: LoginScreen.js file ReactNative commands

Explanation: Using ReactNative's inbuilt design features such as the **<View>** and **<Text>** commands, we were able to configure a **button** to register users and login existing users by linking the buttons to call the **handleLogin** and **handleSignUp function**, which reads from Firebase (as shown by the import **../firebase command**) at the top of the file to communicate with the backend database in Firebase web store and register new users. After successful sign up, users will be redirected to the Homepage of the Web Application (prototyped as such, will be an **Android/iOS app as deployment which is an extension to the existing SIA app**) using **firebase listener functions**.

Limitations: However, due to time constraints, our team was unable to **incorporate Firebase Firestore into the application** to store the **users database** and **fetch accordingly** when updated in the **app json file**. File. Hence, the App only works to register **new users** and **record them in Firebase**. Moreover, the app is provided as an **extension to the existing SIA app**, hence due to the information being lost, much time was spent incorporating **Firebase** as the **backend database** for users.

Feature 2: Homepage Screen



Figure 4: HomeScreen Page

Two side-by-side screenshots of code editors showing the file "App.js". The left editor shows the initial state of the file with imports for StyleSheet, React, NavigationContainer, and Stack.Navigator, along with a basic navigation structure. The right editor shows the file after modifications, including the import of screens from a separate folder and the creation of a NativeStackNavigator. The code includes components for HomeScreen, Quiz, and Game, and a style sheet for the container.

Figure 5: App.js page

Explanation: Using **NavigationContainers** and **Stack.Screen options**, we imported the various screens which would be redirected from pressing on the buttons in the Homepage (given the personality quiz buttons) and the **Myths Of Bali game button** to redirect users to a **game screen**. This will be stored in a separate folder named **Screens**, where the code was edited in the separate screen files to **display what was needed accordingly**.

Limitations: Lack of Navigation Bar and where the **buttons or clickability** is located could make the UI **messy and hard** to navigate around for users who are not used to the

new interface deployed. The UI also does not match the **existing App's UI** which could cause clashing.

Feature 3: Personality Quiz Screen

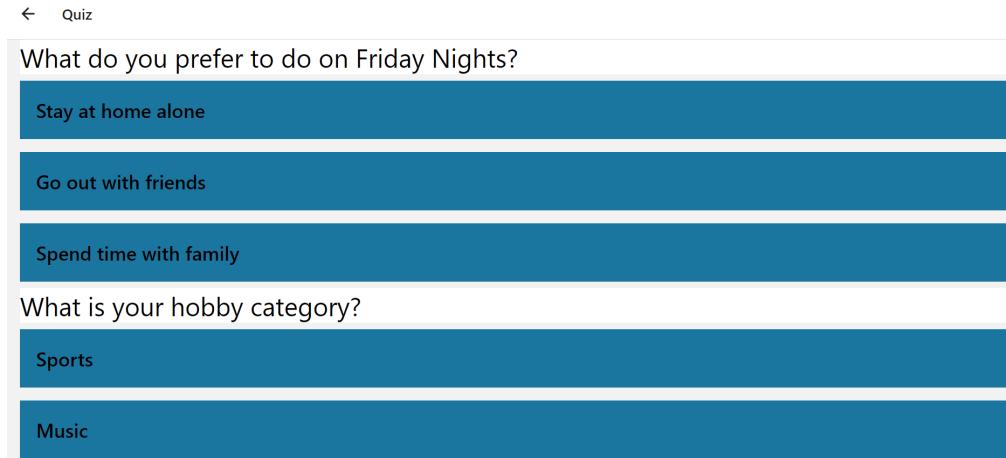


Figure X: Personality Test Screen

```
Personalised-App-Manushri > MyApp > screens > JS quiz.js > ...
18     title: "Why did you choose to fly with us?", ...
19     data: ["Peaceful", "Meet exciting people", "Entertainment sources"]
20   }
21 ];
22
23 const Item = ({ title }) => (
24   <View style={styles.item}>
25     <Text style={styles.title}>{title}</Text>
26   </View>
27 );
28
29 const Quiz = () => (
30   <SafeAreaView style={styles.container}>
31     <SectionList
32       sections={DATA}
33       keyExtractor={(item, index) => item + index}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER AZURE

Figure X: quiz.js file

Explanation: Using a **special array** to store the data for the **questions and the options** provided in the quiz, we were able to make a “list” with the questions in the personality test.

Limitations: However, button features/onclick features were not **appended to the array** of items accordingly, hence, for now it is currently **only a display or prototype** of what the personality test would **look like** in the application.

Feature 4: Game Screen

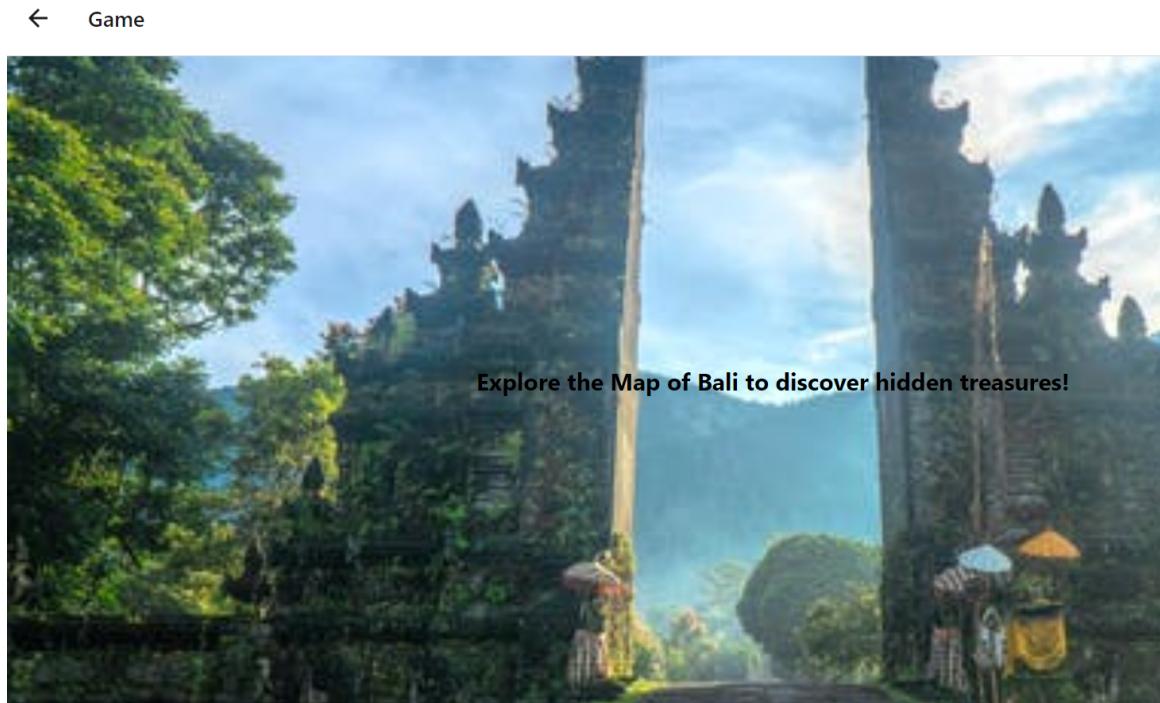


Figure X: Game Screen (Maps of Bali)

```
Personalised-App-Manushri > MyApp > screens > JS game.js > ...
1 import React from 'react';
2 import {StyleSheet, View, Text, ImageBackground} from 'react-native';
3
4 export default class Game extends React.Component {
5   render() {
6     return (
7       <ImageBackground source={require('./bali.jpeg')}
8         style={styles.container}>
9         <Text style={{fontSize: 20, fontWeight: 'bold'}}> Explore the Map of Bali to discover hidden treasures!</Text>
10      </ImageBackground>
11    );
12  }
13}
14
15 const styles = StyleSheet.create({
16   container: {
```

Figure X: game.js file

Explanation: The background of the **Game** will be a **live walkthrough** of a specified location in the **desired location of the user** (in this case, we are walking through the Maps

of Bali). From here, users will be able to **navigate with the on hold click of mouse** and go around with a **360 degree virtual reality** like experience to collect hidden treasure chests which contain discount codes and product discoveries. To append a **text on top of the image** (overlay) instead of using the <Image> key in reactnative, we used **<ImageBackground>** instead.

Limitations: There aren't any animations available, and what is shown is a static image once entered into the App as a demo. Information regarding where the product will be located, and further development is not available yet.

User Testing

The application was being tested on the web and the functions are not as complicated for the login feature and the general navigation tabs. Because of this, we conducted simple UI testing to see whether there are any UI bugs to be tackled and we carried out user acceptance testing. As such, we independently showed our program to friends and family members to determine how they would go about an app which is new to them, and any suggested improvements or advantages were taken note of. We underwent a few steps to determine the usability of the app.

Step 1) Presented the application to the audience and let them play around while giving them an overview of the app

Step 2) Let them experiment with the different buttons in the homescreen

Step 4) Let them think and walk through the personality test app

In total, 20 family and friends were interviewed and the advantages and disadvantages of the program are listed below.

Core advantages:

- The app is easy to navigate and understand
- The purpose of each functionality is clear without any prior instructions (the personality test and the game function)
- Unable to login as an existing user

Suggested Improvements:

- Could add more explanations, such as where the buttons are located in the page
- The personality test page could be more interactive

Version Control

Github

An organisation was created on Github, where there is a common repository which was forked into my own repository. From there, changes were added and committed, then we pushed any changes into our own repository. Any new commits will be displayed and the pull requests will be merged should there be no errors into the local organisation.

The purpose of creating the organisation was to ensure that the main code, should there be any errors in our individual files, can still be retrieved and was treated with more sensitivity and care.

Each commit and push is paired with a comment to understand the changes made to each version. For any bugs that are found on both ends, each of us individually are able to work on the files to fix the code before updating the particular files again. The person who updates the code should test the code to prevent any buggy code being pushed.

GitHub Repository

<https://github.com/SIA-APP-CHALLENGE/Personalised-App.git>

Video Walkthrough of App

https://drive.google.com/file/d/1hgeGwl3SS59fDnqbQ-Q52m_FvAzrfgx2/view?usp=sharing

Past Experiences:

Manushri:

- Involved in Independent Software Development Project, Orbital (NUS)
- Orbital: Created a sustainable Fashion Based App for Android with frontend and backend deployed (beginner project for students)
- Familiar with coding languages: JavaScript, C, and React Native Development servers

Vishnu:

- Presented a practical computer engineering system under CG1111A, a sensor-assisted autonomous vehicle, a robot
- Designing a complex computer engineering system under CG2111A that facilitates information processing, real-world interfacing, using bare metal coding
- Familiar with Python, C

