MAPG REPORT

Name: Jamie Gonzales Esguerra

Student ID:

Class: PA33

Contents

Introduction

For this project, I used android studio with the help of flutter to create a simple mobile application similar to the Ngee Ann website. Using flutter, I used different widgets that enabled me to go to a new screen, show images or even add animations to make my app more appealing for the user. I took the images from the Ngee Ann website and put them together in the list view widget as its children so that the images could be scrolled vertically or horizontally. I have used the 6 basic widgets that the project requires. The 6 different basic widgets are “Text”, “Button”, “CheckBox”, “Image”, “SnarcBar” and “ListView”. Furthermore, I added 4 advanced features. This includes “Icon”, “Animation” , “Card Widget” and “Drop Down Button Widget”.

Mobile Application Pages

Main Page

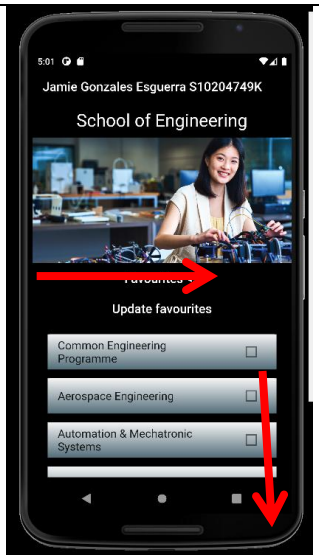


Figure 1

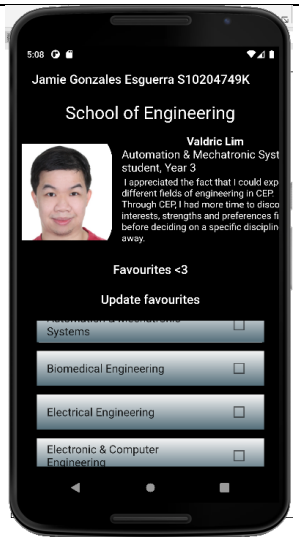


Figure 2

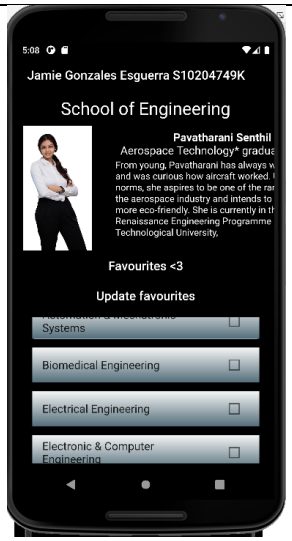


Figure 3

My app allows the user to scroll horizontally on top and vertically below. I have added the horizontal scroll on top as I want to show the different alumni of Ngee Ann Polytechnic. It would first show a picture of themselves and on the right side of their picture would be the description about them.

Favourites page

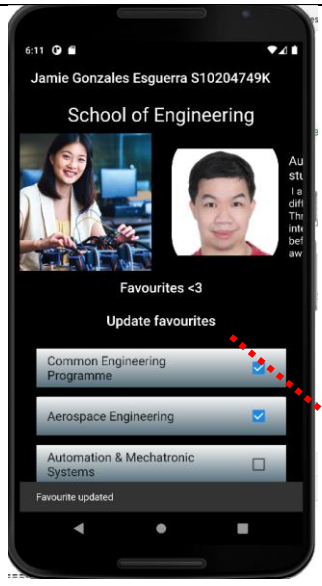


Figure 4: Main Page

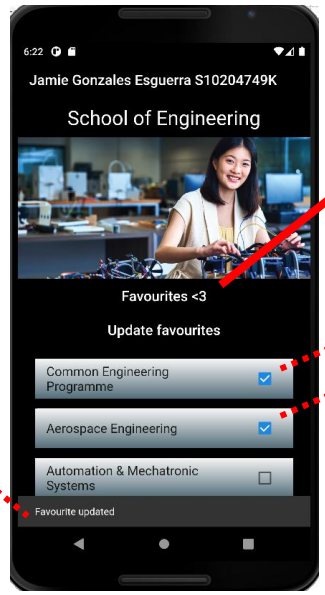


Figure 5: Main page

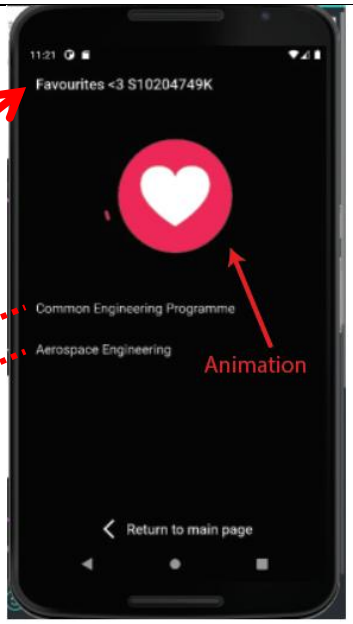


Figure 6: Favourite page

My app would consist two buttons after the horizontal scroll view. The two buttons are “Favourite <3” and “Update favourite”. As you can see from the figures, each course has a checkbox. When this checkbox is checked and the “Update favourite” is tapped, those respective courses would be put inside the “Favourite <3” page. Furthermore, a snackar would appear below stating that the “Favourite <3” page has been updated. Pressing the “Favourite <3” button would lead the user to the favourite page in which they would be able to see the courses that they have checked.

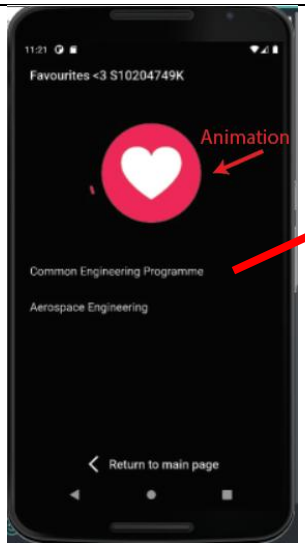


Figure 7: Favourite page

Figure 8:Main page

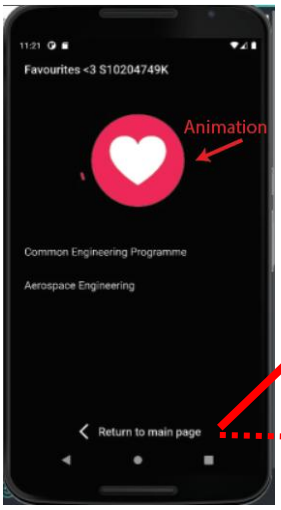


Figure 9: Favourite page

Figure 10:Main page

This favourite page is useful as the user would not have to scroll down to all the different courses to find the course they want to check. They can just go to the “Favourite <3” page and click on the course they want to go to (Figure 7 and 8). Furthermore, clicking the button “Return to main page” would return the user back to the main page with a “snackBar” below showing “Returned back to School of Engineering from Favourites”.

Engineering programme pages

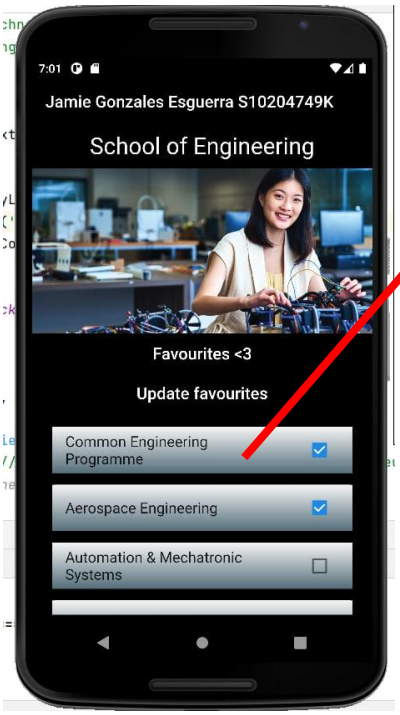


Figure 11: Main page

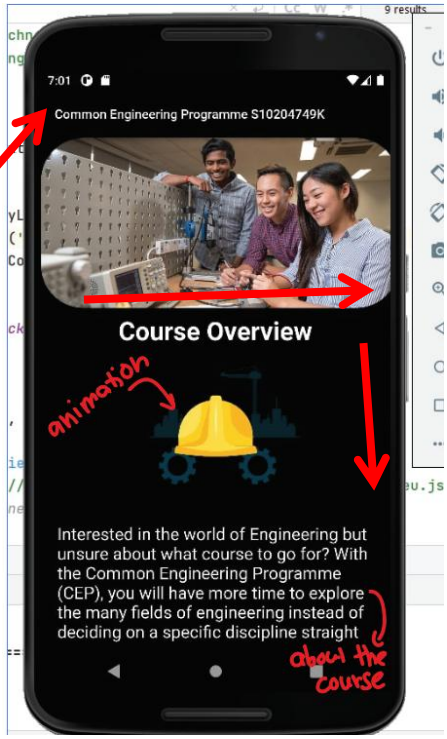


Figure 12: CEP page

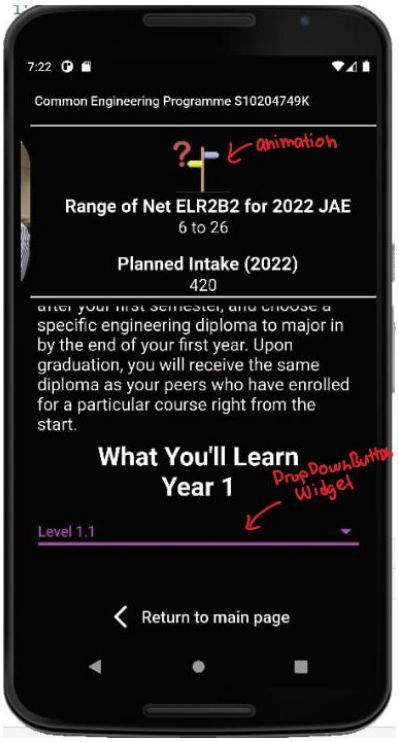


Figure 13: CEP page

The course pages format would be similar to the main page. It has a horizontal scroll on top and vertical scroll below. Scrolling horizontally would show the course's entry requirements and an animation on top. The course overview can be fully seen if you scroll down. Scrolling down to the very end would lead the user to a "Drop Down Button" widget.

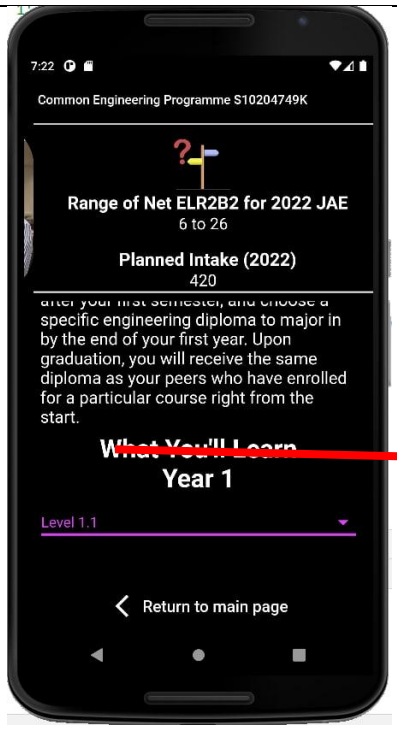


Figure 14: CEP page

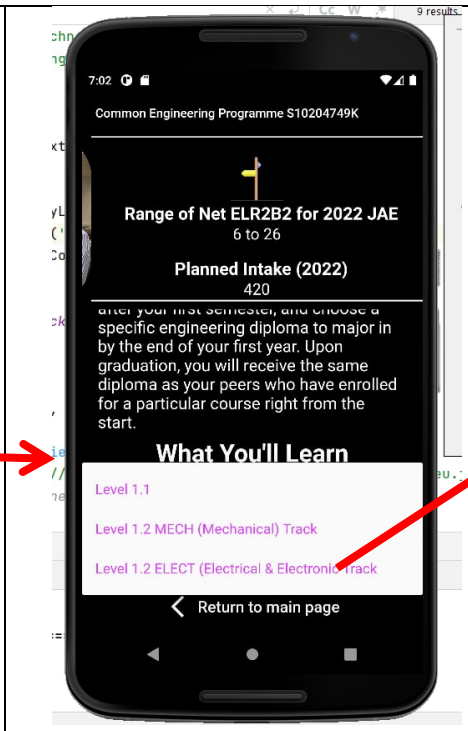


Figure 15: CEP page

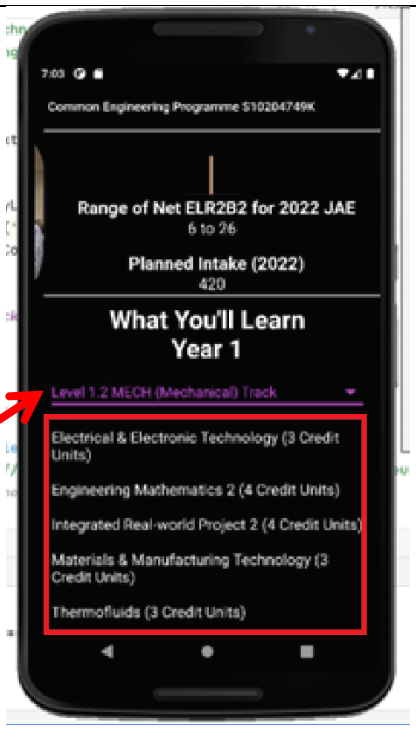


Figure 16: CEP page

The "Drop Down Button" widget would show the different semesters. Clicking a specific level as seen from figure 15 and 16, would show the modules for that particular semester. Figure 16 shows the modules of "Level 1.2 MECH (Mechanical Track). Lastly, clicking the "Return to main page" would return the user to the main page with a "snackBar" below showing "Returned back from CEP".

This format is used for the other courses on this app.

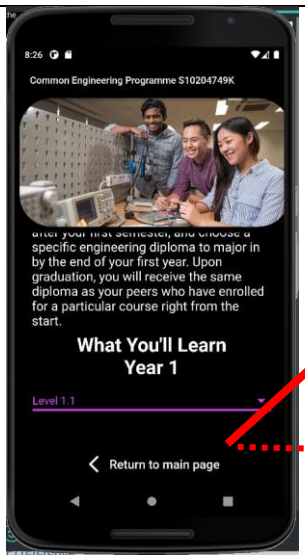


Figure 17: Main page

Figure 18: Main page

Enhancements

Horizontal scroll

```
child: Expanded(
  child: ListView(
    scrollDirection: Axis.horizontal,
    children: [
      ClipRRect(
        //borderRadius: BorderRadius.all(Radius.circular(20.0)),
        child: Image.asset('images/SEO-hero.png'),
      ), // ClipRRect
      //-------------------------------------------------------
      //VALDRIC
      Container(
        padding: const EdgeInsets.fromLTRB(30, 0, 15, 0),
        height: 100,
```

Figure 19: Horizontal code

The "scrollDirection: Axis.horizontal" allows the children of my listview to be scrolled horizontal instead of vertical. Hence, it allows me to have two scroll ways on my app. One on top and another one below that scrolls horizontal and vertical respectively.

Icon

```
icon: const Icon(Icons.arrow_drop_down,
    color: Colors.purpleAccent),  // Icon
iconSize: 40,
//elevation: 16,
style: const TextStyle(color: Colors.purpleAccent),
isExpanded:
    true,
//so that the icon is moved to the right and fills entire dropdownbutton
```

Figure 20: Icon code

The icon allowed me to have an arrow for my DropDownButton. I can change the size of this icon, colour and where this icon would be placed as seen from the code on the right. "isExpanded : true" made my icon move all the way to the right hand side of the app screen.

Animation

```
dependencies:
  flutter:
    sdk: flutter


  # The following adds the
  # Use with the Cupertino
  cupertino_icons: ^1.0.2
  lottie: ^1.3.0
```

Figure 21: pubspec.yaml file

Figure 22: Animation

```
import 'package:lottie/lottie.dart';
Container(
  height: 80,
  width: 80,
  child: LottieBuilder.network(
      'https://assets2.lottiefiles.com/private_files/lf30_dleb6zk6.json'),
), // Container
```

Referring to this link "https://pub.dev/packages/lottie/install", I added the new version of lottie inside my pubspec.yaml file. "lottie: ^1.0.2"

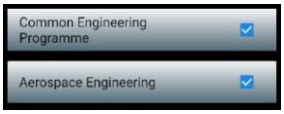
To use the animation, I imported the package on the dart file in which I am going to use the Lottie. After importing the Lottie, I copied the URL of the animation I want to import from this link "https://lottiefiles.com/". This URL link would be added inside my LottieBuilder.network to show on my app.

Card widget

```
child: Card(
  //elevation: 8, //to add a shadow below
  margin: EdgeInsets.all(
      6), //space in between each card  //
  child: Container(
    decoration: BoxDecoration(
      gradient: LinearGradient(
        colors: [
          Colors.grey.shade100,
          Colors.blueGrey.shade600,
        ],
        begin: Alignment.topCenter,
        end: Alignment.bottomCenter,
      ),  // LinearGradient
    ),  // BoxDecoration
    child: ListTile(
```

Figure 23: Card widget code

I have added a card widget as it allowed me to decorate all the boxes of my engineering courses text.



Each course has a gradient colour from a lighter shade to darker. This is possible due to the decoration of my card widget. It starts with the colour “Colors.grey.shade100” on top and ends with “Color.blueGrey.shade600”.

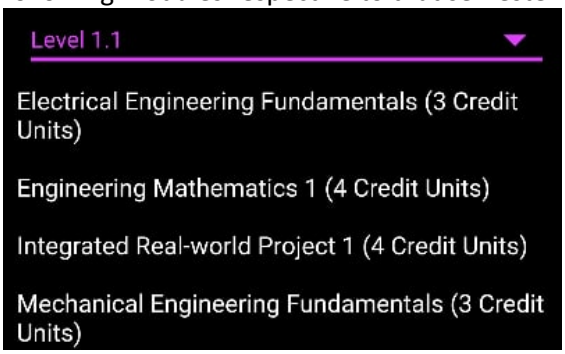
Drop Down Button widget

```
child: DropdownButton<String>(
  value: dropdownValueYearTwo,
  icon: const Icon(Icons.arrow_drop_down,
      color: Colors.purpleAccent),  // Icon
  iconSize: 40,
  //elevation: 16,
  style: const TextStyle(color: Colors.purpleAccent),
  isExpanded:
      true, //so that the icon is moved to the right and fi
  underline: Container(
    //box
    height: 3,
    color: Colors.purpleAccent,
  ),  // Container
  onChanged: (String? newValue) {
    setState(() {
      dropdownValueYearTwo = newValue!;
      if (dropdownValueYearTwo == 'Level 2.1') {
        DisplayTextYearTwo = LevelTwo[0];
      } else if (dropdownValueYearTwo == 'Level 2.2') {
        DisplayTextYearTwo = LevelTwo[1];
        DisplayTextYearTwo = LevelTwo[1];
      } else {
        DisplayTextYearTwo = '';
      }
    });
  },
  items: <String>[
    //what is seen on the drop down menu button
    'Level 2.1',
    'Level 2.2',
  ].map<DropdownMenuItem<String>>((String value) {
    return DropdownMenuItem<String>(
      value: value,
      child: Container(
        //so that it would not overflow the text
        child: Text(
          //to show the text in the drop down menu
          value,
          style: TextStyle(
            fontSize: 17,
          ),  // TextStyle
        ),  // Text
```

Figure 24: Drop down button code

I used the “Drop Down Button” widget to allow the user to choose the respective semester’s module they want to check. The level is mapped with the DropDownMenu by using the code “map<DropdownMenuItem<String>>”. Therefore, when I clicked on the “Drop Down Button” widget, it would display the list inside the “items:”. Since my list “items” contains “Level 1.1” and “Level 1.2” it would show:

Once, the level has been chosen, it displays the following modules respective to that semester.



Conclusion

Through this project, I became more confident in programming flutter and learnt its basics. After imitating the Ngee Ann Website, I acquired the knowledge of using the button widgets of flutter to move to the next screen. However, I have faced difficulties in passing data from the “Main” screen to the “Favourites <3” screen. Android studio did not allow me to use the data I have passed. After seeking for solution online, I learnt that I should put a “widget.” to the passed data on the second screen for it to be used.

```
Expanded(
  child: ListView.builder(
    itemCount: widget
        .list.length, //to access the list, i mu
    itemBuilder: (BuildContext context, int inde
      return ListTile(
        onTap: () async {
          //added an async as i am passing data
          if (widget.list[index] ==
```

Moreover, adding the “Favourite’s <3” page helped me learn the skills of replacing, adding and removing items inside the list. After experimenting on removing and adding items on the list, I was able to show the expected courses I wanted on the “Favourite’s <3” page.

```
setState(() {
  if (_listStatus.elementAt(0) == true) {
    _FavouritesList.remove(
        'Common Engineering Programme');
    _FavouritesList.add(
        'Common Engineering Programme');
  } else {
    _FavouritesList.remove(
        'Common Engineering Programme');
  }
```

To summarise, I have gained adaptability and resilience by doing this project as I encountered problems along the way. I have come to realise that searching online to debug codes helps a lot as other people may have had the same issues. Not only it can help me debug programs but also learn new features that I can add on to make my app more exceptional. With perseverance, I was able to imitate the Ngee Ann Website into an app using the Dart language.