



Design & Innovation Project (DIP)

Project Report

Smart Mobile Apps (SMAPPS) Nanyang Marché



Project Group: E020

**Project Supervisor:
Ling Keck Voon**

**School of Electrical and Electronic Engineering
Academic Year 2022/23
Semester 1**

Table of Contents

1. Purpose/Project Objectives	1
2. Project Summary	2
3. Scope	3
4. Schedule	4
5. Cost	6
6. Integrated Development Environment	6
7. User Interface	7
8. User Experience (UX)	10
8.1. <i>Expected outcomes</i>	10
8.1.1. Initial Draft	10
8.1.2. Figma	11
8.2. <i>Actual Outcomes</i>	13
8.2.1. Onboarding Page	13
8.2.2. Login Page	14
8.2.3. Sign Up Page	15
8.2.4. Home Page	16
8.2.5. Upload Page	17
8.2.6. Product Details Page	18
8.2.7. Chat Page	19
8.2.8. Profile Page	20
8.3. <i>Chat Function</i>	21
8.4. <i>Navigation Bar</i>	21
9. Front-End Analysis	22
9.1. <i>Introduction:</i>	22
9.2. <i>Criteria</i>	22
9.3. <i>Learning speed and process</i>	22
9.4. <i>Cross-Platform</i>	23
9.5. <i>Performance</i>	23
9.6. <i>Popularity and Community Support</i>	25
9.7. <i>User Interface and Plugins</i>	27
9.8. <i>Documentation</i>	27
9.9. <i>Our Choice</i>	28
10. Back-End Analysis	29
10.1. <i>Version Control</i>	29
10.2. <i>Firebase</i>	29
10.3. <i>Cloud Firestore</i>	30
10.4. <i>Schema Creation</i>	31
10.5. <i>Firebase Storage</i>	32
10.6. <i>Authentication</i>	33
10.7. <i>Upload</i>	34
10.8. <i>Chat</i>	35
10.8.1. <i>Cloud Firestore for Chat implementation</i>	35
10.8.2. <i>Stream Chat Messaging API</i>	36
10.9. <i>Actual Outcome</i>	36
11. Future Improvements & Recommendations	37
11.1. <i>Profanity Filter</i>	37
11.2. <i>Quality of Life Features</i>	37
11.3. <i>Other Recommendations</i>	37
12. Individual Reflection	38
12.1. <i>Isaac Tay Peng Bin</i>	38
12.2. <i>Chai Jia Zhe</i>	39
12.3. <i>Stanley Wong Qi Ren</i>	40
12.4. <i>Rithikha Vivekanantham</i>	41
12.5. <i>Ke Tangxin</i>	42
12.6. <i>Lim Zhi Wei</i>	43
12.7. <i>Jake Chang Jie</i>	44
12.8. <i>Khairunnisa Ranam</i>	45
13. References	46
14. List of Figure	47

1. Purpose/Project Objectives

The purpose of this project is to provide Nanyang Technological University (NTU) students with a platform to market their items at their convenience. We created an app, named “Nanyang Marche”, that solves the issues with the existing solution to connect buyers and sellers in NTU. To provide some background information on the problem, there is an existing Telegram group, “NTU Marketplace”, where students can post items for sale in hopes of other students seeing the post and contacting the seller to buy the item. There are several issues with the user experience of using this Telegram group.

- A) Telegram is fundamentally a messaging app and as such the user interface is not designed for selling items.
- B) Telegram will show messages in sequence according to the time that it was sent. This causes listings to be quickly pushed out of view.
- C) Regarding the previous point, sellers may post repeated listings to get their items seen by more people. This makes it more difficult for buyers to find products in which they are interested.
- D) There is no categorisation of listings and as such, buyers will be shown many listings that are not relevant to what they are looking for.
- E) There is no standardisation on how the listings should be created. Sellers may omit certain valuable information by accident, or the layout of the listing will simply be different depending on each seller, making it harder to parse the information in the listing.
- F) Sellers have no convenient way of marking items as sold. This leads to some listings remaining on the chat group even though the item is no longer available for others to buy.
- G) Users may have a hard time keeping track of individual chats for each listing and may get confused as to which item the other party is referring to when having multiple conversations simultaneously.

This project serves to enhance the user experience as the current solution, the Telegram group “NTU Marketplace”, has poor user experience.

The outcome is an app which allows NTU students to buy and sell used items to one another. It will have a functional and practical design that will reach a substantial proportion of NTU students. It will solve all the issues with the existing solutions that were mentioned above. On top of that, there will also be additional features that will improve the overall user experience. More details will be provided in section 2 of this report.

External parties that do not have an NTU education email cannot use the app and blacklisted individuals will not be allowed access to the marketplace, in hopes of fostering a bonded NTU community.

2. Project Summary

This chapter describes the problem that was solved by the project. Did the project achieve its goals: was the problem solved completely or were there some issues that will remain unsolved?

Currently, NTU students are using a telegram group named “NTU Marketplace” to advertise and sell their pre-loved items. This group had garnered a strong following, as essential items such as furniture can be purchased and retrieved at their convenience.

However, it was discovered that the items were not given enough shelf time as new listings are sent in the group rapidly, and users are only able to view the listings while they scroll. Furthermore, descriptions of each listing are not standardized, and users will have to spend additional time to understand each message.

“Nanyang Marché” is a mobile application that aims to enhance the Telegram group by adding more User Interface (UI) interactions, allowing users to do more, whilst not compromising on the original sales experience.

These enhancements will ensure that past listings stay relevant, displaying the items and their description clearly at the press of the search button.

This application will also include an authentication feature, made strictly for NTU faculty and students, in hopes of cultivating an even stronger sense of community and closeness within the University.

3. Scope

As this application serves as an enhancement, the team had agreed upon 3 fundamental project outputs, along with a few secondary outputs to be implemented in the application.

Firstly, users must be able to upload their listings. Given that this application is an enhancement, we should not take away its original purpose. Enhancing this feature only involves UI, where the user now has a designated space to upload their image and product description on the cloud.

Secondly, the UI of the marketplace i.e., the listings or home page, must be standardized and clear in terms of the listing's description. A typical listing description will include an image, its name, its price, and additional comments. An item box that shows the listing's description in a standardized manner was implemented.

Thirdly, there must be a form of authentication. This is so that spam and abuse due to anonymity can be filtered out. Furthermore, this feature aims to foster a bonded community in NTU.

It is noted that the authentication feature was to be used to implement a forum page for students to request items they could not find in the marketplace, however, we needed to focus on delivering the 3 main outputs hence this objective was not implemented.

4. Schedule

PHASE	Planned Milestone Date	Actual Milestone Date
Initiating Phase	Week 1 <ul style="list-style-type: none"> • Introduction to the module • Meeting the project supervisor and team members to set expectations 	Week 1
Planning Phase	Week 2-4 <ul style="list-style-type: none"> • Created Microsoft Teams and Telegram groups for document sharing and online meeting • Brainstorming the real-life issues faced and the viable solutions • Gantt Chart planning • Finalizing the programming language for mobile app creation • Allocation of roles for group members • Researching in framework population • Deciding on the finalised project idea and reason for the problem statement • Deciding features • Studying the mobile app developer basic concept • Decided Flutter to be the core programming language for this mobile app • Brainstorming problems and idea of the problem • Learning Flutter through outsource • Successfully created Figma, GitHub, Firebase account for all team members • Designing icon and on boarding pages using Figma 	Week 4
Execution Phase	Week 5-10 <ul style="list-style-type: none"> • Enabling GitHub access for group members • UX/UI design in Figma • Studying on responsive interface design in Flutter and Figma • Reconsidered UX/UI design platform • Mirage Figma to pure Flutter programming • Chat, login, and sign-up function in Flutter • Icon and logo design • UX/UI interface design • Firebase in Google • Final debugging • Uploaded final coding to GitHub platform 	Week 11

Closing Phase	Week 11-12 <ul style="list-style-type: none">• Finalising Flutter App contents• Drafting the final report	Week 12
Project End Date	Week 13 <ul style="list-style-type: none">• Preparing for the presentation	Week 13

5. Cost

We did not have incur any costs due to the nature of the mobile application project; we already had our laptops to write our code on and most of the software used like Android Studio, VSCode, Flutter and Firebase are all open source or have a free tier.

PHASE	Planned Costs	Actual Costs
Initiating Phase	\$0	\$0
Planning Phase	\$0	\$0
Execution Phase	\$0	\$0
Closing Phase	\$0	\$0
Project Total Costs	\$0	\$0

6. Integrated Development Environment

Our team's choice for the Integrated Development Environment (IDE) was Android Studio as the framework we used was Flutter. Android Studio was our preferred choice because it inherently includes Android toolchains like Android Software Development Kit (SDK) and emulators via the Android Virtual Device (AVD), it also has multiple in-built features such as code refactoring, suggestions, start, stop, and hot reload buttons and git functions. Therefore, it was the ideal environment to develop our application. Alternatively, we used Visual Studio Code for ease of use and writing code. While Android Studio had a lot of features, it was also very heavy and slow, so when we wanted to make a minor change, we would use a lightweight IDE or text editor like Visual Studio Code, which came with extensions we could download to help with our coding and development process.



Figure 1 Our First Choice in IDE: Android Studio



Figure 2 Our Alternative IDE: VS Code

7. User Interface

In the matter of UI, our team took inspiration from the NTU Mascot, Leon, as shown below in figure 1, to come up with the colour scheme. This is because our application was implemented to serve the NTU student community, so we decided to use something that is meaningful to NTU. In addition, our application design was created in such a way that it has its own distinctive features and stands out from the current applications available on the market. Like the mascot, our application has a unique responsibility to establish a special visual identity for NTU.

The primary colours used were black, blue and different hues of orange. Neutral colours such as black were used for the text to maintain better readability and promote a sense of formality and security. With regards to the overall design of the app, blue was also primarily chosen because it projects subtle messages of trustworthiness and loyalty. In addition, orange was used because it symbolizes success, and we believe that our app will be successful in achieving our desired objectives eventually at the end of the project.



Figure 3 NTU Mascot.



Figure 4 Colour scheme for the application.

The previously proposed designs of Nanyang Marché's logos are depicted in the figure below.



Figure 5 Previous design of Nanyang Marché.

However, to further refine the Nanyang Marché logo design, the orange paper bag is flipped to the left side to avoid confusion about the word 'Marché'.



Figure 6 Nanyang Marché logo and icon.

The font 'Bauhaus 93' is chosen as the letter 'n' is the most similar to the handle of a paper bag. To ensure the uniqueness of the logo, the team included a 'moustache' on top of the 'e' on the word Marché. For the filter function in Figma, the icons are depicted below.



Figure 7 Icon example from Figma platform.

Our inspiration for the background wallpaper of the Nanyang Marché application was derived from the Lilit Store application as shown in the figure below. The transparency of the background is set to 10% in Figma.

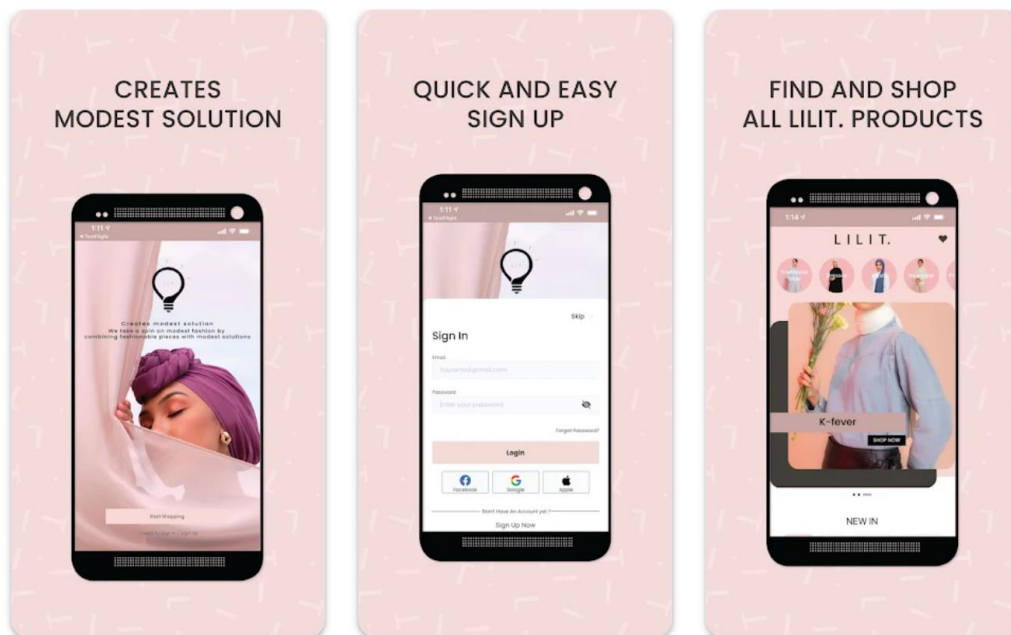


Figure 8 Lilit Store's wallpaper

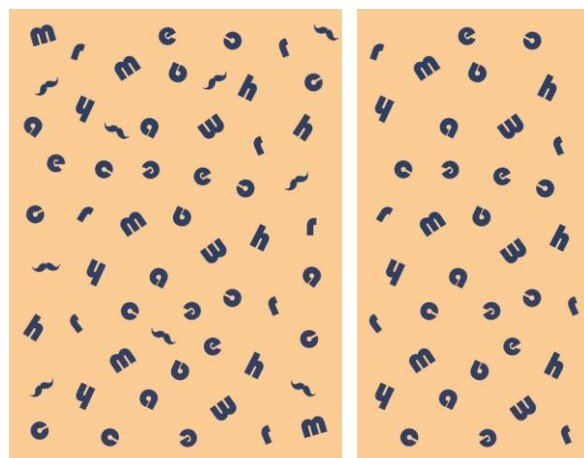


Figure 9 Nanyang Marché's background wallpaper

8. User Experience (UX)

Other than the attractive design of an application, the user experience is also essential in enhancing the users' satisfaction. Hence, easily understandable features have been added to improve users' interaction with our product. Considering all this, our team has come up with a design which is convenient to use and includes interactive features. Our application consists of a total of 8 pages: onboarding page, signup page, login page, home page, product detail page, upload page, chat page and the profile page. The initial sketch of the design of our mobile application was done on paper as shown in Figure 10 below.

8.1. Expected outcomes

8.1.1. Initial Draft

Our UI/UX team initially had a variety of ideas during the 1st phase of the timeline but unfortunately not all of them were implemented in the final product idea due to the time constraints.

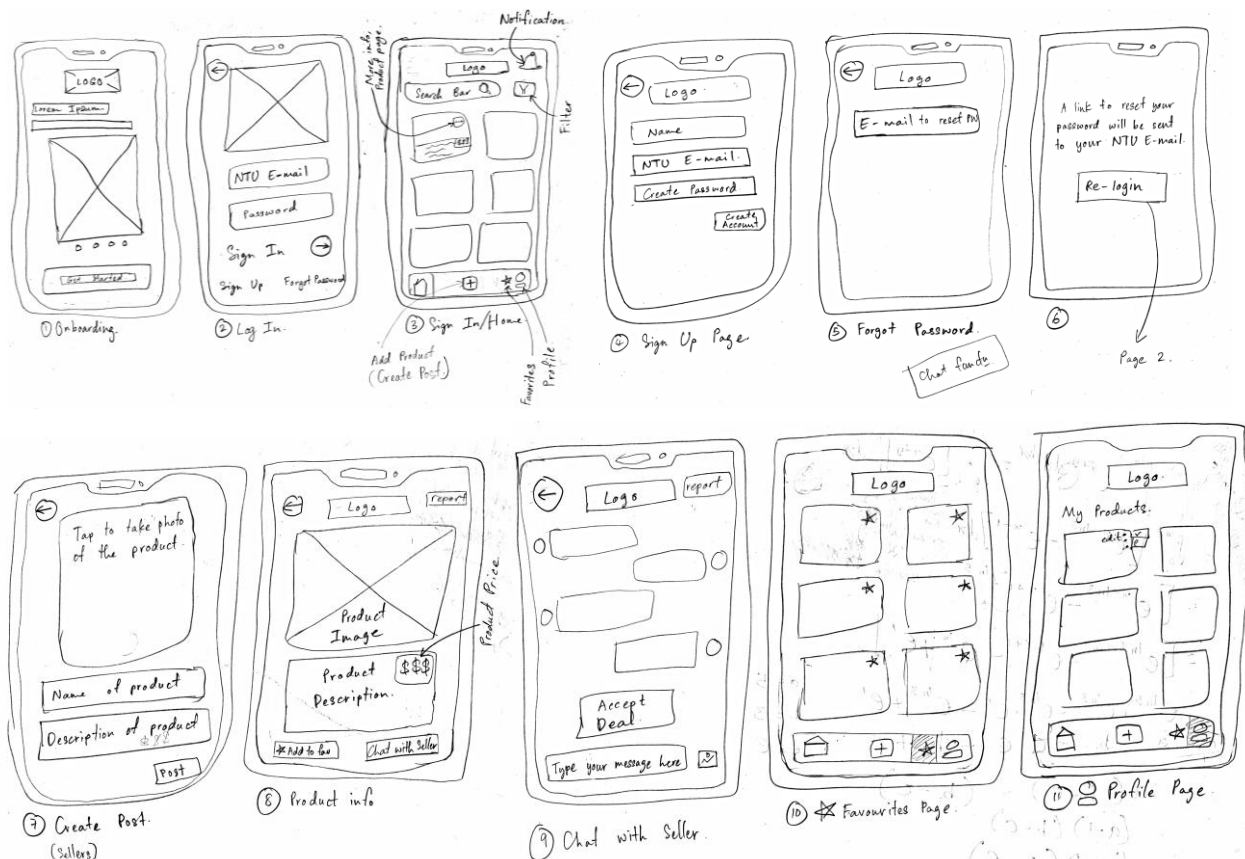


Figure 10 Initial rough sketches of the application.

As observed from Figure 10, the following points will explain in detail the main aspects of our application as well as how they achieve the expected outcome of the app.

8.1.2. Figma

Figma was used as the wireframing tool for the actual UI/UX design of our application. Figma's prototyping was easy to use, and we were also able to add in animations and screen presets, so it was one of the most convenient ways to present our prototype. Furthermore, Figma has a function which enables us to obtain the plugins for each of the widgets that we have added to our design. Therefore, when referencing the design for developing the actual Flutter code, we were able to obtain the exact color scheme, sizing, text formatting, etc. When we were finalizing the final design, we added some components to make our application more interactive while prioritizing minimalist design and removing the unnecessary components.

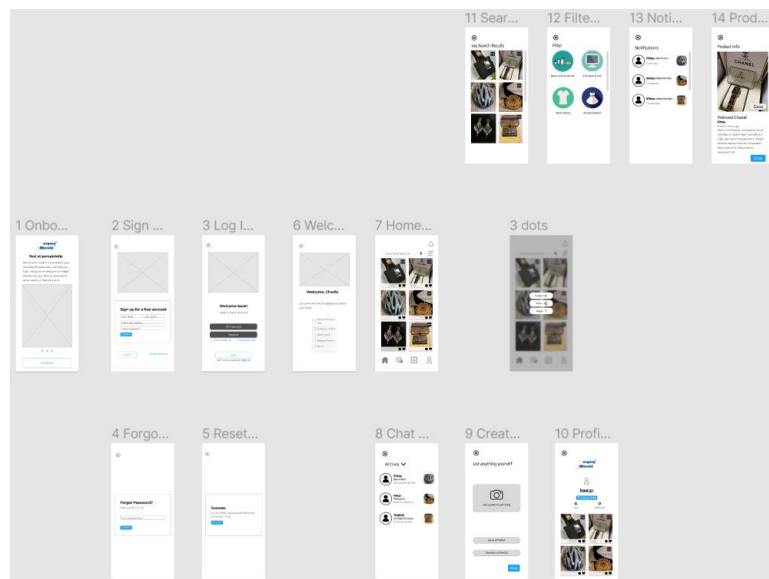


Figure 11 Transferred sketches into Figma.

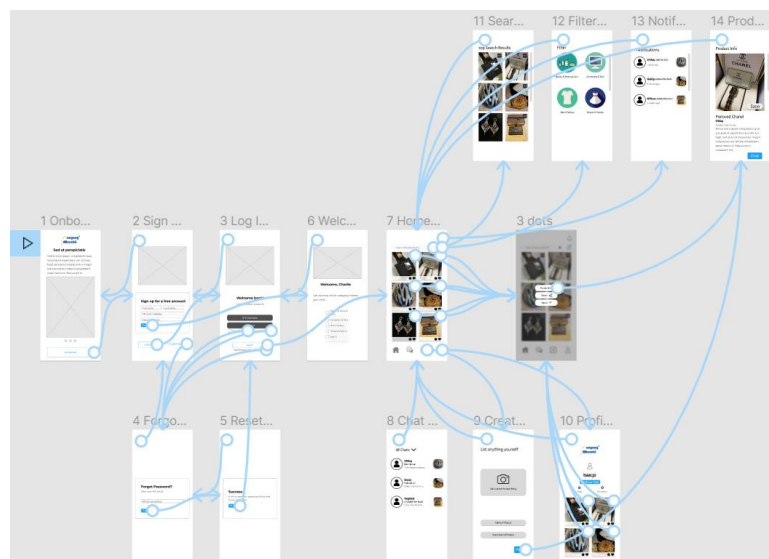


Figure 12 Figma's user flow.

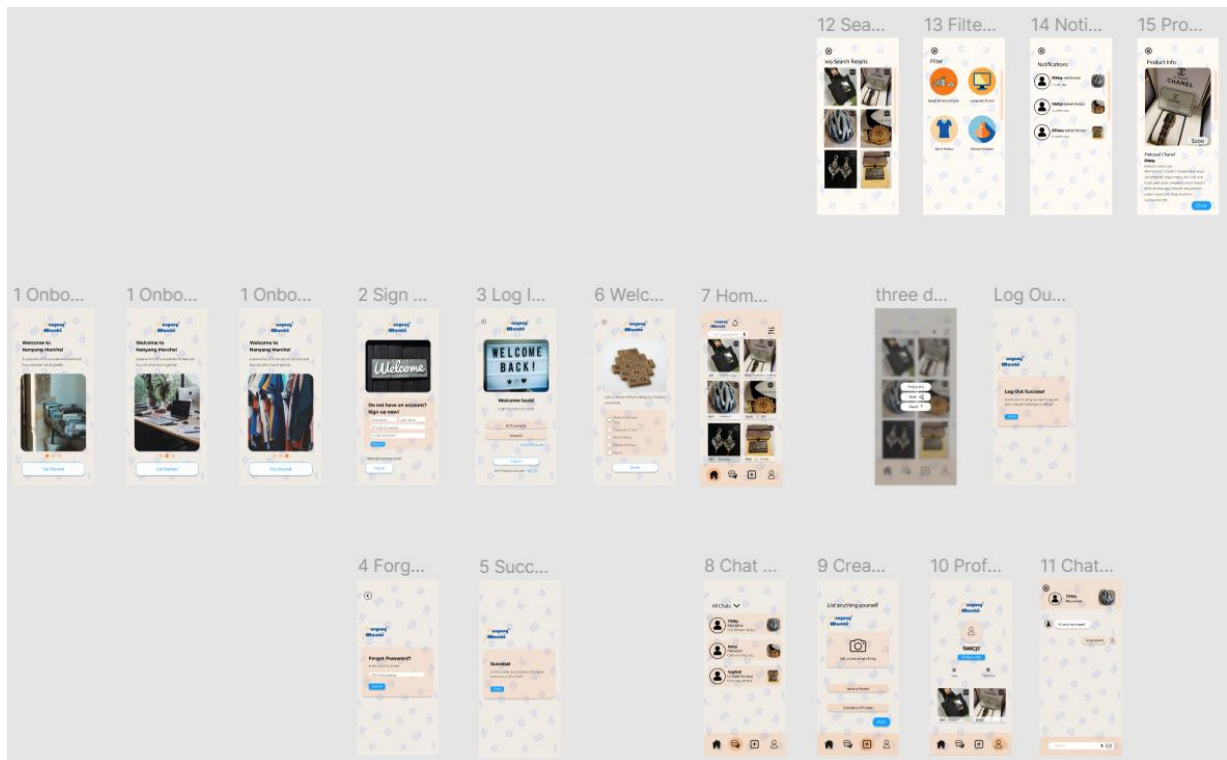


Figure 13 Application's expected outcome

8.2. Actual Outcomes

Figma has been used to serve as a reference to write the code to develop the application as shown in the figures below.

8.2.1. Onboarding Page

A user's first interaction with the application should capture their attention, hence we created the onboarding page to be dynamic. There are a total of three pages, it will display the next page of the onboarding when the user swipes the image at the center as shown in Figure 14 below.

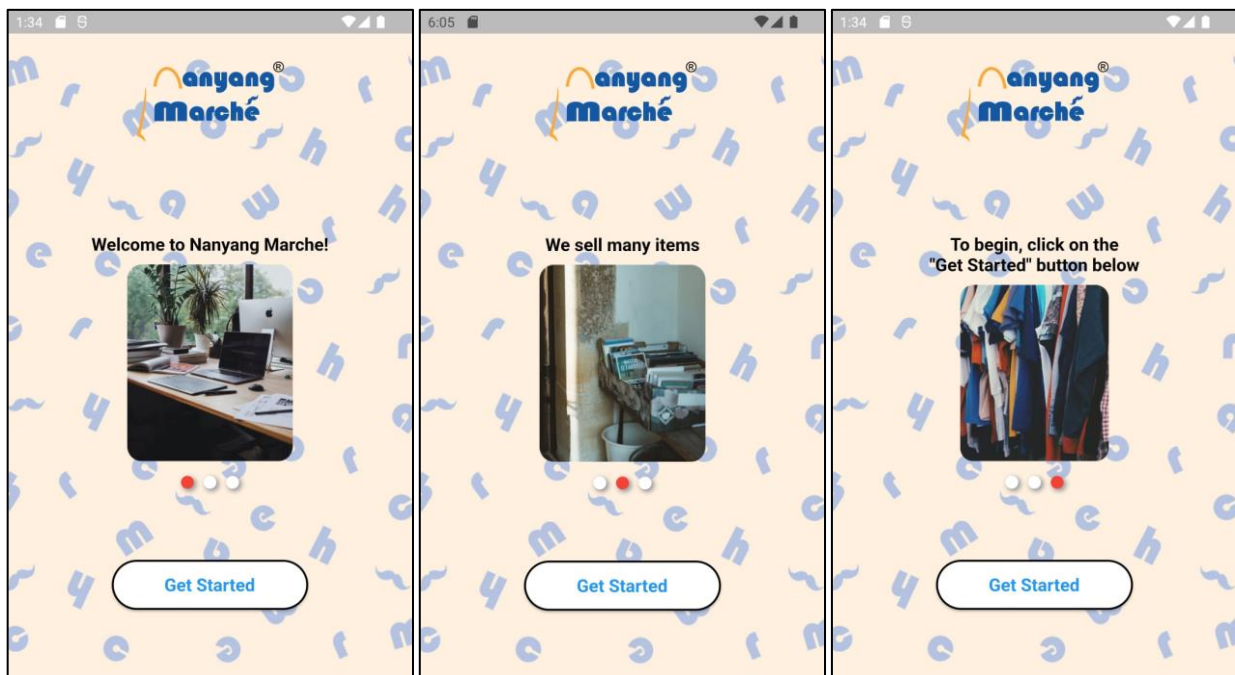


Figure 14 Onboarding Page (Actual Outcome)

8.2.2. Login Page

This page is of utmost importance especially since our application is targeting the NTU student community and authentication with the users' school email is essential to protect the user's data and unique identity.

Creating an account and logging in with their credentials also allows the ease of communication with prospective buyers/sellers.

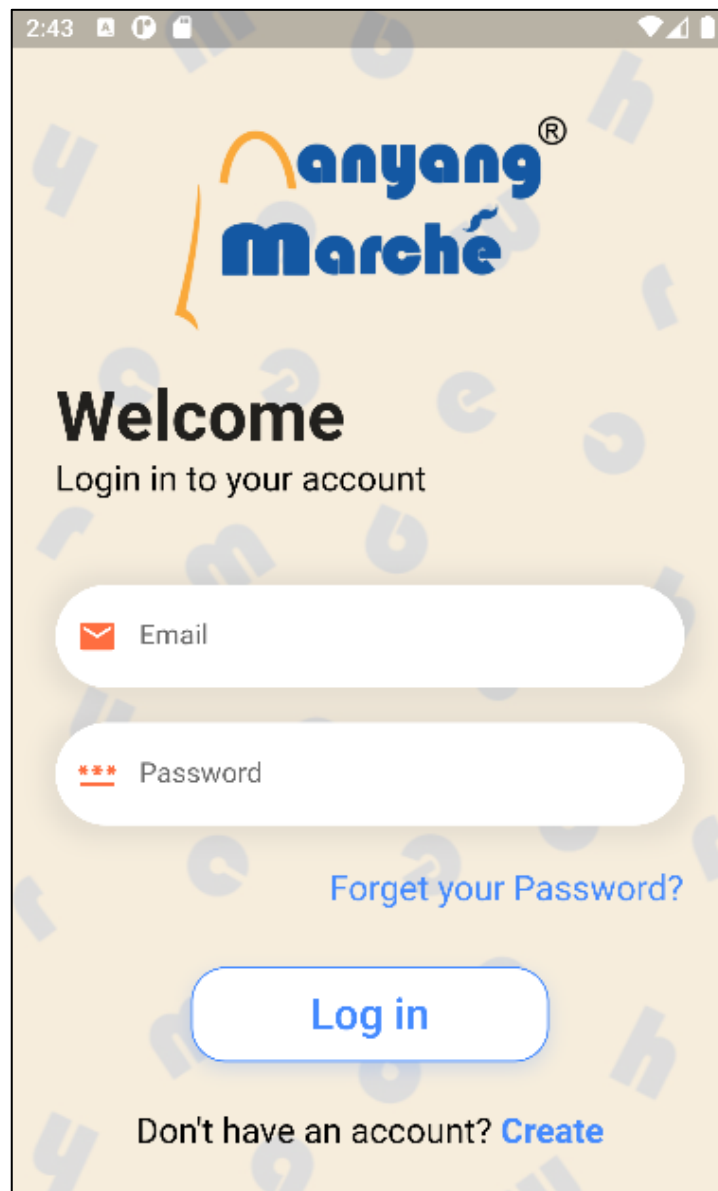


Figure 15 Login Page (Actual Outcome)

8.2.3. Sign Up Page

In this page, new users should provide their names, NTU emails and the password to create an account. The account will be stored and encrypted in Firebase.

Consider the consistency of the interfaces, the format of the signup page is in line with the log in page.

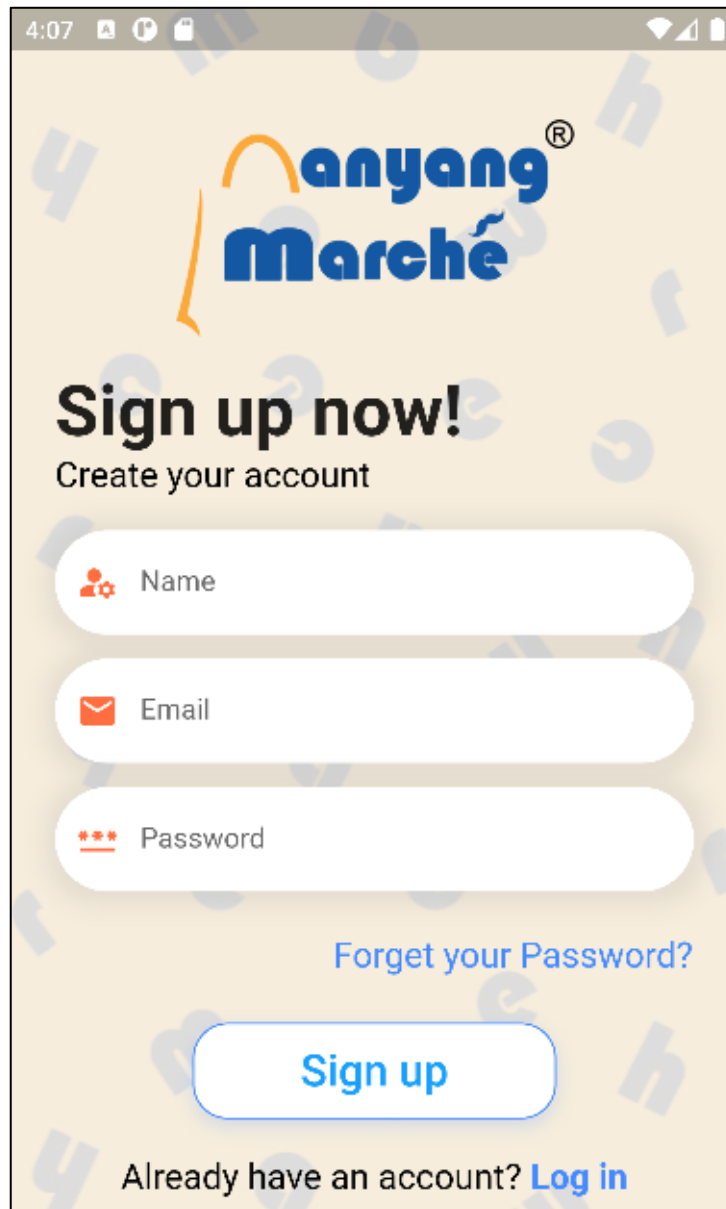
A screenshot of a mobile application's sign-up page. At the top, the status bar shows the time 4:07 and various icons. The app's logo, 'Anyang Marche', is displayed in blue and orange. Below the logo, the text 'Sign up now!' is in large black font, followed by 'Create your account' in a smaller black font. There are three white input fields with rounded corners: the first is labeled 'Name' with a red person icon, the second is labeled 'Email' with a red envelope icon, and the third is labeled 'Password' with three red asterisks. Below the password field is a blue link that says 'Forget your Password?'. At the bottom of the form is a blue button with white text that says 'Sign up'. Below the button, there is a line of text: 'Already have an account? Log in', where 'Log in' is a blue link.

Figure 16 Sign up Page (Actual Outcome)

8.2.4. Home Page

The home page serves as the marketplace, which is a visual representation of the different products available and includes a navigation bar at the bottom to allow for easy navigation across the different pages. It includes the categories section and the recommendation section.

In the categories section, users can swipe to see more, and in the recommendation section, users can swipe down to see more. In order to achieve this, we used the `ListView()`, `GridView.count()` and `SingleChildScrollView()` function in Flutter.

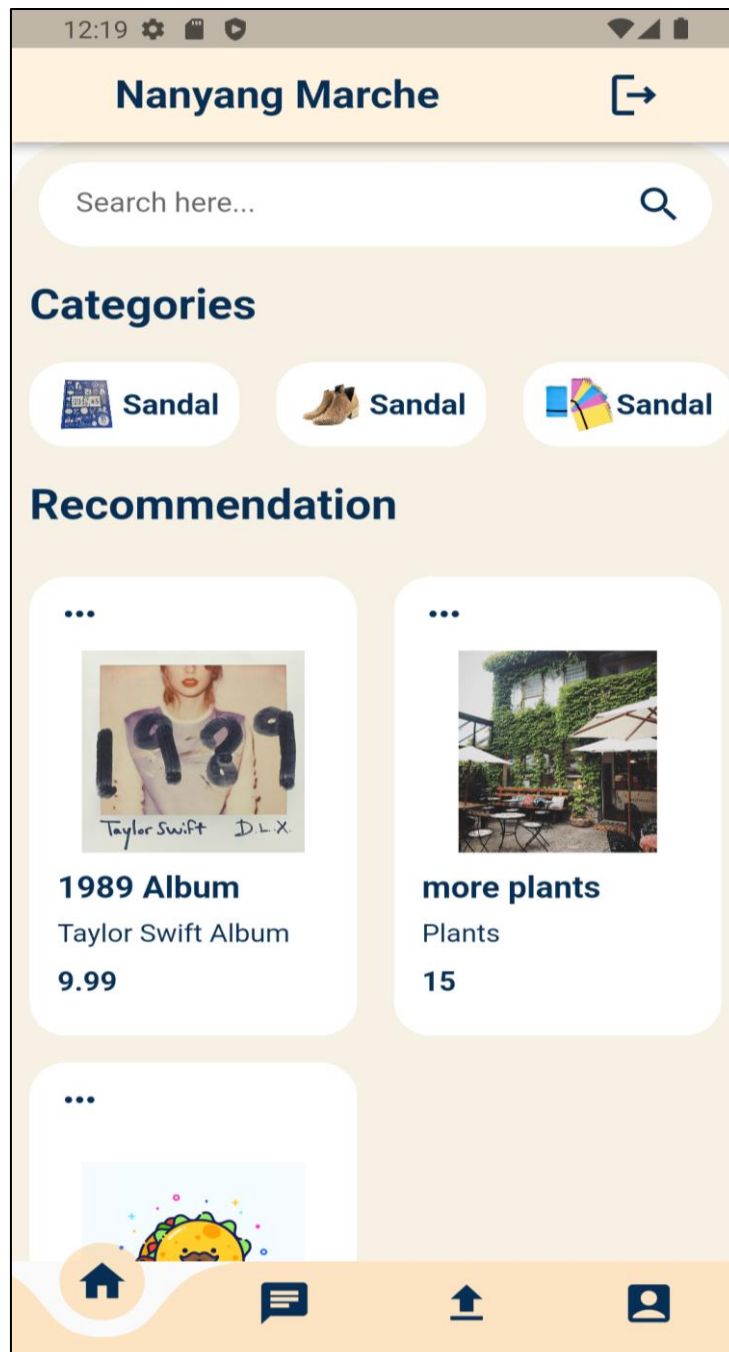


Figure 17 Home Page (Actual Outcome)

8.2.5. Upload Page

The upload page enables users to click a picture of their pre-loved product that they would like to sell and create a post, which includes the title, price, and detailed description of the product. When users press the post button, the post will be uploaded on the application marketplace.

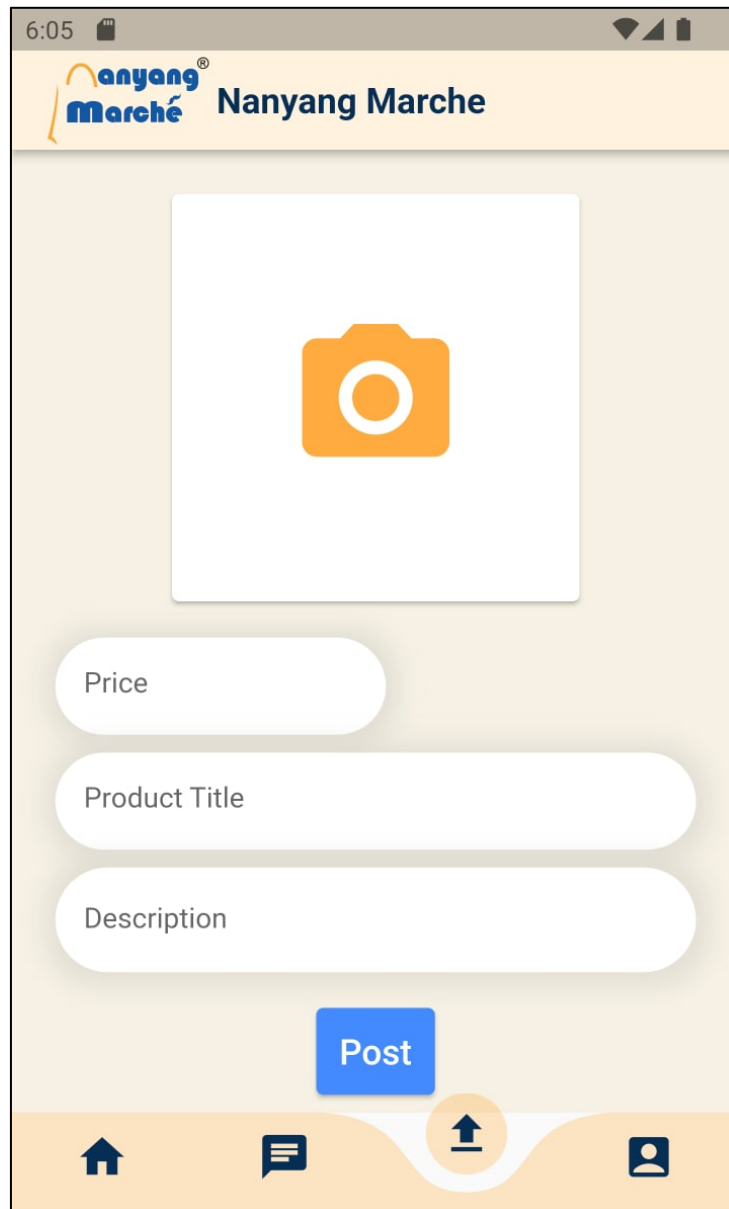


Figure 18 Upload Page (Actual Outcome)

8.2.6. Product Details Page

The product detail page contains information about the individual product such the condition and price. This page also includes a chat feature at the bottom of the page for the ease of communication with the seller if one is interested in knowing more or has the intention of buying the product.



Figure 19 Product Details Page (Actual Outcome)

8.2.7. Chat Page

As stated previously, the chat page serves as a platform for the interaction of the potential buyers and sellers of the products and has separate conversations with each person. The details of the chat function are shown in section 8.3.

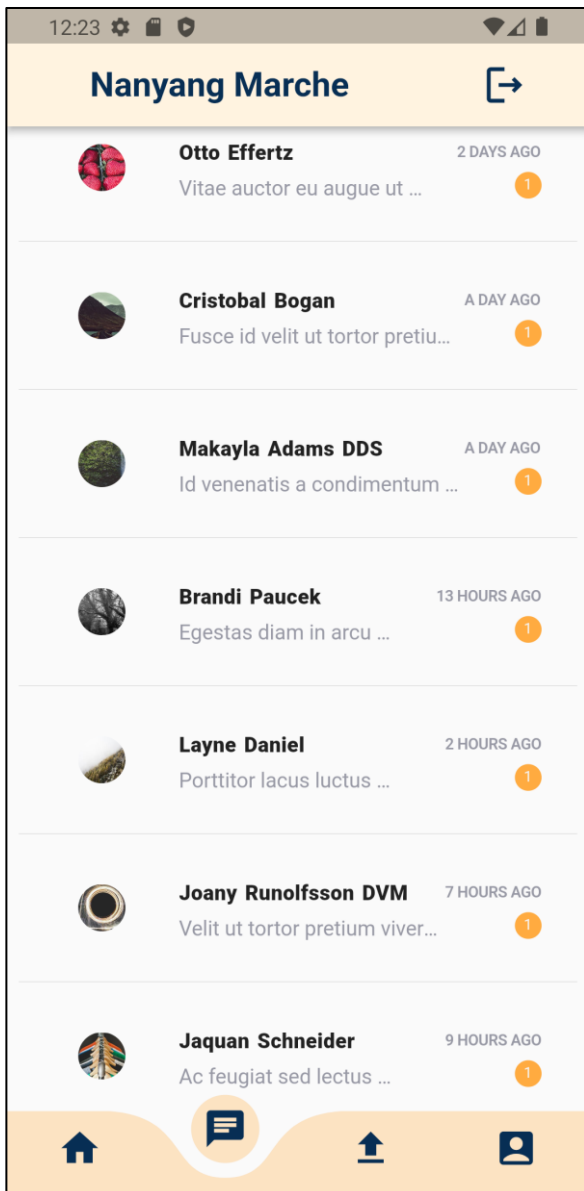


Figure 20 Chat Messages Page (Actual Outcome)

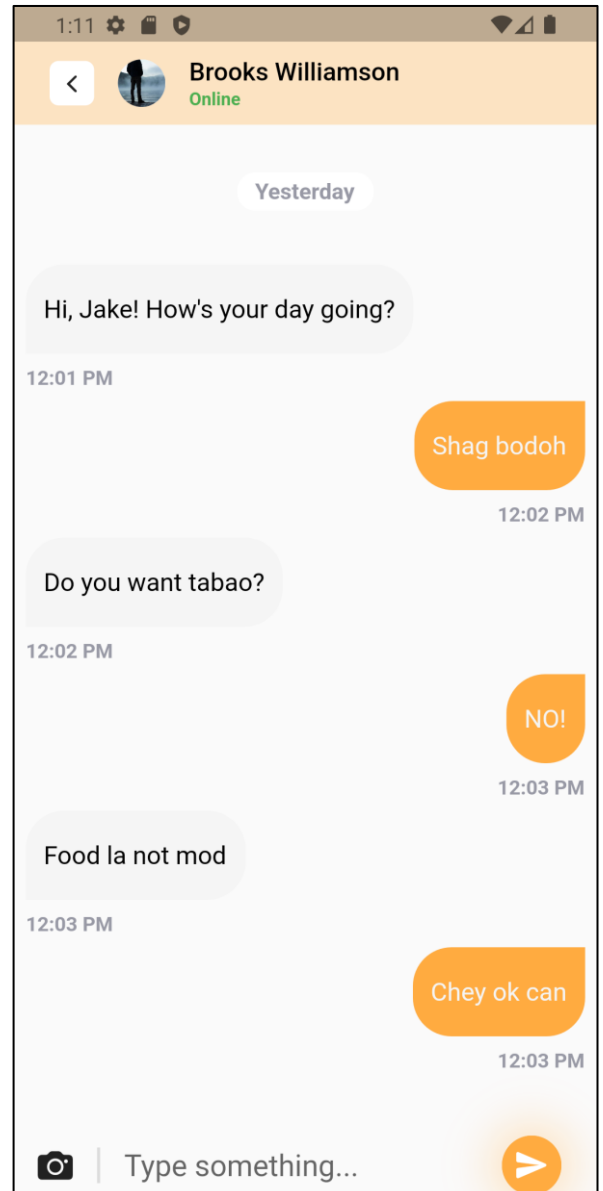


Figure 21 Chat Screen (Actual Outcome)

8.2.8. Profile Page

The profile page is a display of the listings of the user. Furthermore, in the upper right corner of the page, there is a logout button and by clicking it a pop-up window will appear for confirmation, which upon confirmation will then return users back to the login page.

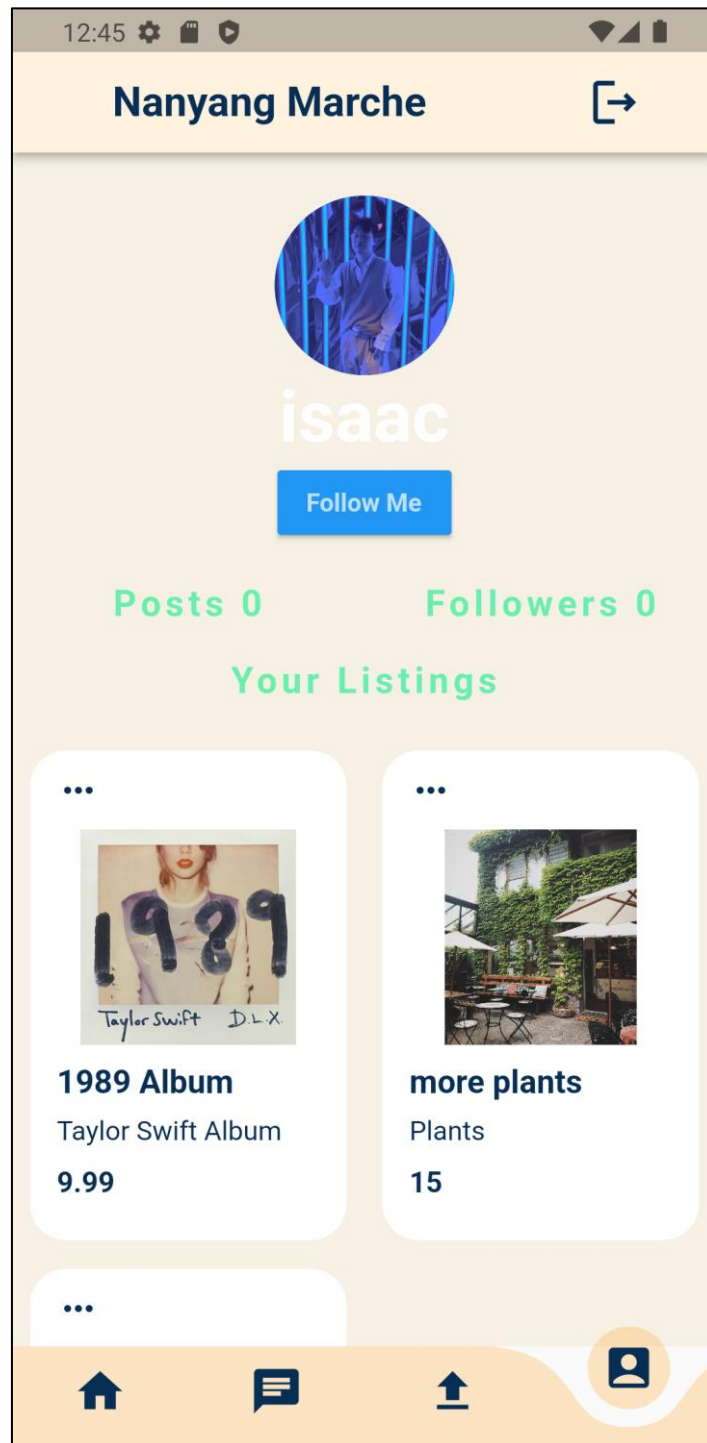


Figure 22 Profile Page (Actual Outcome)

8.3. Chat Function

Figure 19 shows the main chat messages page with included features:

1. Displayed usernames and profile pictures.
2. Scrollable chat messages page.
3. Chat preview with last message timestamps.
4. Number of unread messages badge.

Figure 20 shows the chat screen with included features:

1. Username, online status, and display picture in app bar.
2. Scrollable chat screen of message history with timestamps and receipts.
3. Typing section to input and send messages using the virtual keyboard.
4. Upload image button which allows images to be sent in standard formats (JPEG, PNG, GIF).

Tapping on a chat in the chat messages page shown in figure 19 brings us to the chat screen in figure 20. This is where users can communicate via text and send images, their online status is also reflected if their phone is connected to the Internet.

8.4. Navigation Bar

As shown above, there is a curved bottom navigation bar in the home page, chat page, upload page and profile page. Using this bar, we can simply switch between these pages. To achieve this, we use the packet “curved_navigation_bar” in Flutter.

```
@override
Widget build(BuildContext context) {
  return Scaffold(
    bottomNavigationBar: CurvedNavigationBar(
      backgroundColor: Colors.transparent,
      animationDuration: const Duration(milliseconds: 300),
      height: 60,
      color: Colors.orangeAccent.withOpacity(0.3),
      items: items,
      index: index,
      onTap: (selectedIndex) {
        setState(() {
          index = selectedIndex;
        });
      },
    ), // CurvedNavigationBar
  );
}
```



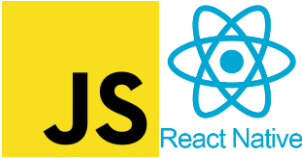




Figure 23 “curved_navigation_bar” in Flutter

9. Front-End Analysis

9.1. Introduction:

This section serves as a documentation and compilation of the various research, libraries, and services used in the entirety of this project. Implementations and the thought processes behind them will be elaborated on in detail in this section.

Our team first researched the different programming languages that could enhance the current UI of our e-commerce. We judged a total of 7 languages based on various criteria. The languages and their respective frameworks were C# with Xamarin framework, Dart with Flutter framework, JavaScript with React Native framework, Objective-C, Swift, Java and Kotlin showed below.

		
C# with Xamarin framework	Dart with Flutter framework	JavaScript with React Native framework
		
Objective-C	Swift	Java
		
Kotlin		

9.2. Criteria

We narrowed down the language by filtering it through each criterion listed below.

9.3. Learning speed and process

Given our limited knowledge of programming, we wanted a language that would be relatively easy to pick up, whilst not compromising on its effectiveness and flexibility. Comparing the learning speeds of most of these languages, this eliminates Objective-C and Java, they are a lot more difficult to learn compared to the other languages listed.

This leaves us with C# with Xamarin framework, Dart with Flutter framework, JavaScript with React Native framework, Kotlin and Swift.

9.4. Cross-Platform

Next, we wanted a language which could run cross-platform for ease of testing and possible deployment into the various app stores. This eliminated Swift and Kotlin as Swift could only produce apps for the iOS and Kotlin could only produce apps for the android OS. Similarly, Objective-C could only produce apps for iOS, and Java could only produce apps for the android OS.

This leaves us with C# with Xamarin framework, Dart with Flutter framework, JavaScript with React Native framework.

9.5. Performance

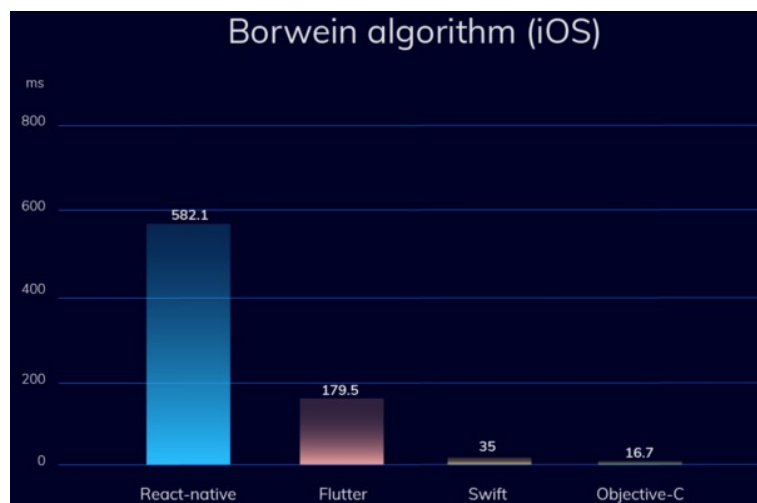


Figure 24 Performance of Frameworks on iOS

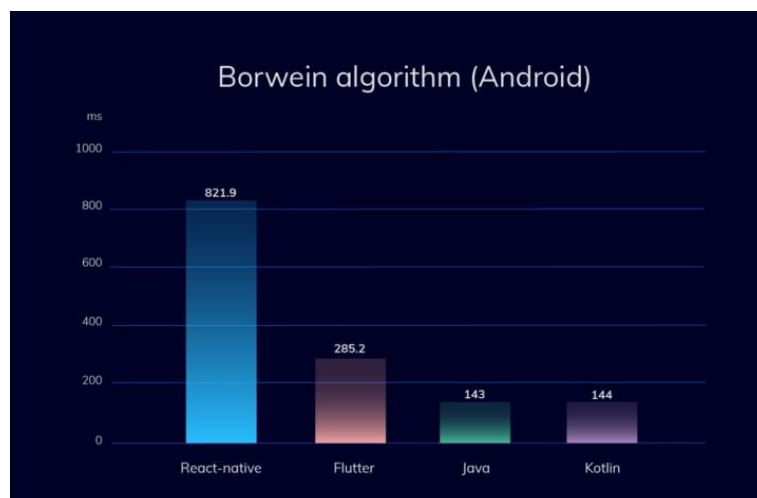


Figure 25 Performance of Frameworks on Android

In terms of performance, how fast the application runs and works, Flutter outperforms the other two frameworks Xamarin and React Native. It achieves near native performance and speed while the other two frameworks are much slower. Flutter achieves this by compiling directly to native code, machine code which could be Intel x64 or ARM, or JavaScript if it is a web application whichever is more suitable.

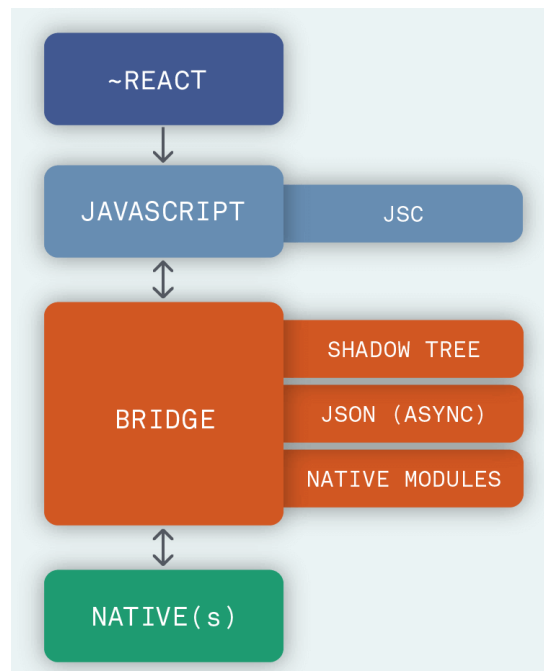


Figure 26 React Native Architecture and Bridging

Meanwhile Xamarin compiles to an Intermediate Language which will be compiled to native code using a Just-In-Time (JIT) Compiler and React Native relies on bridging between the native thread and JavaScript thread to communicate various logic using the JavaScript Interface (JSI) which leads to a slower performance overall.

9.6. Popularity and Community Support

The bar charts show Stack Overflow Survey for the Popularity and the Most Loved and Dreaded of Frameworks in year of 2022 under 54,414 and 53,442 all responses in the Figure 33 and 34 below.

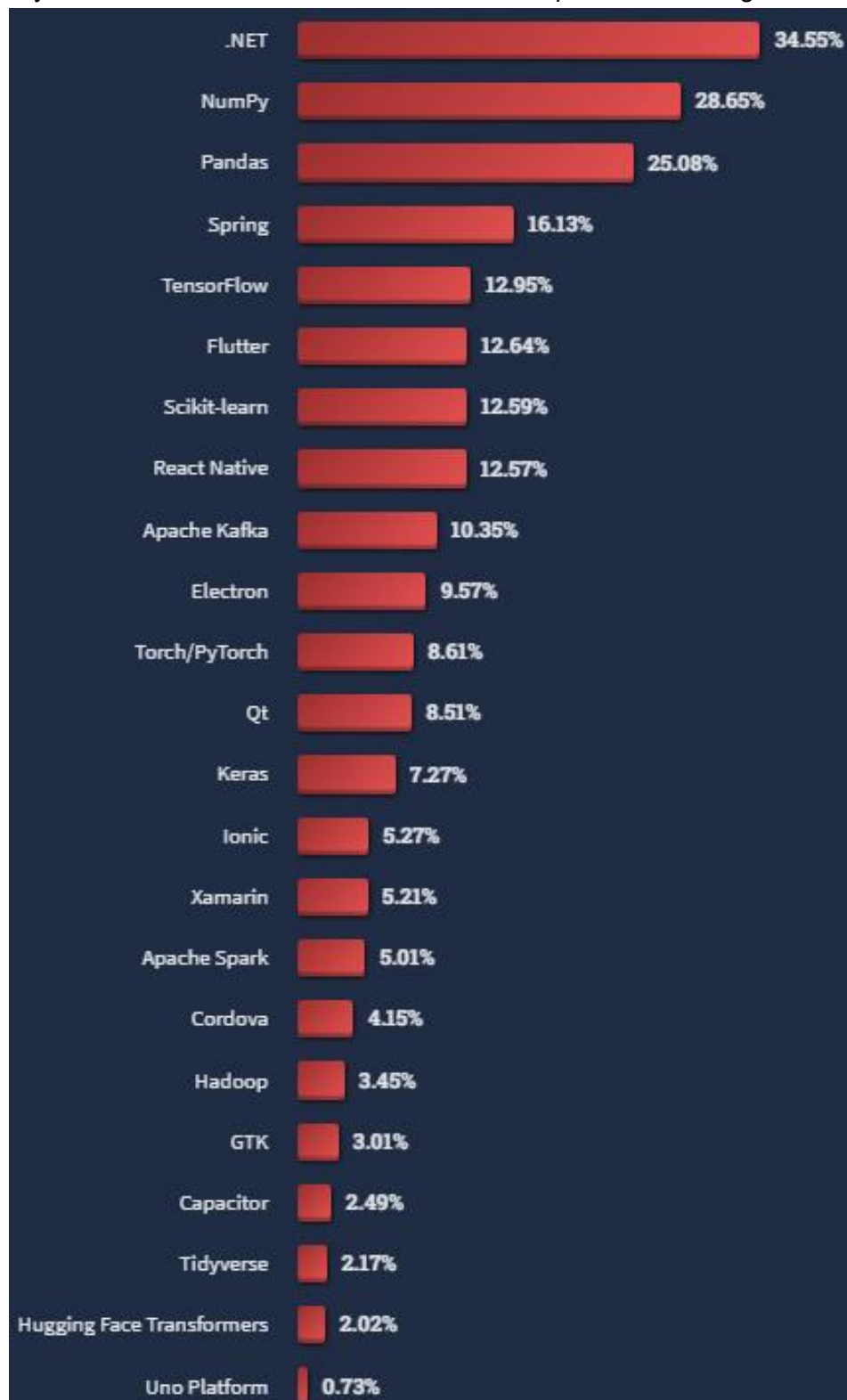


Figure 27 Popularity of Frameworks in 2022

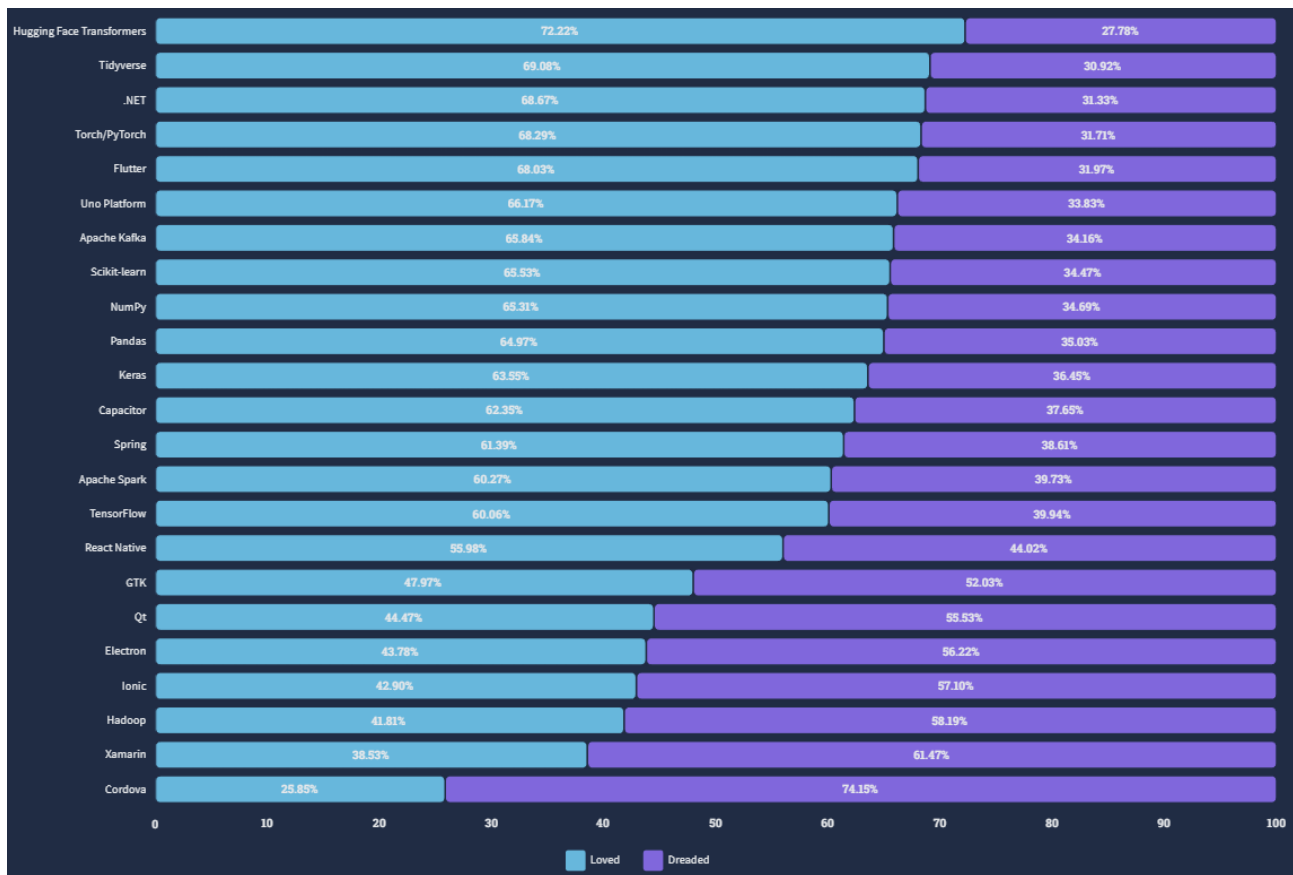


Figure 28 Most Loved and Dreaded Frameworks in 2022

Currently, React Native has the largest user base and maturity of community support, but Flutter has been shown to have quick adoption rate and growing community support.

Xamarin, however, has been overtaken by the other two frameworks despite being developed much earlier.

The importance of popularity and community support is important due to the maintenance of the codebase and dependencies, if the support for the framework is non-existent, most of the dependencies we draw would be poorly documented or outdated.

Therefore, we decided to eliminate Xamarin and choose between React Native and Flutter.

9.7. User Interface and Plugins

For React Native, they use bridging technology first to connect the JavaScript thread to the main thread which contains the native components, which allows us to view the apps created with React Native similar to those written natively.

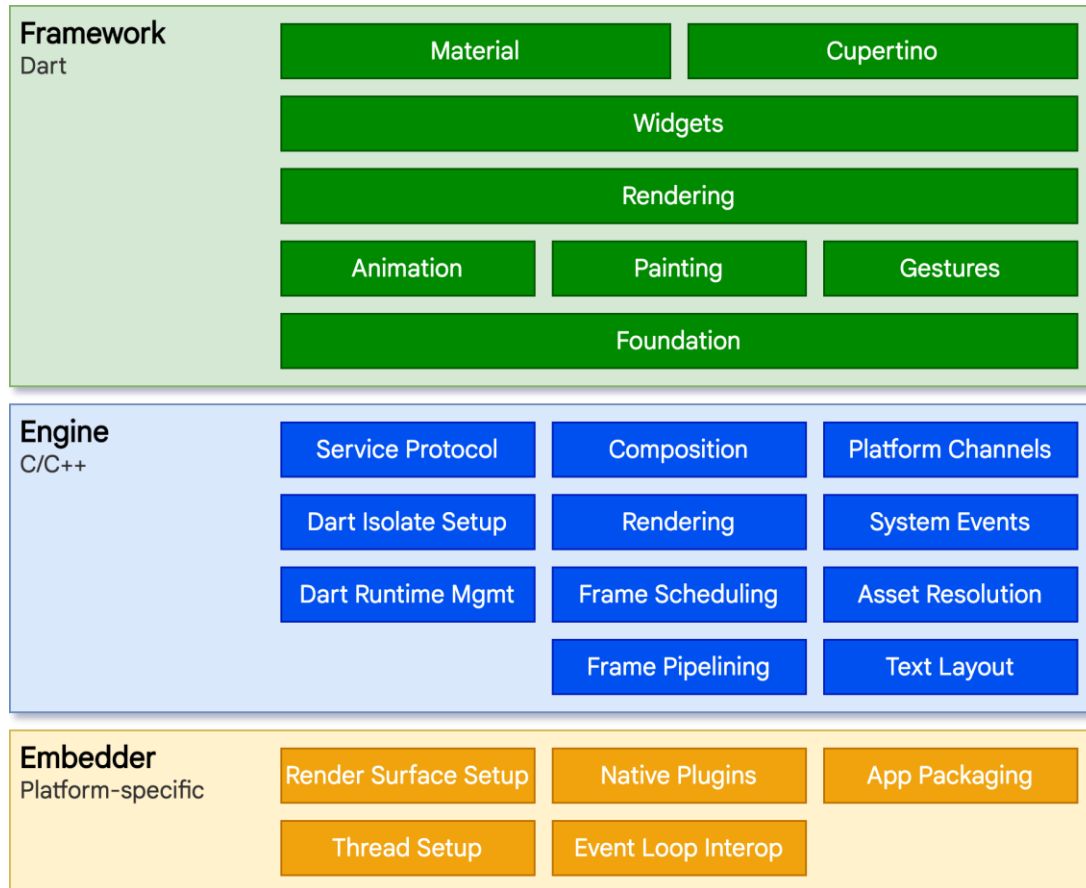


Figure 29 Flutter Architecture

For Flutter, they have an engine built with C++ and SKIA which renders its own custom pixels which perfectly copies those native iOS and android widgets and allows the flexibility of creating your own custom widgets, and they have something known as platform channels that communicates with the native system in case some native code is needed to render a component not found in Flutter.

While React Native relies heavily on third-party support, Flutter has more in-built features and plugins created by the Flutter team. Thus, we view it as React Native having more flexibility and leanness, while Flutter has more reliability and consistency.

9.8. Documentation

Due to the third-party nature of React Native, the documentation for some dependencies is not as clear as Flutter's, which is a lot more organized and understandable as most of their plugins are created by the Flutter team.

9.9. Our Choice

Therefore, based on all the above criteria, we choose Dart with the Flutter framework, a popular mobile framework which can develop high-performance cross-platform apps which use plugins that are more reliable and have easier documentation.

Flutter is an open-source UI software development kit created by Google which uses “Dart” as its main programming language. Flutter can be deployed on both Android and iOS devices, which we consider to be an extremely convenient feature, as we would want to develop an app that caters to as many users as possible. On top of its convenience, Dart’s syntax is relatively easy to understand and write. Furthermore, the hot loading feature available in Flutter helped us to view the UI changes instantly, improving the developer experience tremendously.

10. Back-End Analysis

This chapter serves as a concise report of the work the backend team had done. This section includes our research, code snippets, and reveals our thought processes behind the implementation of the codes.

10.1. Version Control

We decided to use Git as our version control software and GitHub as our online repository due to the multitude of functions it provides and the simplicity of its Graphical User Interface (GUI) and how we can pull and push files with ease using the in-built Git command-line interface (CLI).

10.2. Firebase

As our app has uploading and retrieving features, we needed an online database to be able to handle read-write operations. Since we were using the Flutter framework, it was a clear decision on which Back-End we should rely on. We chose Firebase as it was also developed by Google and had direct integration with Flutter which made it easier to implement them into one another. Firebase is a backend as a service (BaaS) which provides various services like Realtime Database, Crash Reporting, Authentication, Cloud Storage, Hosting, etc. It provides real-time updates and has a free tier, which is excellent for our project.

We first created a Firebase account that will store the database for this project.

To add Firebase into our Flutter project, we first must run FlutterFire, which is a plugin for Flutter to connect to Firebase.

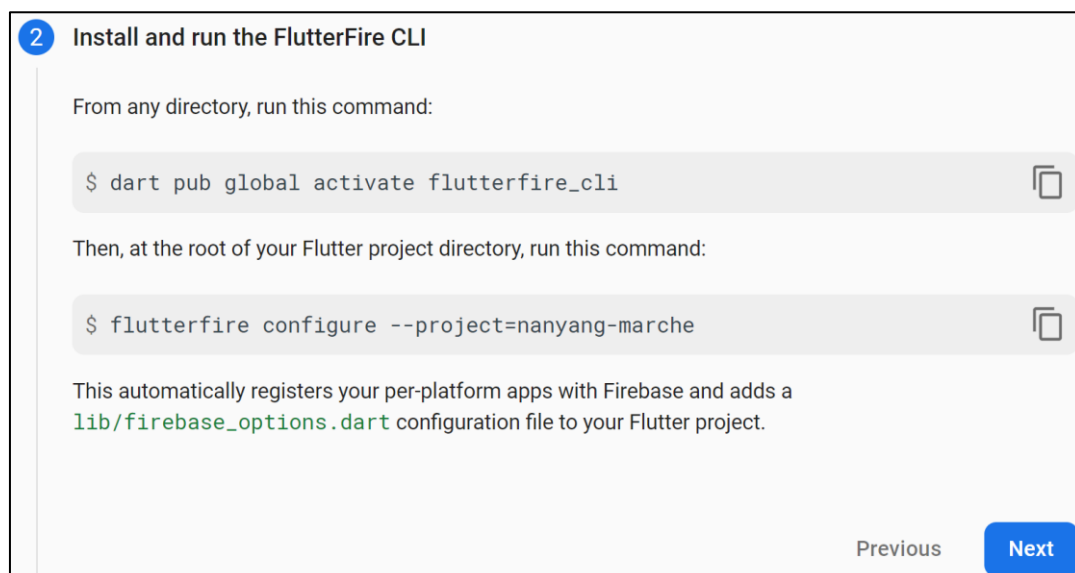


Figure 30 FlutterFire CLI

Afterwards, we must manually add this file created by Firebase into our project.

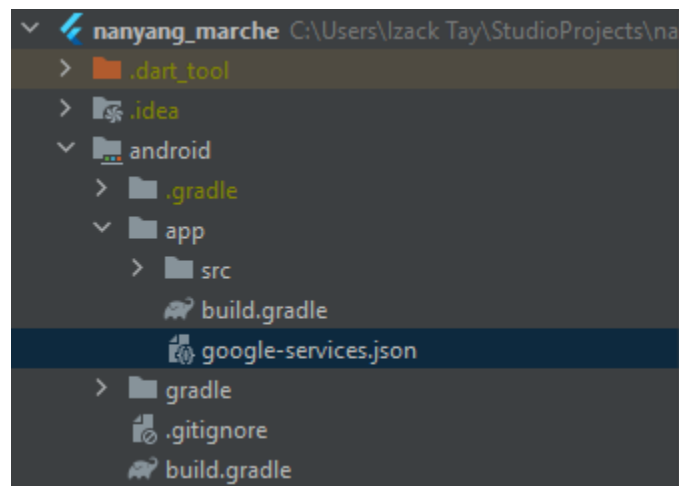


Figure 31 Project folder Structure in Android Studio

Lastly, we have to import dependencies that our app uses, in the “pubspec.yaml” file.

```
firebase_core: ^1.22.0
firebase_storage: ^10.3.8
firebase_auth: ^3.9.0
firebase_database: ^9.1.4
cloud_firestore: ^3.5.0
```

Figure 32 Import dependencies in Android Studio

The services of Firebase are now at our disposal!

10.3. Cloud Firestore

We chose to use Cloud Firestore, a database optimized for mobile applications, instead of Firebase’s original Realtime Database. Cloud Firestore has better scalability and better organization of data and better querying experience due to the structure. Cloud Firestore also offers multi-regional support, so the data received is likely to be the latest.

Cloud Firestore is a NoSQL database which means that data is stored in the form of JSON documents under a collection.

This means that while there is an increase in flexibility and adaptability in terms of data storage and querying functions and inputs, there is also the downside of a NoSQL database having a lack of a consistent structure, and relationships are harder to define.

This means that figuring out the relationship between the documents is crucial and a proper schema in the Firestore should be planned and implemented properly.

10.4. Schema Creation

We first created a “users” collection, to store their different user IDs. Each user collection contains the user’s basic sign-in information, and a unique user ID in the form of fields.

As user to products has a one-to-many relationship, we added a collection named “products” for each user.

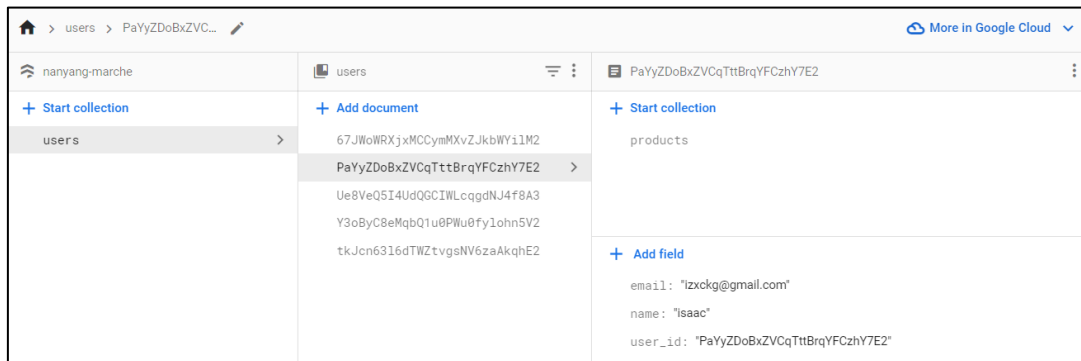


Figure 33 Users collection in Cloud Firestore

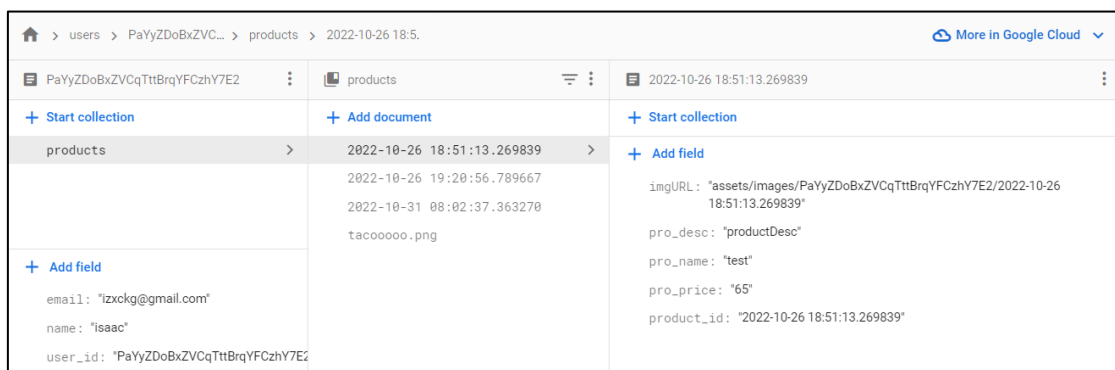


Figure 34 Products collection in Cloud Firestore

For these operations, we require a class to store Database related functions, and execute the functions wherever and whenever needed.

```
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:nanyang_marche/models/users.dart';
import 'package:nanyang_marche/models/products.dart';

class DatabaseManager {

  final CollectionReference userColl =
    FirebaseFirestore.instance.collection("users");
  static String? usr_id = FirebaseAuth.instance.currentUser?.uid.toString();
  static String? usr_id_doc = FirebaseFirestore.instance.collection("users").doc(usr_id).toString();

  Future<void> createUserData(String name, String email, String uid) async {
    await userColl.doc(uid).set({'name': name, 'email': email, 'user_id': uid});
  }

  Future createItemData(String p_id, String p_name, String imgUrl, String p_desc, String p_price, String uid) async {
    await userColl.doc(uid).collection("products").doc(p_id).set({'product_id': p_id, 'pro_name': p_name,
    'imgURL': imgUrl, 'pro_desc': productDesc, 'pro_price': p_price, 'user_id': uid});
  }
}
```

Figure 35 Functions to access Firebase using Flutter

These functions tap into the Firebase database, and either update existing, or create new JSON objects.

With each new sign up or upload of a product listing, a new entry is added, and the new user or product is created in their respective collections.

10.5. Firebase Storage

To allow the uploading of images from users, we utilized Firebase Storage, a cloud storage option in firebase. Images uploaded by users will be saved in the cloud servers, with its directory named as the user's ID, for ease of access.



Name	Size	Type	Last modified
1989.png	518.23 KB	image/png	Nov 1, 2022
2022-10-26 18:51:13.269839	195.93 KB	image/jpeg	Oct 26, 2022
2022-10-26 19:20:56.789667	195.93 KB	image/jpeg	Oct 26, 2022
2022-10-31 08:02:37.363270	195.9 KB	image/jpeg	Oct 31, 2022
coffeehouse.png	428.79 KB	image/png	Nov 1, 2022
tacooooo.png	59.12 KB	image/png	Oct 27, 2022

Figure 36 Images uploaded by user to cloud servers

The retrieval of images can be performed by using in-built Firebase functions to generate an image URL, and then populate an image widget accordingly.

```
String url = "assets/images/" + DatabaseManager.user_id! + "/";
Future awaitLists = FirebaseFirestore.instance
  .collection("users")
  .doc(FirebaseAuth.instance.currentUser?.uid.toString())
  .collection("products")
  .get()
  .then(
    (res) async {
      for (var element in res.docs) {
        final imgURL = await FirebaseStorage.instance
          .ref()
          .child(url + element.id.toString())
          .getDownloadURL();

        try {
          imgURL_list.add(imgURL);
          name_list.add(element.get('pro_name'));
          desc_list.add(element.get('pro_desc'));
          price_list.add(element.get('pro_price'));
        } on FirebaseException catch (e) {
          // Handle any errors.
        }
      }
    }
  )
}
```

Figure 37 Accessing Firebase Storage using Flutter

10.6. Authentication

Authentication from Firebase allows us to perform access control and validate our users without the hassle of writing our own authentication code.

A class is created for basic services that require authentication like logging in, signing up, and accessing the current session's User ID.

```
class AuthenticationService {
  final FirebaseAuth _auth = FirebaseAuth.instance;

  var usrId = "";
  // registration with email and password

  Future createUser(String name, String email, String password) async {
    try {
      UserCredential result = await _auth.createUserWithEmailAndPassword(
        email: email, password: password);
      User user = result.user as User;
      await DatabaseManager().createUserData(name, email, user.uid);
      return user;
    } catch (e) {
      print(e.toString());
    }
  }

  Future createNewItem(String p_name, String imgurl, String p_desc, String p_price) async {
    final user = _auth.currentUser;
    final uid = user?.uid;
    var p_id = DateTime.now().toString();
    await DatabaseManager().createItemData(p_id, p_name, imgurl, p_desc, p_price, uid!);
  }

  // sign with email and password
  Future loginUser(String email, String password) async {
    try {
      UserCredential result = await _auth.signInWithEmailAndPassword(
        email: email, password: password);
      return result.user;
    } catch (e) {
      print(e.toString());
    }
  }
}
```

Figure 38 Authentication using Flutter

10.7. Upload

This function executes whenever a user clicks on the “Post” button. It has 3 mandatory parameters, product name; product description; product price, to ensure that the essential information about the listing will be uploaded & displayed clearly on the home page.

```
Future uploadItem(p_name, p_desc, p_price) async {  
  try{  
    var picId = DateTime.now().toString();  
    var img_url = "assets/images/"+usrId!+"/"+picId;  
    var uid = usrId;  
    final imagesRef = storageRef.child(img_url);  
    await imagesRef.putFile(File(image!.path));  
    DatabaseManager().createItemData(picId, p_name, img_url, p_desc, p_price, uid!);  
  }catch(e){  
    print("Did not upload");  
    print(File(image!.path));  
  }  
}
```

Figure 39 Upload function using Flutter

10.8. Chat

The backend implementation of our chat function was one technical hurdle we faced. We initially considered building chat in-house using Cloud Firestore, but we discovered the Stream's Chat Messaging Application Programming Interface (API) which enables developers to quickly add chat functionality to their application saving hours of time spent on backend integration. We describe the pros and cons followed by the integration and application of the respective methods below.

10.8.1. Cloud Firestore for Chat implementation

In addition to the user's collection that stores the user data, a messages collection is created to store message data between our users. Each respective user's conversations are stored in their respective subcollection as seen below:

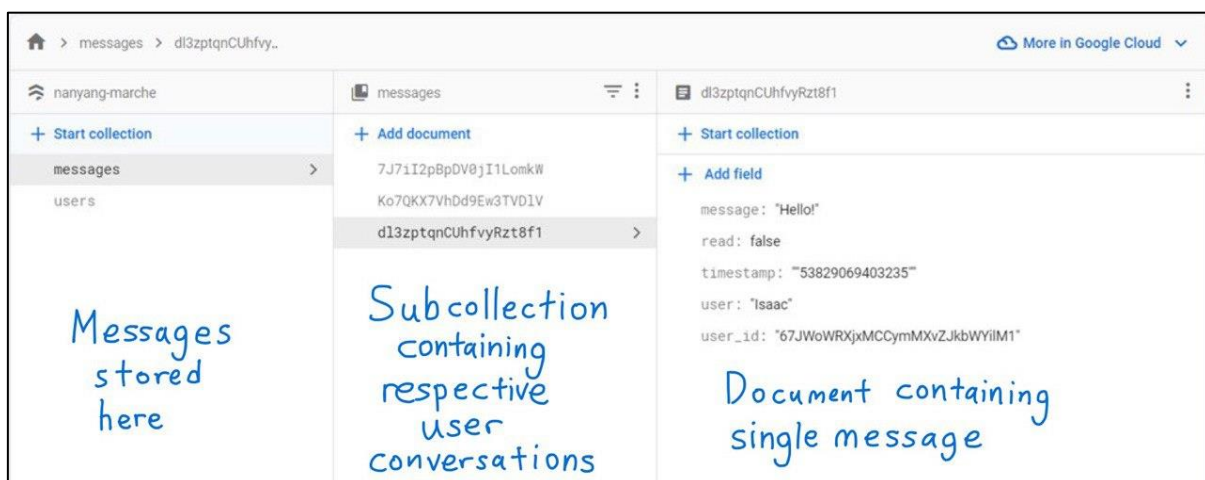


Figure 40 Actual Chat Messaging Cloud Firestore Database

The final database, when fully implemented, should look like the updated figure below:

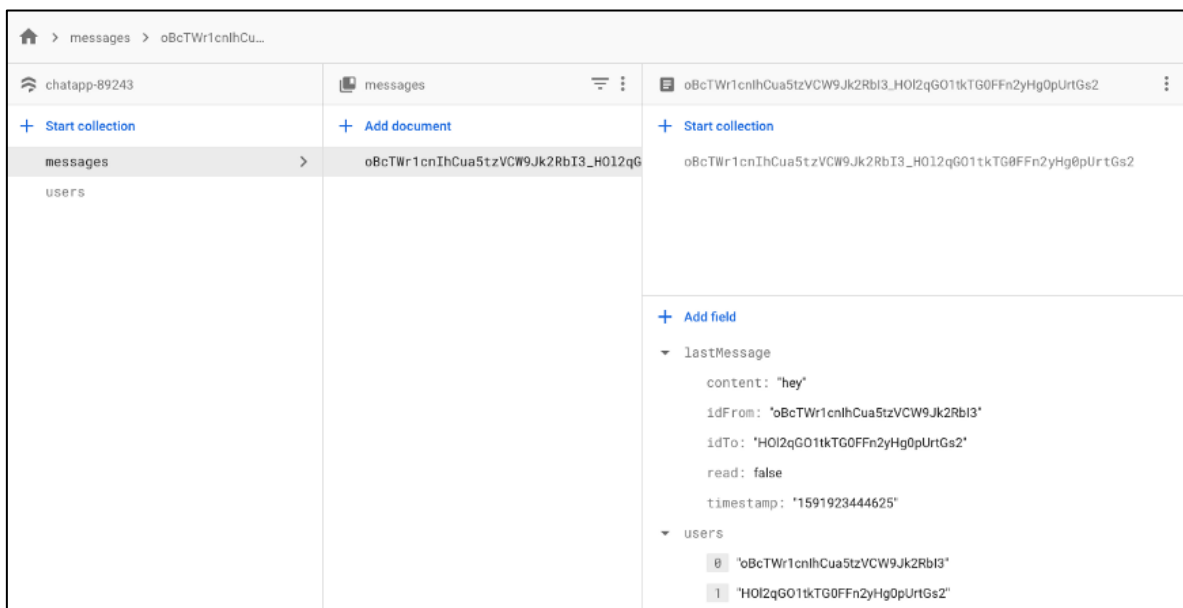


Figure 41 Expected Chat Messaging Cloud Firestore Database

Unfortunately, after countless weeks of troubleshooting, integration of the in-house chat function has not been very successful due to repeated random crashes and errors encountered. Hence, we look at the advantages of using the Stream Chat Messaging API below.

10.8.2. Stream Chat Messaging API

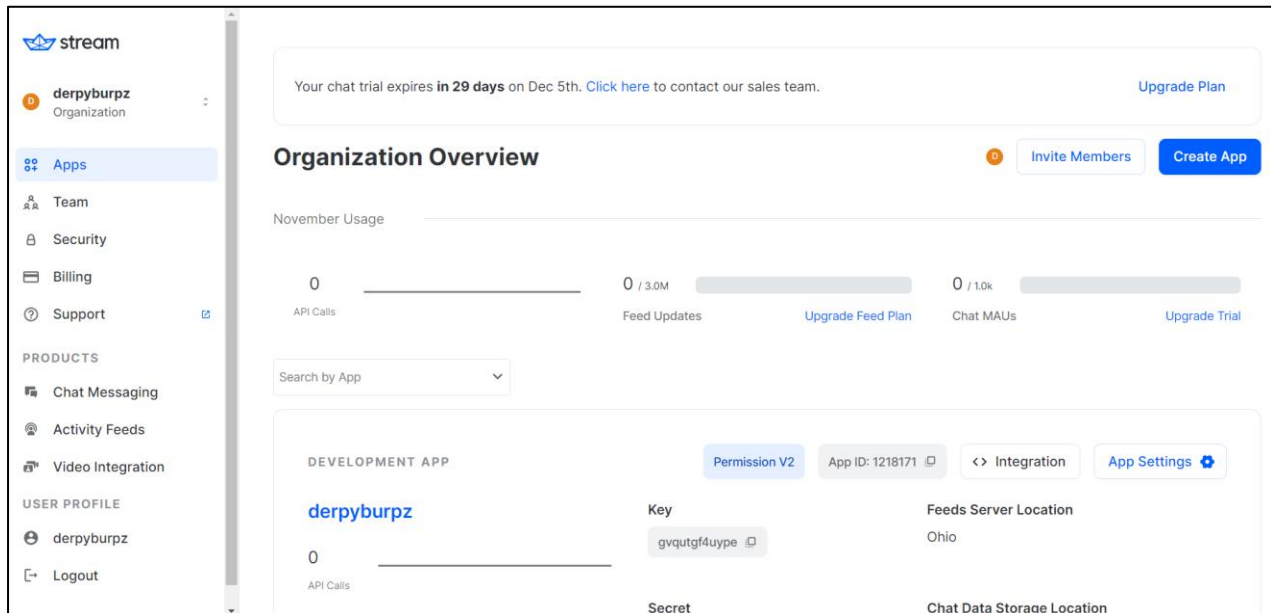


Figure 42 Stream Chat Messaging API

Stream Chat is a tool that provides the API and UI SDKs for custom messaging applications by providing a reliable chat infrastructure for all major platforms.

Perks offered by the API include:

- Performance and Scalability
- Automated Moderation (Profanity filter and image moderation)
- Built in local caching (Faster loading)

10.9. Actual Outcome

While the Stream Chat Messaging API is easier to implement, we struggled to refactor the existing codebase to accommodate this new implementation.

We did not manage to add the following features:

1. Realtime notifications when receiving messages.
2. Displaying real time online status when chat tab is open with an option to set it as invisible.
3. Message receipts synced with Cloud Firestore.
4. Fully functional chat utilizing either only Cloud Firestore or a combination of Cloud Firestore and Stream's API.

The team had many features we wanted to implement into the chat such as image uploading and online status with full backend support. However, due to time constraints, backend features were not implemented. Hence, the chat function is only functional at the front end.

11. Future Improvements & Recommendations

Firstly, we would want to release the app with all the features we mentioned in our purpose and scope to its optimal extent.

11.1. Profanity Filter

Next, to take the application to greater heights, our team also foresees the need to have a way to moderate the listings posted by students in the application. Furthermore, we feel that it is necessary to allow the filtering of any explicit content in the marketplace and chat to create a safe and conducive platform for NTU students.

11.2. Quality of Life Features

One new feature which could be implemented is a request feature which is a forum platform for students to request items which are unavailable on the marketplace. Other features, like supporting various payment methods or a GPS locator, can be added. This can make our application more convenient, but it comes with potential privacy and security risks, which requires additional security measures to be implemented such as two-factor authentication and securing backend connections.

11.3. Other Recommendations

Other teams could secure industry sponsorships for our app since it only caters to NTU community. This app could also go further by contacting the admin of the telegram channel NTU Marketplace, so that they can be included in the continuation of this app and its future improvements.

This app could even be further expanded and perhaps Nanyang Marche can someday do brand collaborations with big partners like Shopee, Lazada or Grab.

12. Individual Reflection

12.1. Isaac Tay Peng Bin

Over the course of the 13 weeks, I am elated to say that I have learned many things in this process of app development. These are skills I hope to refine on, so that I can bring more value in future projects I will be working on.

As I was taught a good amount of app development skills back in my Polytechnic, I was confident that I would not have difficulties working on this project. However, I have since realized that as new innovations are introduced, there will also be new ways of writing code. In other words, it was apparent to me that some of the things I've learned were no longer in practice, and I would have to do my own research all over again.

Flutter's programming logic felt very foreign to me, and I always end up with errors when testing. The research was intense, as there were a lot of gaps in my knowledge about this development kit. In many instances, it felt as if I was going down a rabbit hole, simply trying to understand how a few lines of code were supposed to work.

The epiphanies came when I was trying to integrate Firebase with Flutter. After watching a series of tutorial videos, the logic of the codes managed to fall in place in my mind, all at once. I was relieved to have finally built a working authentication feature which includes registration. The rest of the Firebase features of the app were built by reading documentation provided by their website, albeit there were a few hiccups because there was missing documentation.

Afterwards, I've worked closely with my teammates: they would send me dart files containing UI codes and I would simply add the backend functions after, as well as compiling all the pages together, ensuring a smooth experience.

I would like to express my gratitude towards each and every single one of my teammates, who had made the process of creating an app from scratch possible. Our work was always distributed according to our strengths, so everyone can have a chance to work on the project. The front-end team came up with unique designs and explored color theory for a pleasant user experience. The back-end team whom I can always have discussions on how to further improve the syntax of my codes. My teammates had constantly shown passion and interest for this project, which made the journey of developing the app pleasant.

12.2. Chai Jia Zhe

This is my first time working on a team project of this size. Previously, I had only ever done small projects that can be completed in a few weeks, or projects in groups up to 4 people. I have only had experience doing basic programming in C++. Unfortunately, due to unfavourable circumstances, I ended up joining this project where app programming is the focus. My lack of programming and design knowledge made this an unfavourable placement for my DIP.

Having limited experience in both the design and programming aspects, I did my best to come up with some initial ideas, give suggestions along the way and in general see what small areas I could help with. At the start I tried to learn Dart along with the rest of the team. However, it proved not to be viable for me with the heavy academic load of my other ongoing modules, as well as having to learn python at the same time for another programming module. I was able however to help with proofreading of the final report and help to edit minor mistakes that were overlooked when they were first typed out.

It has been an eye-opening experience to see how a team can work together to develop an app. Although I did not gain a lot of technical skills and experience, I did learn a lot about working as a team. Overall, this has been an enriching experience for me, and I now am more comfortable with working in larger groups. I can also appreciate the effort that goes into the development of apps that I take for granted in my daily life.

I would like to thank my teammates for all the effort that they have put into this project. I also really appreciate my teammates' patience when dealing with my lack of knowledge and experience. These last 13 weeks have highlighted areas that I need to work on, including my communication and interpersonal skills.

I would also like to thank our professor Ling Keck Voon for checking in on us and imparting his knowledge and experience. I feel that the conversations we had with him, though limited in number, were enlightening,

12.3. Stanley Wong Qi Ren

After going through this Design and Innovation Project, I am happy to say that I have learnt a lot about mobile development. The project gave me insights into software development and application design, an area which I am interested in and hope to explore and effectively utilize my skills.

Going into the project, I had prior knowledge of web, mobile and software development and have done projects that require these skills. I had multiple paths to choose from, either we could build a website and then convert it into a mobile application later using tools like Ionic and Electron, but I decided against it due to the level of experience of my team, which I also considered when choosing the language and framework for this mobile app. I chose Flutter as it would be a suitable mobile framework due to its gentle learning curve and integration with Firebase which would serve as our backend.

Throughout this project, due to the various features we have seen in the UI design in Figma, I continue to learn more about Flutter and the various plugins available so that I could implement these in the app, for example, the image picker plugin which allows the app to access the camera and gallery of the phone or the carousel slider which allows us to swipe and display various screens and the use of asynchronous functions to get image data from the camera or gallery.

Since I had prior knowledge on mobile development, I took on a mentoring role in this project, where I will help and guide my fellow teammates in the development of applications and complete each component after reviewing the code. I also aided in the project where I see help is required, for example, switching development to pages that need help progressing.

I really appreciate my teammates willingness to learn and cooperate, along with their ability to learn quickly and apply the knowledge they have gained to successfully create this application. I think the project is an overall success.

12.4. Rithikha Vivekanantham

This entire project module has really been a place for me to hone my skills in both working on UI/UX design and app development skills. I am glad to have improved these skills because application development has always been something in which I have been interested in getting more involved and this definitely would help me in future projects. Having an interest in programming as well as UX/UI design enabled me to work on both aspects together since they are interlinked.

Having some previous experience in web development and Figma design initially made me think that creating a mobile application would not be very different from what I had done before. However, learning the flutter framework from almost having no prior knowledge about it and learning it through the videos has taught me a lot more about the structure of writing the front-end codes for our app. Moreover, even though Figma was something on which I have some knowledge on, it was really enriching to work with some of my group members who also have an inherent aesthetic eye towards details.

While working on this project, it initially took quite a long time for us to get more used to the different functions in Figma and learn how to use them to better present our prototype. Even with some experience, there were and still are quite several functions in Figma which I am not completely familiar with. Furthermore, our Figma prototype as observed from the figures in the UX section consists of approximately 14 to 15 wireframes for each of the pages and this entire process was very timing consuming, but it was really worth the effort since we are proud of our final end product.

Furthermore, we also wanted to use a Figma to flutter plugin for the convenience of converting our desired design into the actual code. However, our team faced quite a few obstacles using this since it was difficult to integrate the backend to the code due to the large number of widgets utilized. Hence, we decided to write the code from scratch instead for the front end which took up the middle stage of our project and it was a success when our final app was very similar to our design prototype in Figma.

In conclusion, I am truly grateful to my team members for advising and helping me out whenever I face difficulties in solving a particular issue while working on the project. I would like to also take this opportunity to thank our supervising professor for asking us thought provoking questions so that we would think more critically about our project and come up with a well-thought out idea.

12.5. Ke Tangxin

As a person who is interested in programming and developing apps, I have learned a lot from this project. I not only deepened my knowledge of UI/UX design and software development, but also knew better how to conduct team cooperation and communication. This is a precious experience for me, and it will certainly help me a lot in my future projects and career.

In the project early, I chose to work on the UI/UX part, to be a frontend coder, because I was more interested in it and had some experience in design before, which I thought might be helpful to our project. Having no prior knowledge about Flutter framework and Dart languages, during the first few weeks, I learned and practiced by following YouTube videos and some websites that teaching programming. I also work with other frontend contributor closely, after drawing the draft of our wireframes, we decided to use Figma to preliminary show the whole UI/UX, it will help us to stand in the angle of the user to view our pages, based on the concision, aesthetics, functions, and user habits. I also set the color tone throughout the program to make it more like a whole.

In the middle of the project, Rithikha and I focused on programming the front-end part. It certainly has not been a smooth process, with the largest challenge occurred when we were programming with flutter, our code was so complex due to the use of too many widgets that it was extremely difficult to debug and connect to the backend. We had to rewrite our code to follow the rule that one page is only included in one Dart file. We were worried that this would affect the progress of our whole project. Fortunately, with the help of the team members, we finally succeeded in making satisfactory results.

I am satisfied with our final outcomes, and really grateful to my teammates and our supervisor prof Ling Keck Voon. I think we make a great team. When we encounter difficulties, we will work together to solve them. When we encounter differences, we will listen to the opinions of others with an open mind. Everyone will play to their strengths and complete their work conscientiously and responsibly. I was glad I had the opportunity to be a part of this project.

12.6. Lim Zhi Wei

Before doing this smart mobile app project I had prior projects experience using ROS, C and Python programming alone. Mobile application development is a new endeavor for me, and I am willing to expand my programming knowledge during this 13-week course.

In the beginning, I volunteered to join in backend team to get a hand in understanding the platform of android studio to develop a flutter project from scratch. Many difficulties were encountered in studying the back-end coding and filtering of most online sources due to the different versions of the Flutter code and the dependencies that directly affect programming errors. I work closely with Isaac and Stanley who have prior experience doing mobile development.

Halfway through the project, I tried to help the frontend team to convert the Figma design to Flutter code for enabling the availability to access Firebase in google. Although Figma was a collaborative web application software for UX/UI design that did not require coding. However, due to its wide range of functions, this is not as simple as I thought and requires more extra time to study. I felt very appropriate for UX/UI design group members who put in the best effort to the beautiful icon, logo, pages, and wireframe design.

My role on the team is to also play the role of "generalist", I must take on multiple roles because it helps the overall project to go where it should go, and to be able to go through any problems they are facing and propose effective solutions while keeping track of each team's progress.

Although the whole project was not very smooth, I felt that it was a very interesting learning journey because of the excellent team performance. Every member of our team is very united because we will gladly accept the opinions of each member and quickly find the corresponding effective solution. Finally, I would like to thank our professor for his sincere guidance in this project and our team for their efforts and time in completing the project.

12.7. Jake Chang Jie

Before working on this project, I only had prior experience in solo coding projects in C, C++, and Python. Thus, I was not prepared for the intricacies of such a massive undertaking with so much work divided amongst eight people.

Initially, I volunteered for the role of a backend coder with several others as I was interested in learning more about backend implementation. Having no prior knowledge on the Flutter framework, I set about watching mobile app video tutorials and coding along with them to learn the syntax and gain a better understanding of Dart. I spent the first few weeks researching and practicing familiarizing myself. The hardest part was understanding the code as most of the video lecturers just briefly explained the codes and moved on which meant that I had to resort to google for many explanations. Regardless, my knowledge of Dart improved.

Midway into the semester, I was tasked with working on the full-stack development of the chat function and focused learning frontend development to create the UI/UX component of the chat first. However, my experience with Dart was lacking and I spent too much time on the front-end portion of the chat. I struggled to integrate the backend and only managed to finish the frontend portion of the chat. Nevertheless, I am still very proud of the chat function which I have produced, and I learnt the importance of time management and proper role allocation.

All in all, through these past 12 weeks I have gained invaluable learning experience of working in a team and am thankful for the help I have received from them for the implementation of the chat UI. I would also like to thank our Associate Professor Ling Keck Voon for checking on us and ensuring that we were on track halfway through the semester. Additionally, I realized that being a full-stack developer is very difficult and I aim to achieve the versatility of one before I embark on my next group project. I am glad I was given the opportunity to gain coding experience in a group context which has given me the confidence for my internship which is also software related.

12.8. Khairunnisa Ranam

This is my first experience in creating an application. I must say that it is a very enriching experience to work in a team. Even though I have had experience in creating websites before, I personally find that there are a lot of differences between website creation and app creation. Among them being, for web creation, because it is a solo project, I only had to craft the skeletal in my mind but for this project, because it is a group work, I find it hard to let my ideas be known and translate them down onto paper. Therefore, I truly appreciate my team member's help especially at the start of the project where, together, we brainstorm to complete the skeletal of the application (figure 10), also at the end where I just feel so stuck on how to proceed with the Figma wireframes. Their ideas and inputs have helped in which the skeletal of the app is not only fixed at the start of the project but is refined and further improved in the final product.

I was initially not confident in taking on the role for the design aspect of this application but input from team members has helped. Though I am proficient in photoshop and have had experience in using CAD software, I came to know of design collaboration platform called Spline 3D only in the middle of this project. This, I feel, is one of the major takeaways I get from this project as a designer.

With the rise in AR and VR technology, I carry the responsibility to create nice images in hopes of not letting the team down. I learned numerous things relating to the creation of an app. Such as how the app can stand out from current market products. This is the first time I have used Figma, so I truly appreciate the help I get from team members and YouTube tutorials. I have pre-knowledge in programming language in C, C++, and JavaScript, so it was relatively easy to a certain extend for me to understand the jargons used for the design portion of the app. An example would be transparency of images, text overflow and such. I understand why building the codes from scratch would be a better option than to use a machine-generated code. This experience has taught me ways in which I could further refine my skills in app designing. As a person who is interested in designing and developing apps, I was glad I was given the opportunity to be part of this project. I came to really enjoy doing the skeleton of the application in Figma.

I also learned the importance of teamwork, time management and communication between team members. There are more areas that I should improve on for my own personal growth.

In a nutshell, I would like to thank prof Ling Keck Voon and my hardworking teammates for giving me an opportunity to grow, as well as supporting me and giving me feedback whenever I am stuck or unsure of how to proceed with this project. The ride was not a smooth one, but I am glad that I managed to gain valuable knowledge and experience in a short period of time.

13. References

- [1] 70+ Flutter Projects For Beginners And Final Year Students. (2022). Retrieved from The Insane App: <https://www.theinsaneapp.com/2021/06/Flutter-projects-with-source-code.html>
- [2] dbestech. (2022, January 28). *Flutter E-commerce App With Backend | Food Delivery App Development for iOS and Android*. Retrieved from YouTube: <https://www.youtube.com/watch?v=7dAt-JMSCVQ>
- [3] Demedyuk, I., & Nazar Tsybulskyi. (2020, March 11). *Flutter vs Native vs React-Native: Examining performance*. Retrieved from Medium: <https://medium.com/swlh/flutter-vs-native-vs-react-native-examining-performance-31338f081980>
- [4] Developers, S. (2021, August 20). *Build a Flutter Chat App*. Retrieved from YouTube: <https://www.youtube.com/watch?v=vqqBc7jni8c>
- [5] Flutter. (2022). *Widgets Library*. Retrieved from Api Flutter Dev: <https://api.flutter.dev/flutter/widgets/widgets-library.html>
- [6] HowtoFlutter. (2021, September 10). *How to Add Opacity / Color Filter to an Image in Flutter*. Retrieved from How to Flutter: <https://howtoflutter.dev/how-to-add-opacity-color-filter-to-an-image-in-flutter/>
- [7] Kodeversitas. (2018, March 6). *Clean Login UI with Hero Animation in Flutter*. Retrieved from YouTube: <https://www.youtube.com/watch?v=efbB8-x9T2c>
- [8] Manning, N. (2022). *Tutorials*. Retrieved from Flutter Crash Course: <https://fluttercrashcourse.com/blog>
- [9] Martin, S. (2020, September 16). *Building Chat App in Flutter with Firebase (Latest Updated 2022)*. Retrieved from Medium: <https://medium.com/flutter-community/building-chat-app-in-flutter-with-firebase-888b6222fe20>
- [10] Most Popular Technologies Misc Tech. (2022). Retrieved from Stack Over Flow: <https://survey.stackoverflow.co/2022/#most-popular-technologies-misc-tech>
- [11] Sarah Goff-Dupont, & Matt Shelton. (2022). *Install Git*. Retrieved from Atlassian Bitbucket: <https://www.atlassian.com/git/tutorials/install-git#install-the-git-credential-osx>
- [12] Tran, Q. D. (2018, August 24). *Building chat app with Flutter and Firebase*. Retrieved from Medium: <https://medium.com/flutter-community/building-a-chat-app-with-flutter-and-firebase-from-scratch-9eaa7f41782e>
- [13] Utter. (2021, May 11). *Figma Design to Flutter Code*. Retrieved from YouTube: <https://www.youtube.com/watch?v=sApYJKJAcZo>
- [14] Way, T. F. (2020, July 3). *Online Shop App - Flutter UI - Speed Code*. Retrieved from YouTube: <https://www.youtube.com/watch?v=XBKzpTz65lo>
- [15] Yogan, R. (2019, August 2). *FlutterUI - Minimal Designs - Nutrition app*. Retrieved from YouTube: https://www.youtube.com/watch?v=K1uH_SN4X0w

14. List of Figure

Figure 1	Our First Choice in IDE: Android Studio	6
Figure 2	Our Alternative IDE: VS Code	6
Figure 3	NTU Mascot.	7
Figure 4	Colour scheme for the application.	7
Figure 5	Previous design of Nanyang Marché.	8
Figure 6	Nanyang Marché logo and icon.	8
Figure 7	Icon example from Figma platform.	8
Figure 8	Lilit Store's wallpaper	9
Figure 9	Nanyang Marché's background wallpaper	9
Figure 10	Initial rough sketches of the application.	10
Figure 11	Transferred sketches into Figma.	11
Figure 12	Figma's user flow.	11
Figure 13	Application's expected outcome	12
Figure 14	Onboarding Page (Actual Outcome)	13
Figure 15	Login Page (Actual Outcome)	14
Figure 16	Sign up Page (Actual Outcome)	15
Figure 17	Home Page (Actual Outcome)	16
Figure 18	Upload Page (Actual Outcome)	17
Figure 19	Product Details Page (Actual Outcome)	18
Figure 20	Chat Messages Page (Actual Outcome)	19
Figure 21	Chat Screen (Actual Outcome)	19
Figure 22	Profile Page (Actual Outcome)	20
Figure 23	"curved_navigation_bar" in Flutter	21
Figure 24	Performance of Frameworks on iOS	23
Figure 25	Performance of Frameworks on Android	23
Figure 26	React Native Architecture and Bridging	24
Figure 27	Popularity of Frameworks in 2022	25
Figure 28	Most Loved and Dreaded Frameworks in 2022	26
Figure 29	Flutter Architecture	27
Figure 30	FlutterFire CLI	29
Figure 31	Project folder Structure in Android Studio	30
Figure 32	Import dependencies in Android Studio	30
Figure 33	Users collection in Cloud Firestore	31
Figure 34	Products collection in Cloud Firestore	31
Figure 35	Functions to access Firebase using Flutter	31
Figure 36	Images uploaded by user to cloud servers	32
Figure 37	Accessing Firebase Storage using Flutter	32
Figure 38	Authentication using Flutter	33
Figure 39	Upload function using Flutter	34
Figure 40	Actual Chat Messaging Cloud Firestore Database	35
Figure 41	Expected Chat Messaging Cloud Firestore Database	35
Figure 42	Stream Chat Messaging API	36

Appendix A - Project Members Information

	Name	Project contributions	Report Contribution
1	Isaac Tay Peng Bin	<ul style="list-style-type: none"> Group Leader Come up with the name of the app Built various features for the backend like uploading the pictures and text of each product Setup the Firebase Schema and other functions like Authentication and Storage Integrated frontend and backend by allowing the frontend to fetch data from the backend 	Chapters 1, 2, 3 and 10
2	Chai Jia Zhe	<ul style="list-style-type: none"> Treasurer Proofreading of final report 	Chapters 1, 2, 3
3	Stanley Wong Qi Ren	<ul style="list-style-type: none"> Built the front-end pages like the onboarding page and profile page Assisted the backend by giving the app access to the camera and gallery of the phone and setting up the Cloud Firestore Schema Mentoring teammates on Git, Android Studio and Flutter Debugging and reviewing the codebase Introduced the tech stack and IDE 	Chapters 5, 6, 9 and 10
4	Rithikha Vivekanantham	<ul style="list-style-type: none"> Oversaw progress and worked on adding improvements in Figma by making it more aesthetically pleasing and interactive Designed the initial hand drawn draft of the app Front end coder for the onboarding page and details page 	Chapters 6, 7, 8, 10 and 11
5	Ke Tangxin	<ul style="list-style-type: none"> Set the app's tone in Figma, like colours, widgets Be a front-end coder of login, sign up, upload and home pages etc. Find some pictures and icons for the app Integrate frontend and backend for login and sign-up pages 	Chapters 7, 8 and 11
6	Lim Zhi Wei	<ul style="list-style-type: none"> Idea of bringing NTU Marketplace to an actual app Sliding and responsive interface design in Figma 	Chapters 4, 14 and overall figure and report formatting

		<ul style="list-style-type: none"> • Final report and presentation slide format editor • Chat Functionality • Debugging 	
7	Jake Chang Jie	<ul style="list-style-type: none"> • Chat Functionality and UI • Backend Research • Frontend and Backend Integration • Quality Assurance and Control • Debugging 	Chapters 8, 10 and 11
8	Khairunnisa Ranam	<ul style="list-style-type: none"> • Draw hand drawn draft of app • Design app's logo, icon & wallpaper • Figma wireframes & prototype • Find images and icons for Figma 	Chapters 7, 8 and 11