

UNIVERSITY OF LONDON

CM2020: AGILE SOFTWARE PROJECT SUBMISSION [Part 2]

Team Name: Team 36

Team Members:
Goh Sin Hau
Kwang Kian Hui
Koh Bao Wen
Koh Jia Xiang
Lawrence Ho

Date Submitted: 20 September 2021

Application Name: DEALL

1.0 Background	6
Introduction	6
Problem Statement	6
Aims	6
Objectives	7
Meeting schedule	7
Proposal summary (for Part 1)	7
Scope	7
Deliverables - Development phase	7
Constraints/limitations	8
2.0 Planning & Research	9
Gantt (Overall milestones)	9
Gantt (Sprint Details)	13
Sprint 1	13
Sprint 2	14
Sprint 3	14
Sprint 4	15
Sprint 5	15
Task allocation for report	16
Role in software development	16
Task allocation for software development	16
Resources	18
Stakeholders	18
Methodologies and techniques used	20
3.0 Prototyping & Iteration	20
Strategies and techniques	21
Prototype Dimensions	21
Prototype Sources	21
Lo-Fi Prototype	22
V1.0	22
V2.0	24
V3.0	26
Findings and Evaluation	28
Hi-Fi Prototype	28
V1.0	28
V2.0	31
Final design	33

4.0 Final Design	44
Specifications	44
UI/UX Patterns	44
Progressive disclosure	44
Clear primary actions	45
Design Justification	46
Features Justification	48
Features for consumers	48
Features for retailers	50
5.0 Requirements	51
Functional requirements	51
6.0 Use cases	53
7.0 System Development	65
Technologies	65
Language and framework	65
Minimum device requirement	65
Authentication	65
Database	65
Firebase cloud firestore	65
Firebase cloud storage	66
Database Structure	67
Software Architecture	67
Presentation layer	69
Application layer	69
Infrastructure layer	69
Example of data flow	70
Summary	72
File structure	73
Error and exception handling	74
Official documentation	75
External resources	76
Black box testing	77
Browse through retailers (consumer)	77
Search for retailers (consumer)	78
View retailer details (consumer)	79
View available surplus food (consumer)	80
Mark retailer as favourite (consumer)	81
View favourite retailers (consumer)	82
Login (consumer & retailer)	83
Sign up (consumer)	84

Sign up (retailer)	85
Create food (retailer)	86
View foods (retailer)	88
Edit food (retailer)	89
Delete food (retailer)	91
Toggle shop visibility (retailer)	92
Select available food (retailer)	93
White box testing	94
Sprints	99
Sprint log sorted by sprints	100
Sprint log sorted by members	105
Version control tool	111
8.0 User Testing	113
User interface evaluation (online):	113
Usability testing:	114
System Usability Scale(SUS):	114
Additional usability testing:	115
First Sprint:	115
Second Sprint:	116
Third Sprint:	117
Fourth Sprint:	118
Summary:	120
Findings	122
9.0 Evaluation	124
Teamwork Processes	124
Challenges Faced	125
Product feature variations from original proposal	125
Application Evaluation(User)	125
Potential Application improvements	127
Application services:	127
Application UI/UX:	128
SWOT ANALYSIS	128
STEEPLE Analysis	129
Market comparison	130
Key takeaway	131
Our position in the market	131
10.0 User Guide	132
Sign in page:	132
Consumer Sign Up Page:	133
Consumer Home Page (Retailer List):	133

Consumer Product List Page(Product list of selected retailer):	134
Consumer - Retailer Detail Page (Selected retailer's detail):	134
Consumer Favourites Page:	135
Retailer Sign Up Page:	135
Retailer Home Page:	136
Retailer Profile Page:	137
Retailer Add Product/Edit Product Page:	137
Regardless, we have provided a demo video stored under Appendix - section A2.	137
11.0 Conclusion	138
Life cycle beyond delivery stage	138
Application of software	138
Impact of software	139
Appendix	140
A1 - Test accounts	140
Consumers	140
Retailers	140
A2 - Demonstration video link	141
Consumer perspective	141
Retailer perspective	141
A3 - Github link	141
A4 - Other Technical details:	141
A5 - Full git commits	141
References	228
Reflection	230

1.0 Background

Introduction

Given the task of developing a software with a team of student developers, we sought a subject where we could explore the design and development integration with current affairs in the world. During the early planning phases, many different ideations were raised, but we settled for a solution tackling food waste.

Problem Statement

In the annual statistics provided by Singapore's National Environment Agency (3R Programmes and Regulations, n.d), food waste accounts for about 11% of the total waste generated in 2020. Over the past decade, food waste production has seen a clear upward trend, with an estimated growth of 20% since 2010. Though small as we may be, this is yet a contributing factor to an estimated 1.3 billion tons of food wasted globally every year, which is the rough equivalent to one third of all food produced (Worldwide Food Waste, n.d).

This is a deeply concerning issue, food waste contributes significantly to Singapore's only landfill which is expected to be filled by 2035 (Food Waste, n.d). Food waste stems from many different sources, but food retailers are the main focus of the project. It has been estimated that at a hotel banquet, 33% of the food will be wasted. The customary convention of keeping buffets fully stocked during business hours creates even more waste. Kitchen culture for over preparation of food to ensure smooth operations such as sushi restaurants etc may face this dilemma as well.

Aims

It is in our common interests to develop an intuitive mobile application that allows food & beverages outlets to sell their surplus/excess food through a listing platform. Business owners can upload food listings for consumers to browse and purchase at a discounted rate.

Through this application, we aim to reduce amounts of carbon footprints from food waste, and provide a platform for consumers to access a more economically friendly approach to purchasing food. It is also in our due interests to advocate for a greener environment, and promote eco-friendly motives amongst the public.

As we move along into the development phase of the project, we intend to continue pursuing the same aims and objectives that we have established.

Objectives

1. To reduce food waste and promote environmentally friendly motives.
2. Drive foot traffic to food & beverages outlets.
3. Maximize business revenue for food & beverages outlets.
4. Provide a platform of discounted food sources for consumers.

Meeting schedule

- Meetup at the start and end of each sprint
- During a sprint, we will meet up on every alternate day to discuss what was done and what will be done in the next two days for each member

Proposal summary (for Part 1)

In the previous project phase, we have discussed the issue of food waste, its severity and impact. We have laid out our aims and objectives of how we hope to reduce carbon footprint from food waste through a mobile application. This mobile application acts as a platform for both food & beverages retailers to promote their surplus/excess food, and for consumers to look for discounted food. With both parties incentivised to use this application, we believe that it can make an impact in reducing carbon produced from food waste.

We have also discussed the scope of work and deliverables through the different phases of this project. Market research, analysis, and survey, along with use case study was carried out during the first phase. Defining functional and non-functional requirements, methodologies, and development of the Lo-fi prototype was then performed in the second phase. Development of Hi-fi prototype, user feedback gathering, and evaluation of project state was then performed in the final phase.

We have also defined the available resources, risks & challenges, and limitations & constraints of this project to further define what can or cannot be done. We also presented how the tasks were allocated to each member of the group. We then evaluated our Hi-fi prototype and concluded that some changes to the user interface will be needed which will be achieved in the development phase of the project.

Scope

Deliverables - Development phase

1. Fully functioning application
 - a. Functional and Non-functional requirements

- b. Improvement to Hi-Fi prototypes detailing overall UI/UX for the application
2. Fully functioning database
3. Gantt chart
4. Sprint boards (Trello)
5. Survey: User feedback
6. Black box testing
7. White box testing
8. User guide
9. Demonstration video
10. Github commit logs
11. Final deliverables
 - a. Project Report
 - b. Application (.zip)
 - c. Demo video
 - d. readme.txt (for instructions on running the app)
 - e. log.txt (github commit logs)
 - f. apk file

Constraints/limitations

The application will not have the following features due to the challenges raised and the limited time of the project:

- Location and distance related features.
- No payment platforms available through the application. Payment has to be done in store rather than through the application.
- Forget password feature that allows users to reset their password.

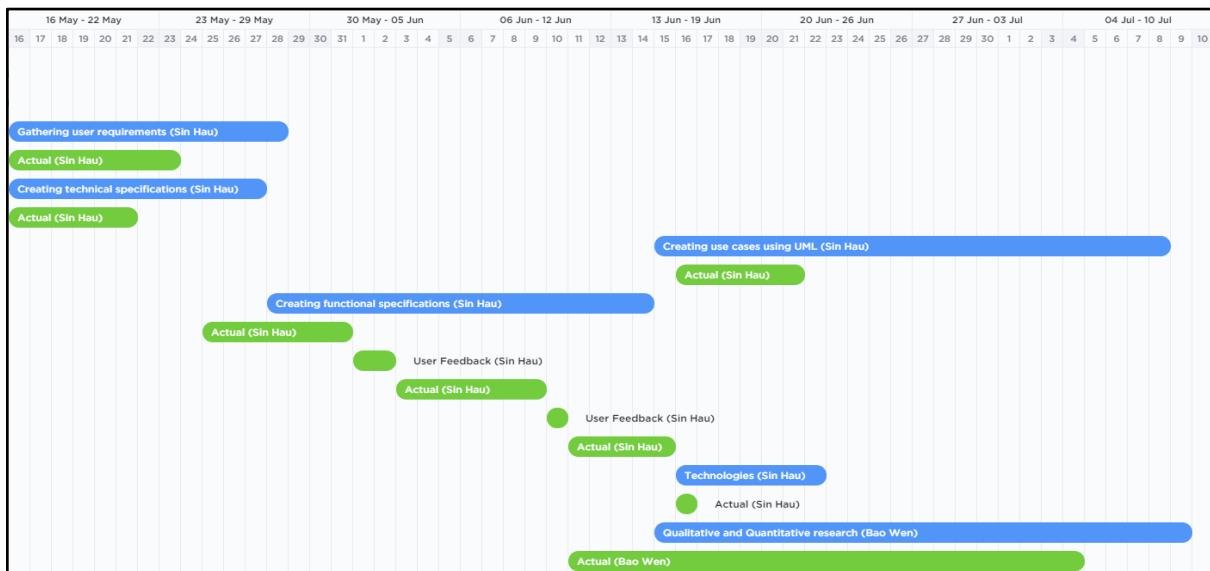
2.0 Planning & Research

Gantt (Overall milestones)

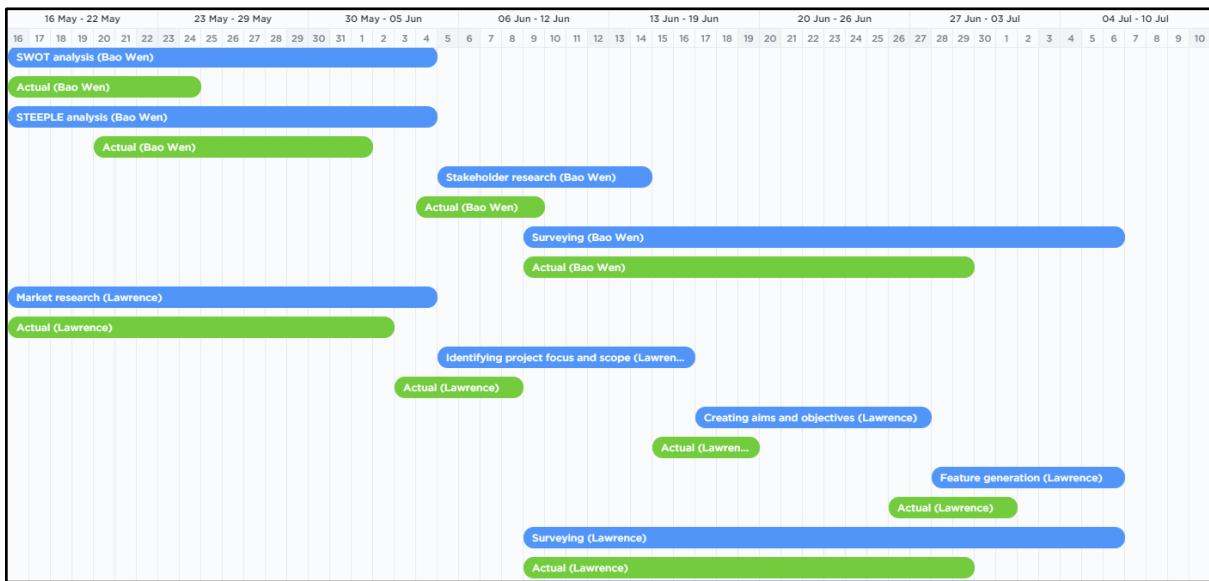
The gantt charts in this section indicates our overall timeline of events starting with ideation stage at the beginning and ending with the completions of sprints. The use of gantt charts was to visualise our progression flow and keep tabs on tasks. Updates of the charts were reviewed weekly with upcoming items added. Planned tasks are coloured in blue and the actual duration taken will be coloured in green.



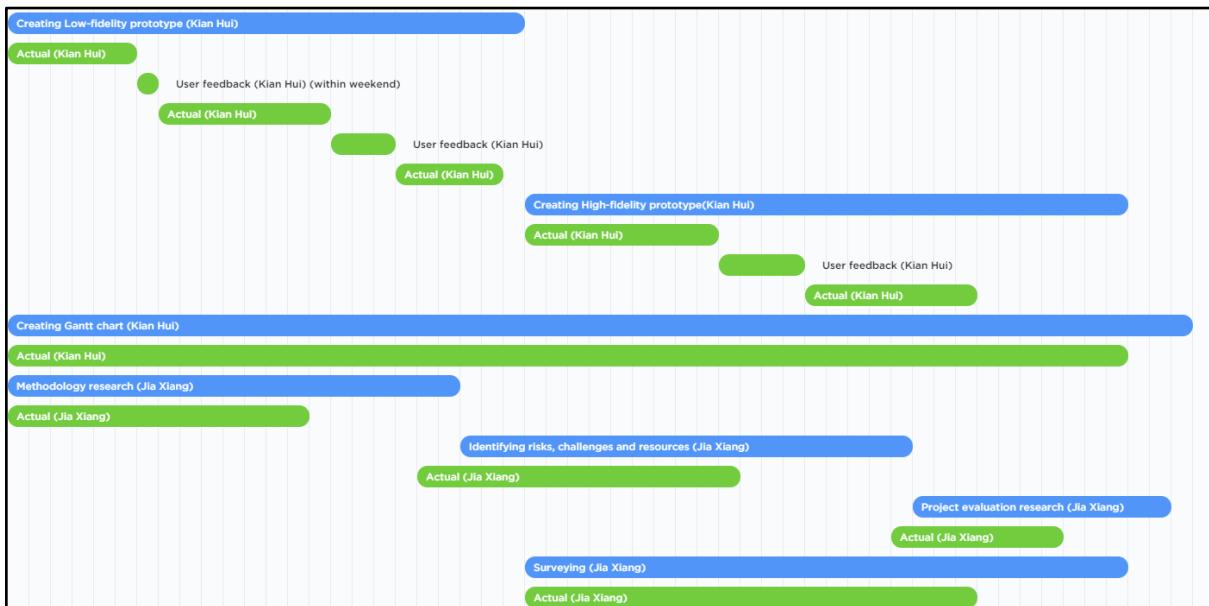
[28 April 2021 - 15 May 2021] - The start of idea generation for the project.



[16 May - 9 July 2021] - Requirements gathering, qualitative and quantitative research and creating functional specifications and use cases modeling.



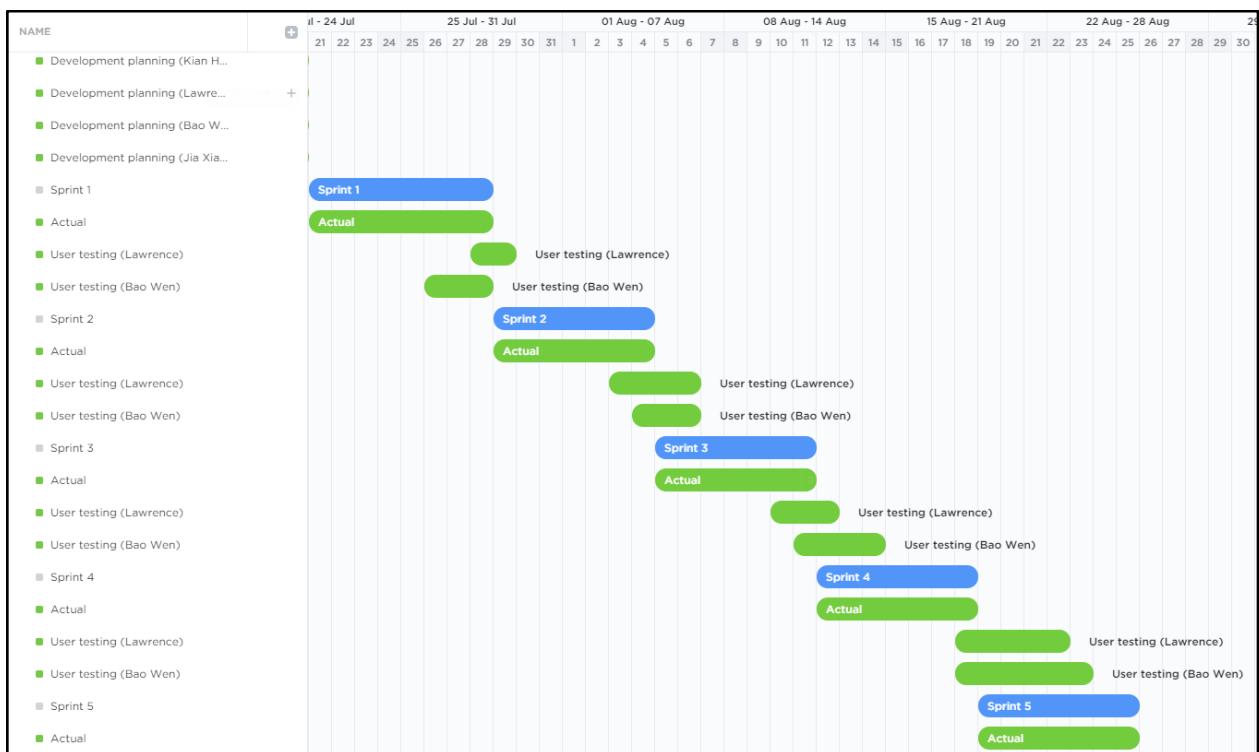
[16 May - 9 July 2021] - SWOT analysis, market and stakeholder research with surveys. Identifying focus and scope of projects with aims and objectives.



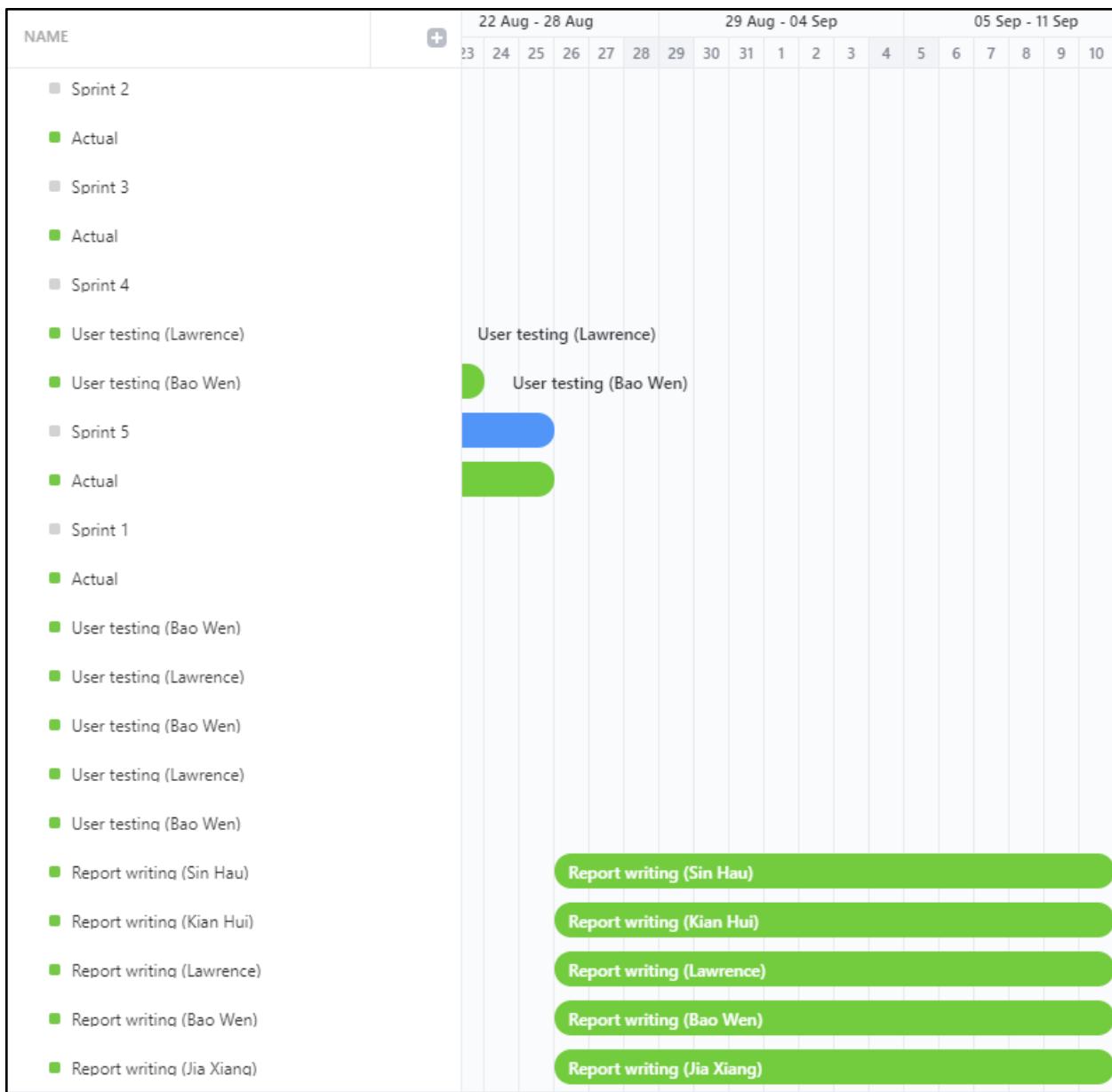
[16 May - 9 July 2021] - Low and High-fidelity iterative prototyping with user feedback, methodologies research and project evaluation. Identification of risks, challenges and resources.

NAME	+	11 Jul - 17 Jul							18 Jul													
		10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
■ Development planning (Sin Ha...)																						
■ Development planning (Kian H...)																						
■ Development planning (Lawre...)																						
■ Development planning (Bao W...)																						
■ Development planning (Jia Xia...)																						

[9 July - 20 July 2021] - Planning for software development and upcoming sprints.



[21 July - 25 August 2021] - Software development in sprints with continuous user testing.

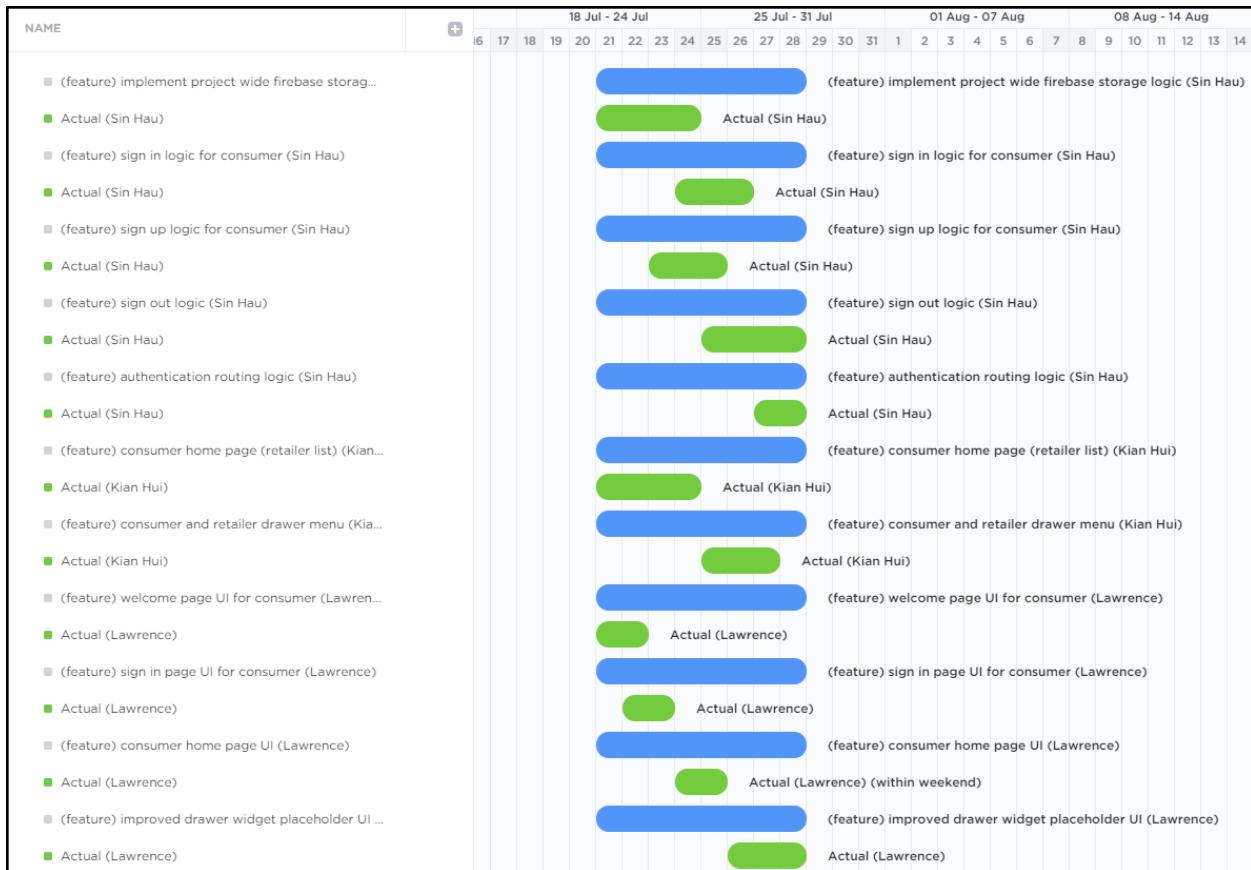


[21 July - 10 September 2021]- Final report writing

Gantt (Sprint Details)

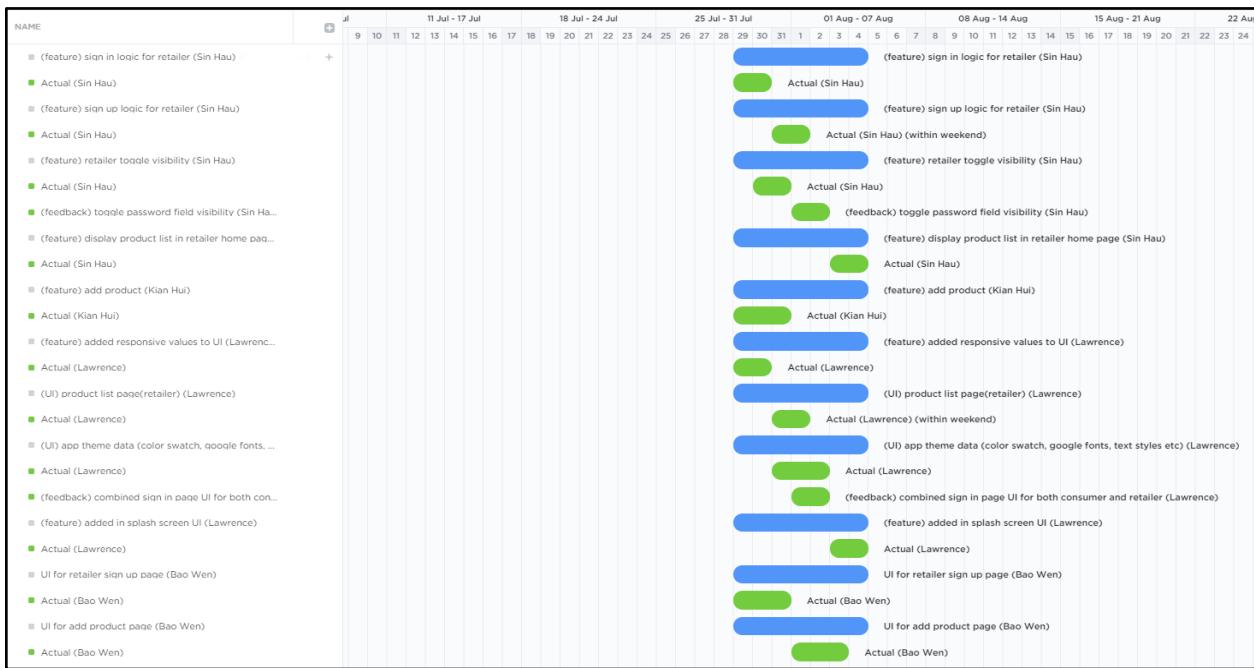
In this section are the detailed gantt charts for sprints. As Scrum was used to manage development workflow, sprints were employed to increase productivity. Compared to the milestones gantt chart in the previous section, the gantt charts here detail activities for each sprint. Before each sprint, selected tasks were added into the charts. The blue bars denote planned tasks and green bars represent the actual time taken. Reviews were conducted during alternate days and concluded again during the end of the sprint.

Sprint 1



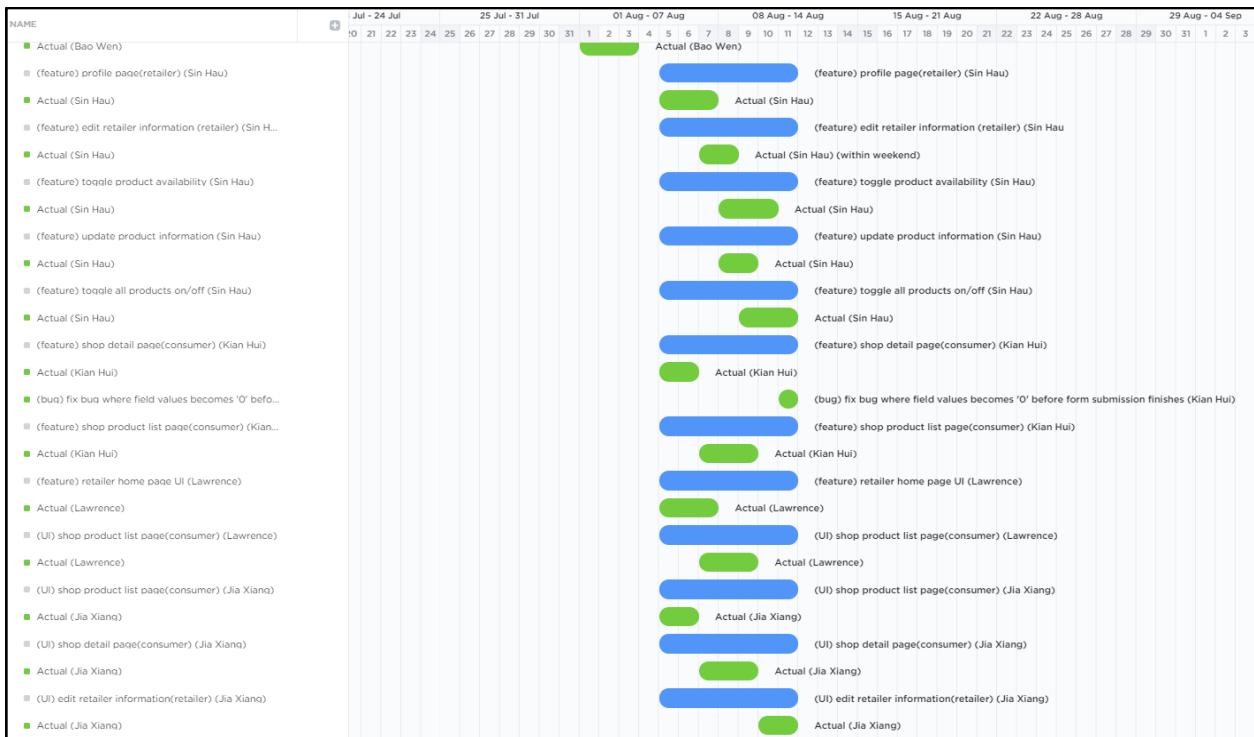
[21 July - 28 July 2021]

Sprint 2



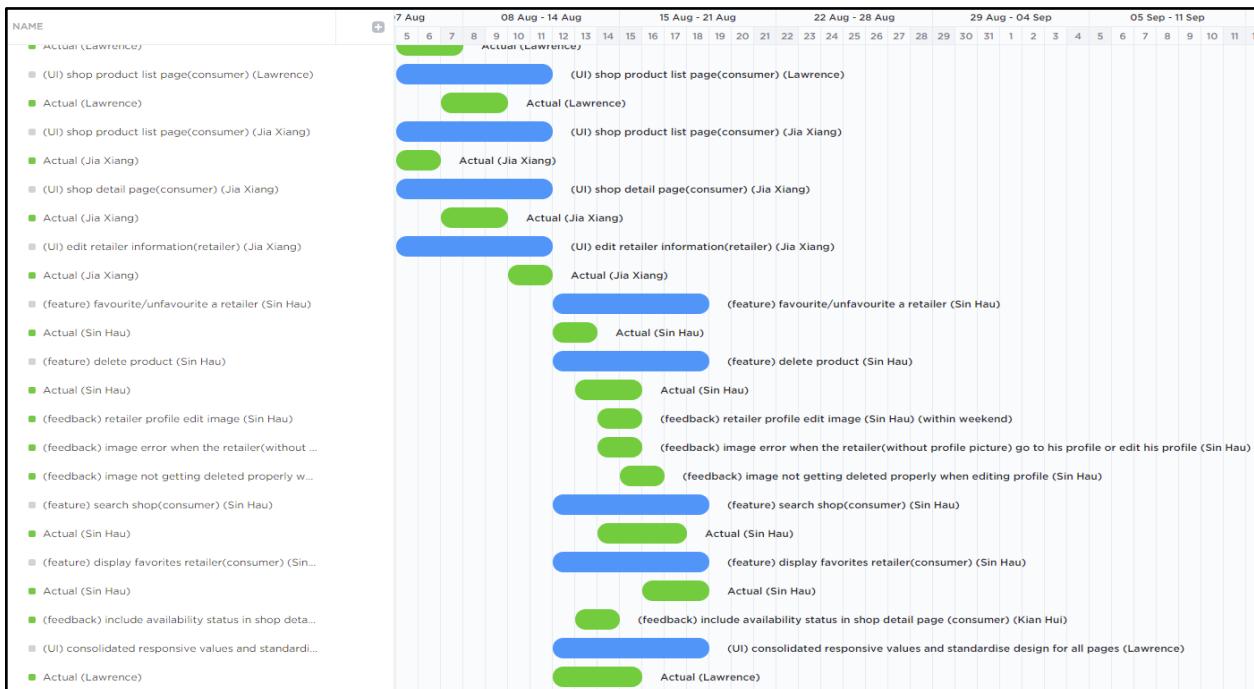
[29 July - 4 August 2021]

Sprint 3



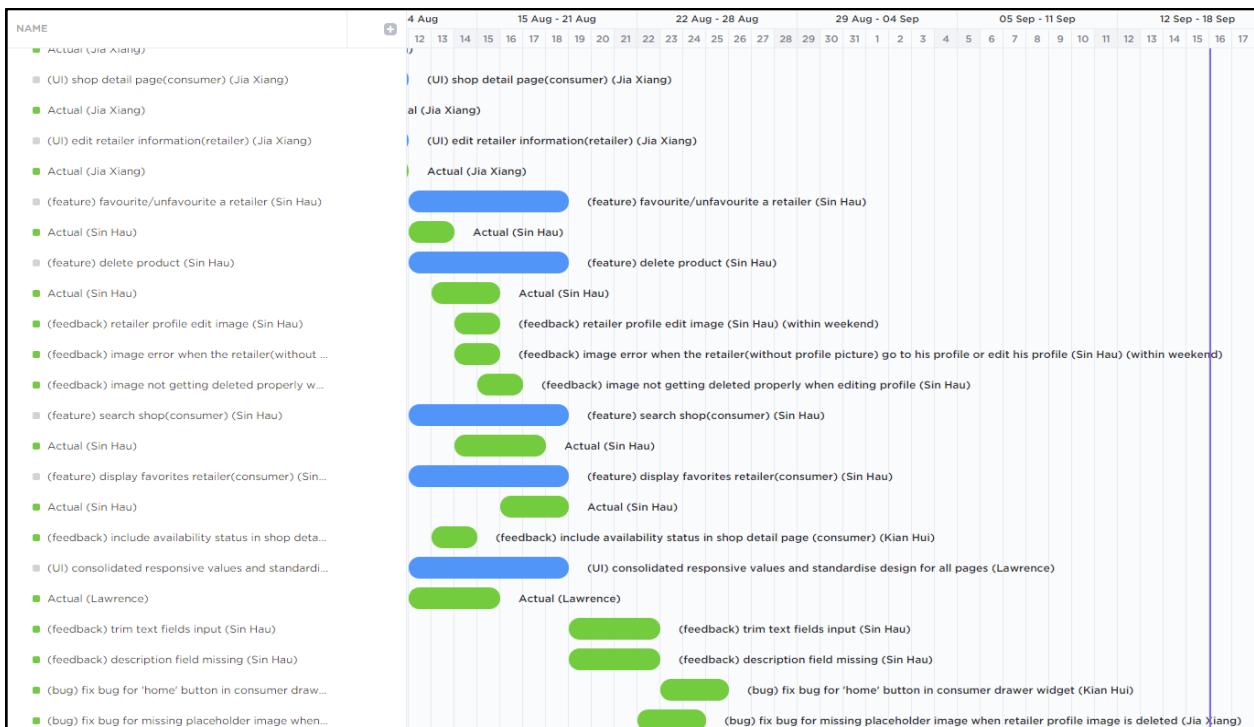
[5 August - 11 August 2021]

Sprint 4



[12 August - 18 August 2021]

Sprint 5



[19 August - 25 August 2021]

Task allocation for report

Name	Responsibilities
Sin Hau (Team leader)	<ul style="list-style-type: none">• System development• Requirements• Use cases
Bao Wen	<ul style="list-style-type: none">• User Testing• Evaluation
Lawrence	<ul style="list-style-type: none">• Background• Evaluation• User guide• Conclusion
Kian Hui	<ul style="list-style-type: none">• Prototyping and Iteration• Final design• User guide
Jia Xiang	<ul style="list-style-type: none">• Planning and research• Evaluation

Role in software development

Name	Role
Sin Hau	Backend developer
Bao Wen	Frontend developer
Lawrence	Frontend developer
Kian Hui	Backend developer
Jia Xiang	Frontend developer

Task allocation for software development

Consumer

Page	Contributed members	
Sign in page	Front end	Lawrence
	Back end	Sin Hau
Sign up page	Front end	Lawrence
	Back end	Sin Hau
Home page	Front end	Lawrence
	Back end	Sin Hau, Kian Hui
Favourite page	Front end	Lawrence
	Back end	Sin Hau
Retailer detail page	Front end	Jia Xiang
	Back end	Sin Hau, Kian Hui

Retailer		
Page	Contributed members	
Sign in page	Front end	Lawrence
	Back end	Sin Hau
Sign up page	Front end	Bao Wen
	Back end	Sin Hau
Home page	Front end	Lawrence
	Back end	Sin Hau

Add product page	Front end	Bao Wen
	Back end	Kian Hui
Edit product page	Front end	Bao Wen
	Back end	Sin Hau
Profile page	Front end	Jia Xiang
	Back end	Sin Hau
Edit profile page	Front end	Jia Xiang
	Back end	Sin Hau

Resources

- Manpower:
 - A project team leader
 - 5 Computer Science students
- Skills:
 - Agile project management in Scrum
 - Figma prototyping
 - Software development, with specific domain knowledge in Flutter and Dart
 - Firebase services knowledge
- Equipment and Tools:
 - Flutter software development kit
 - Figma graphics editor
 - Android emulator software
 - Cloud Database

Stakeholders

- Developers: Person who develops the application, fixing bugs based on user's feedback. Update / implementation of new features of the app

- Retail owners: Users that will be producing the foods for customers to choose
- Consumers: Users that will be selecting retail owners to transact with
- Employee: Any person(s) that is working under the restaurant or handling the maintenance of the app

Methodologies and techniques used

For this project, we have adopted the Agile Model as the development methodology. Agile is a versatile and realistic approach that allows for development of the software to proceed with small, incremental changes, even late in the development. This model places emphasis on User Centric Design (UCD) and works closely together with customers to look for cross-functional and adaptive changes. A core part of UCD involves thinking in the shoes of the customer, and the feedback received is continually inserted into the development process and iteratively implemented.

The Agile Model is suitable for this project because requirements are constantly evolving through the assessment of end users. As this project is a food discount listing application, user requirements are highly important. User scale is needed for retailers to reach as many potential customers as possible. Based on their feedback, we would be able to improve the application by implementing new features or removing obsolete ones. The Agile model is much more adjustable compared to the sequential waterfall model, which requires each phase to be completed before moving on to the next. It is flexible and responds to changes reactively. However, in this given context, problems in each phase may not be fully completed due to the close interaction with end users.

Scrum was used as the tool to implement the Agile Model framework. As a lightweight development strategy, the team incorporated certain Scrum processes into the development management. Scrum requires a Scrum Master and Product Owner, however we did not have those roles, certain responsibilities pertaining to those roles were assumed by the group leader and decision making often discussed with team input. Sprints, a facilitated working duration ranging from 1 to 4 weeks was implemented as a 1 week duration. The team felt that it was more practical to size sprints in shorter spans as it enables focussed development on a smaller set of priorities. This was also to enable us to stay on schedule. A sprint meeting was held before the start of a sprint. Within a sprint, frequency of daily Scrum meetings were also reduced to alternate days, as we had workload from other degree modules as well, it was not feasible to expect much progress within a single day. Trello were used to implement Scrum boards, with activities during the periods of Sprints updated in Trello. As Scrum is used as part of our agile oriented process, we took certain concepts from those frameworks and adapted it to our requirements. We did not follow the Scrum techniques exactly as it would be too restrictive but we retained the core essence of Scrum in order to work productively and agile.

3.0 Prototyping & Iteration

Both the low and high fidelity prototype built on a free online software tool, Figma. Figma allows us to build the wireframe and prototype.

Strategies and techniques

Different strategies are applied for designing and developing both low and high fidelity prototypes.

For low fidelity prototype designs, we emphasize on the functional elements of the application and ensure they are illustrated in the prototype. For high fidelity prototype designs, we emphasize on the UI/UX design aspects of the application. Some design concepts used are CRAP (Contrast, Repetition, Alignment, Proximity), visual hierarchy and visual journey.

The main font and sub font used in the application are Cantarell and Averia serif libre respectively. The colour scheme selected is soft red, light gray and white of the light and soft colour scheme. These decisions will be justified under the design justification section.

- **Cantarell:** <https://fonts.google.com/specimen/Cantarell>
- **Averia serif libre:** <https://fonts.google.com/specimen/Averia+Serif+Libre>
- **Colour Schemes:** Light and soft colour scheme. Soft red, light gray and white

Prototype Dimensions

Low fidelity - Google Nexus 5X

Reason:

- Relatively small compared to newer larger phone models
- UI/UX design of low fidelity prototypes is not the main focus, smaller dimensions help us lower development time of prototype

High fidelity - Google Pixel 2

Reason:

- Similar dimensions to most modern phone models.
- UI/UX design principles will be practiced
- User feedback collected used to make improvements to future versions

Prototype Sources

The provided sources display a demonstration of the app on the Google Nexus 5X and Google Pixel 2 device. The prototype will navigate to the respective pages when the buttons are clicked. This provides an experience that is very similar to the final product that is to be developed.

The screenshot shows the Figma interface with the 'Pages' tab selected. Below it, three prototype versions are listed: 'Version 1.0' (selected), 'Version 2.0', and 'Version 3.0'. At the bottom of the list, there is a blue, underlined link labeled 'Navigating Prototype Versions'.

Pages	+
✓ Version 1.0	
Version 2.0	
Version 3.0	

Navigating Prototype Versions

3 versions of Lo-Fi prototypes are improved based on user feedback received and they can be accessed in Figma on the top left box under pages.

Lo-Fi Prototype

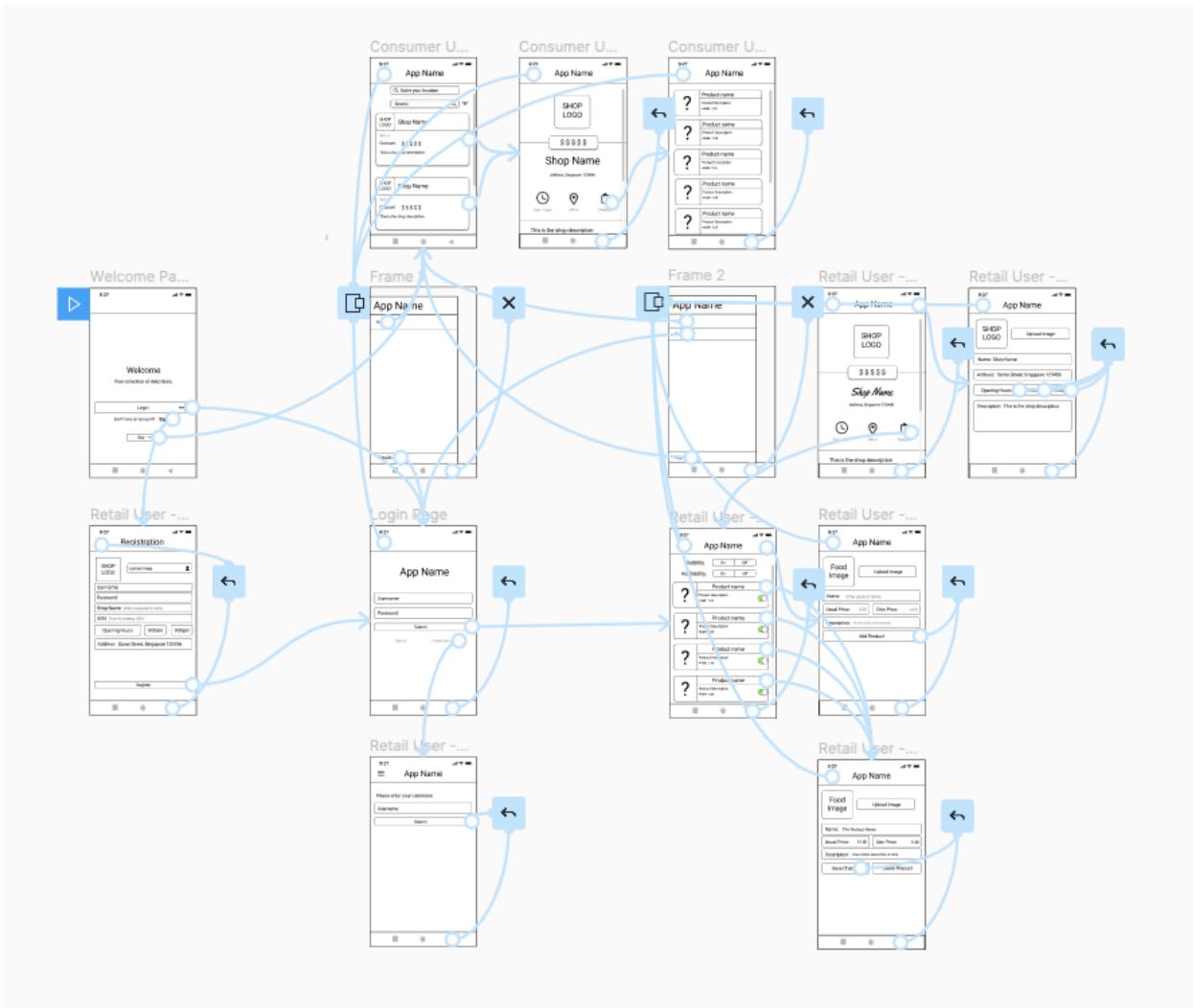
V1.0

- Prototype: <https://www.figma.com/proto/QGGYjGRb81bGE0eCbXqqLo/Lo-fi?node-id=141%3A2&scaling=scale-down&page-id=0%3A1>
- Wireframe: <https://www.figma.com/file/QGGYjGRb81bGE0eCbXqqLo/Lo-fi?node-id=0%3A1>

Our first interface design consists of keyframes and views covering the functions of the application. Interfaces for users and retailers are parallel to each other, with formations being mapped between keyframes that correspond to respective maneuvers.

Following this instance of the prototype, we intend to run 2 more iterations for the lo-fi interface, focusing on the usability and efficiency of the application's functions. We aim to procure a well rounded structure of the application before proceeding to develop the hi-fi prototype, where we will focus on the graphical representation of the application.

Ensuing our Agile software approach, we conducted surveys and interviews to gather in-depth qualitative and quantitative responses, and reviewed them together as a team.



Overview of Keyframe views

V2.0

- Prototype: <https://www.figma.com/proto/QGGYjGRb81bGE0eCbXqqLo/Lo-fi?node-id=274%3A2&scaling=scale-down&page-id=274%3A2>
- Wireframe: <https://www.figma.com/file/QGGYjGRb81bGE0eCbXqqLo/Lo-fi?node-id=274%3A2>

After reviewing Prototype v1.0, we reassessed the usability and effectiveness of the interface. In the second iteration of the prototype, upon the team's review and responses from a conducted survey, the following notable changes were made:

Retailers' catalog

- Container size for retailers reduced to accommodate for more listings displayed.
- Dollar figures "\$" representing price/discount ratings removed as it was deemed unnecessary by most surveyors.
- Surveyor feedback indicated that having the opening hours of the shops would be beneficial.



Retail's Description

- Retailer's name in the app bar was deemed redundant by replacing the application's name in the heading to the retailer's name.
- Icons shifted up to fill the gap.
- Included address information of retailer.
- Pricing/Discount rating indicator removed

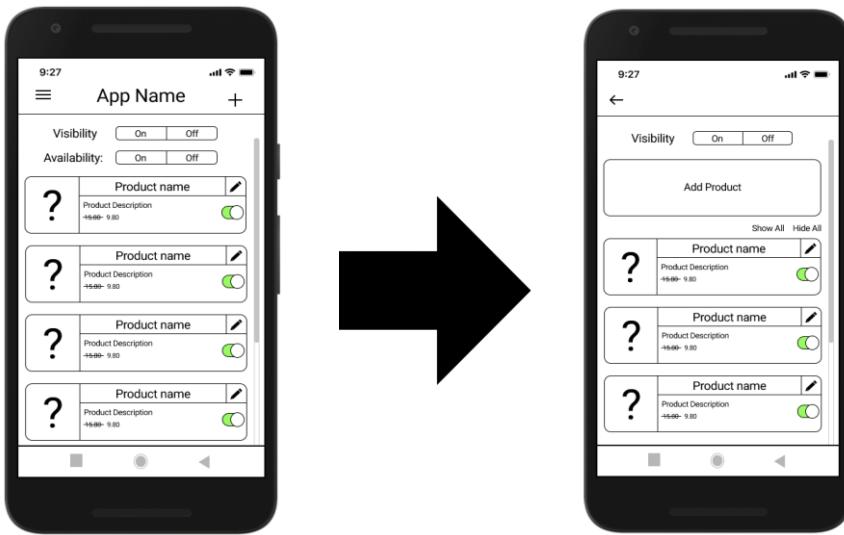


Prototype v1.0

Prototype v2.0

Retailers' product catalog page

- Replaced availability options with 'show-all/hide' options. This was reviewed as a minor option, and the size reduction declutters the options menu at the top.
- Replaced 'add product' button from a '+' icon at the top right to a dummy container sized button. This function was reviewed as a significant feature, hence the improvement for a more prominent presentation.



Prototype v1.0

Prototype v2.0

Some other minor changes that were made to improve overall presentability:

Welcome page

- ‘Skip’ button shifted closer to login button

Account register page

- ‘Sign-up’ text in register button changed to ‘Register’

Registration page

- Added in form fields for address of retailer

Forget Password page

- Changed page title from ‘App name’ to ‘Forget Password’

V3.0

- Prototype: <https://www.figma.com/proto/QGGYjGRb81bGE0eCbXqqLo/Lo-fi?node-id=274%3A817&scaling=scale-down&page-id=274%3A817>
- Wireframe: <https://www.figma.com/file/QGGYjGRb81bGE0eCbXqqLo/Lo-fi?node-id=274%3A817>

Retailers’ profile page

- Retailers had to use the hamburger menu on the left of the app bar to access their profile. As some found it difficult to navigate with that, a bottom navigation bar was added for retailers to navigate effectively from their catalog page to their profile page.
- Added an accessible navigation bar with icons at the bottom of the page.



User navigation to food listing

- The sequence of the pages was not in the most optimum positions, and the user's navigation process of the food listings could be improved to be more efficient. With the updated sequence, users will spend less time navigating to access the product catalog.

In V2.0, the mapping sequence was:

- shop catalog → shop description → product catalog



In V3.0, the mapping sequence was updated to:

- shop → products (with shop description options link)



Some other minor changes that were made to improve overall presentability:

Retail User - Products Page

- Retailer's name added to the middle of the app bar.

Retail User - Add Product Page & Retail User - Edit Product Page

- Retailer's name added to the middle of the app bar.

Findings and Evaluation

Lo-Fi Prototype development phase

The sources of our user assessment stems from surveys to gather feedback regarding overall UI/UX of the application.

Application feedback:

- Based on the application's interface, what do consumers think the application is for?
- Ratings for application's interface
- Ratings for application's readability
- Ratings for user experience
- Would they use the application?
- Best aspect of the application
- Worst aspect of the application

Survey forms links:

- Version 1: https://docs.google.com/forms/d/1yy1E1Oqx-mE1p_UoZI5nAPasLYspoborjYmujRIR_kc/edit?ts=60e6b970#responses
- Version 2: <https://docs.google.com/forms/d/1XEHqK4XMDC-MCP0J5Mlp-cugC2doDS4QA7kl7cxUJ4o/edit?ts=60e6b956#responses>
- Version 3: <https://docs.google.com/forms/d/1IJYYJtwUyQN9ftwXFQXxYi-FBF4gak6mUNDUwgUPHkY/edit?ts=60e5fa40#responses>

Hi-Fi Prototype

V1.0

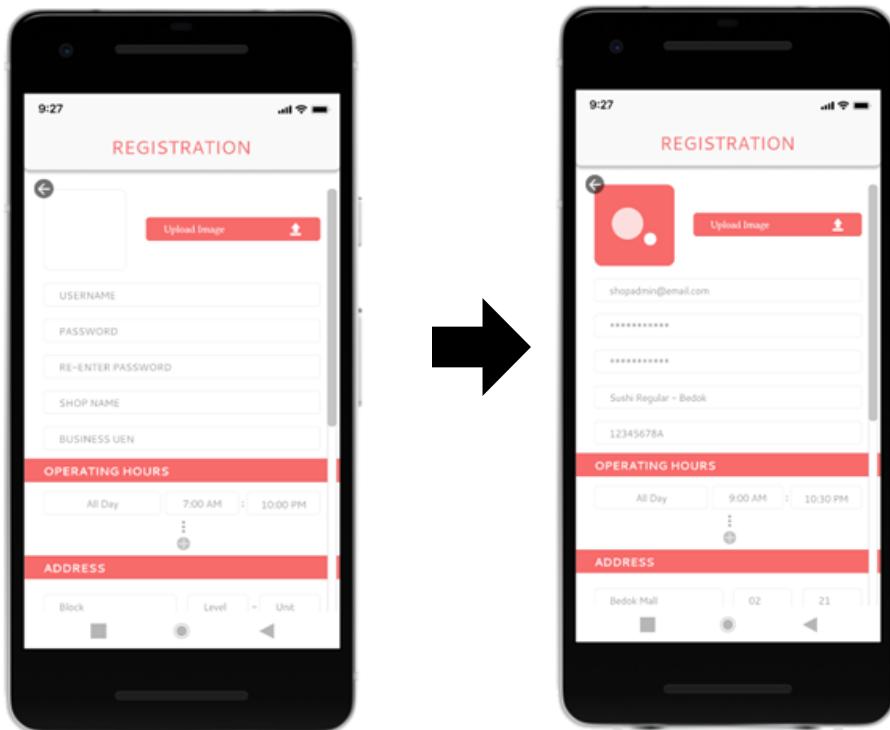
- Prototype: <https://www.figma.com/proto/ydr0N3qhP3d9sjWAwJbxU/Hi-fi?node-id=1%3A2&scaling=scale-down&page-id=0%3A1>
- Wireframe: <https://www.figma.com/file/ydr0N3qhP3d9sjWAwJbxU/Hi-fi?node-id=0%3A1>

Succeeding the 3rd iteration of the lo-fi prototype, the Initial graphical user interface(GUI) design implementation was improved, with more robust function workflows inclusive of sample input representations (dummy data for text forms).

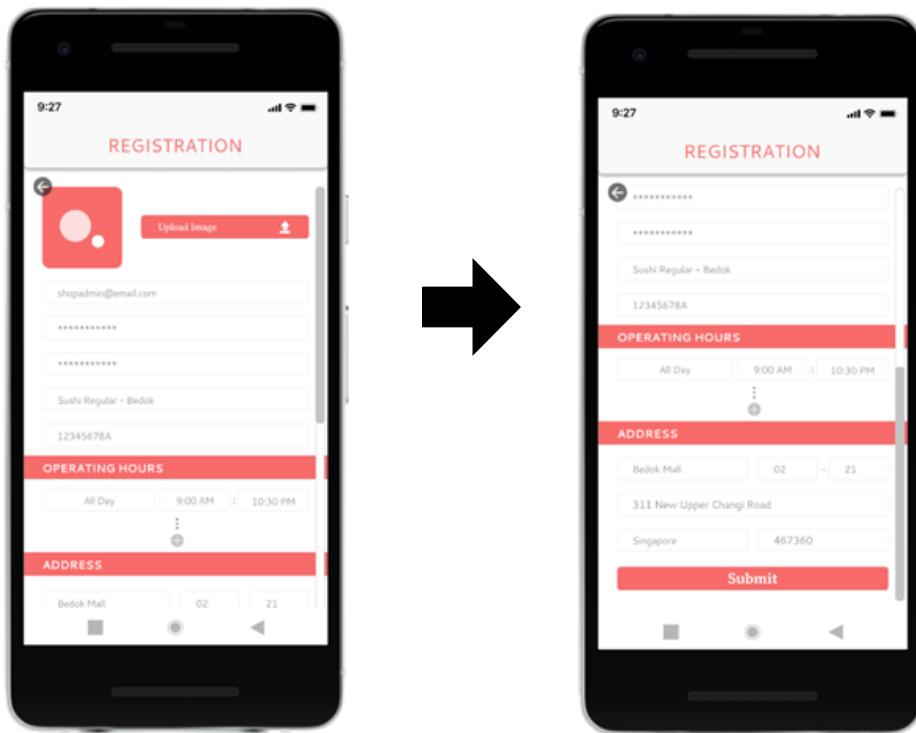
In the high fidelity prototype, we can interact with form fields, buttons and scroll bars to navigate through each of the pages. Elements in the prototype can be interacted with by clicking on the respective elements. When clicking on elements with no interactions, all clickable elements will be highlighted briefly. The following interactions were added:

- Added in clickable fields to switch between populated and unpopulated fields
- Form submissions available only when fields are populated (validation or hint messages not shown)
- Added in clickable scroll bars to simulate scrolling of page
- Switch buttons to toggle availability of items and visibility of shop to users

Populating form fields with value (click on any of the fields):



Scrolling down a page with scroll bar (click on scroll bar):



V2.0

- Prototype: <https://www.figma.com/proto/ydr0N3qhP3d9sjWAwJbxU/Hi-fi?node-id=315%3A3&scaling=scale-down&page-id=315%3A2>
- Wireframe: <https://www.figma.com/file/ydr0N3qhP3d9sjWAwJbxU/Hi-fi?node-id=315%3A2>

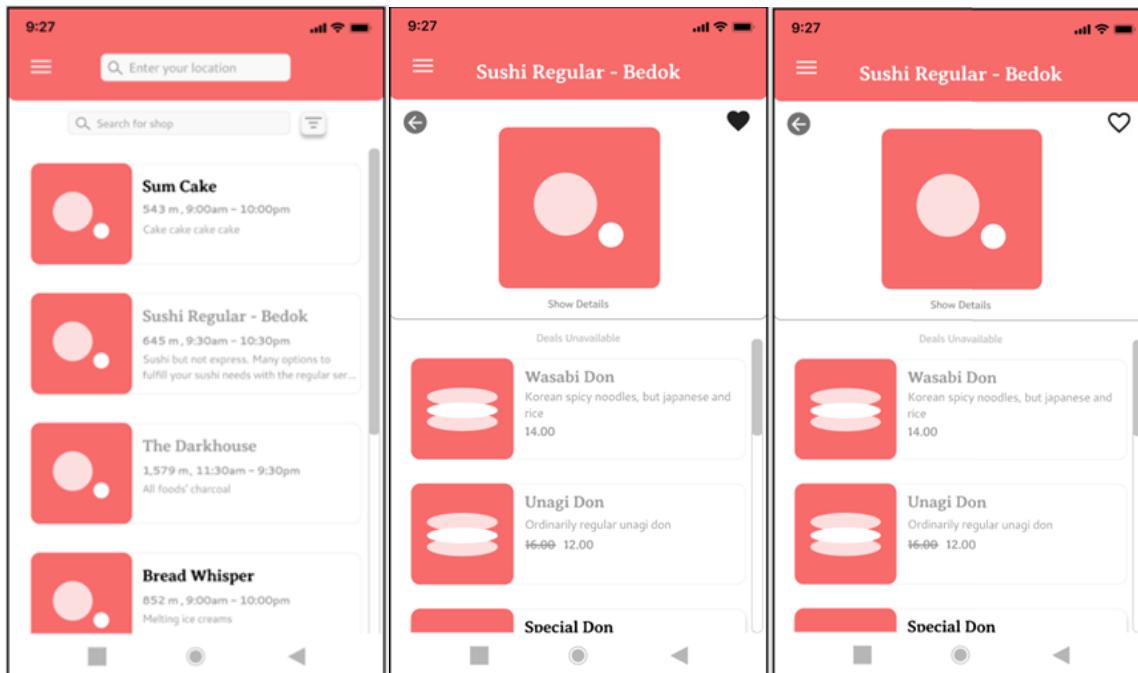
A new feature was added, with certain graphical improvements.

- Favourites function
- Explicit instructions for login process, deals availability for retailer, and visibility toggle button
- Removal of bottom navigation bar

Favourites function:

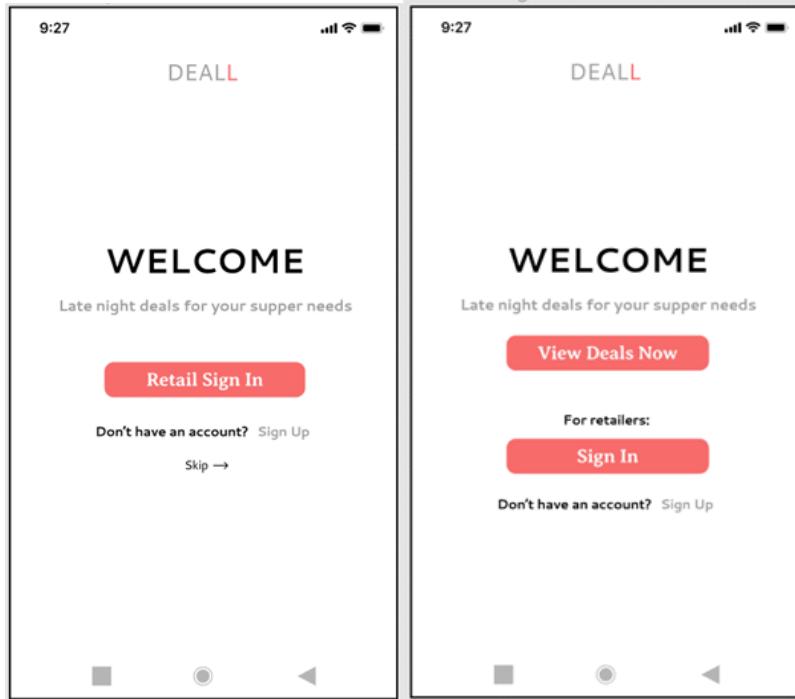
Feedbacks received through interviews indicates that having a feature to store retailers that consumers like will enable faster access to deals they want. All retailers added to the users' favourites list will be displayed in the page regardless of whether the retailer chooses to turn its visibility to public on or off. Some changes made to the pages:

- Retailers with visibility turned off will have its shop greyed out
- Clickable icon to add(black) or remove(white) retailer from favourites list



Welcome Page changes:

Feedback suggests that “Skip” instruction was confusing and a more explicit instruction was requested. The new welcome page now has a “View Deals Now” button. Based on the concept of visual hierarchy, it would be more intuitive to click on the “View Deals Now” button. There is also a text above the sign in button that states “For retailers:”.



Other changes:

Multiple feedback states that the app is cluttered. Since the hamburger menu is known universally as the “main menu” for navigation, we decided that the bottom navigation bar can be removed without affecting the overall functionality of the app. Some other changes that occurred are:

- Toast message indicating successful changes made to products. (Retailers feedback).
- “Visibility” -> “Make Shop Public?” for clearer understanding of toggle usage
- Indicator of deal availability in food catalog page for each retailer
- “Add product” container is reduced to “+” icon with “add product” text which is aligned with “show all” and “hide all” product buttons for cleaner UI
- Other minor changes to overall app UI made to follow design principles

Some survey results were collected from consumers and retailers and they can be found here:

- HiFi (consumer): https://docs.google.com/forms/d/1xeaX-5S_3qOM3ZzWrYE_mqCYnYp81uePhtGATL45acM/edit?ts=60e5fa64#responses

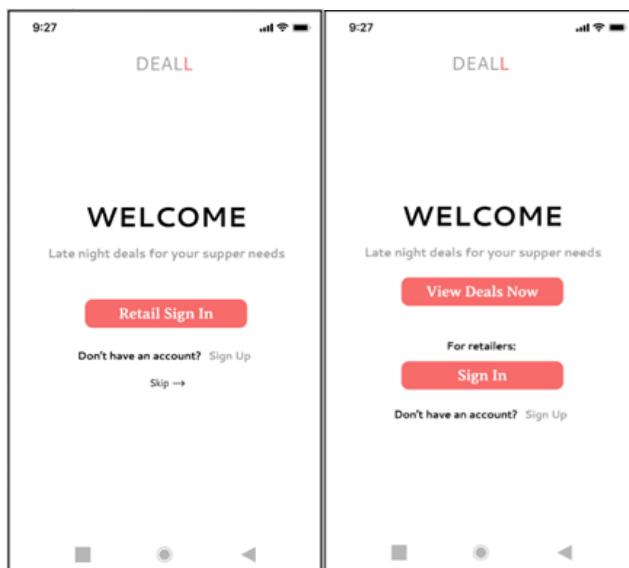
- HiFi (retailer):
https://docs.google.com/forms/d/1k8kelu_j7RnX1hpLNgXo4SEqqwMzeNsGufuFilwpJPU/edit?ts=60e72335#responses

Final design

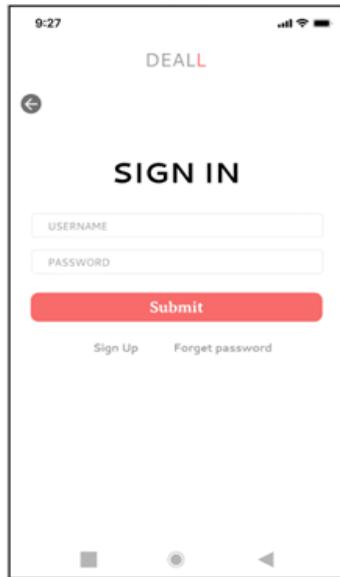
Design iterations for sign in/sign up process:

The initial design of the Hi-Fi prototype consists of a welcome page where users will press certain buttons to proceed to either sign in, sign up or skip signing in for consumer users.

The feedback received for the Hi-Fi v1 prototype was that the 'skip' button was not intuitive and in v2, we changed it to 'View Deals Now'. Based on the concept of visual hierarchy, users will see the 'View Deals Now' button first followed by the sign in button for retailers. When the 'View Deals Now' button is clicked, users will be brought to the consumer home page.

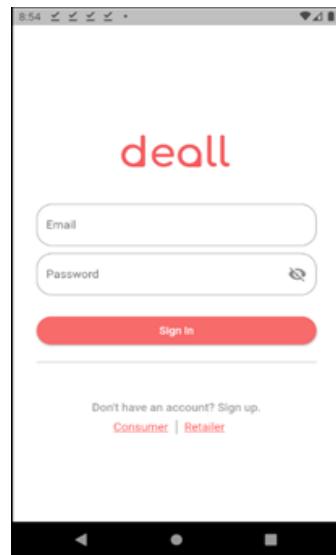


When users click on the sign in button, they will be brought to the page shown below.



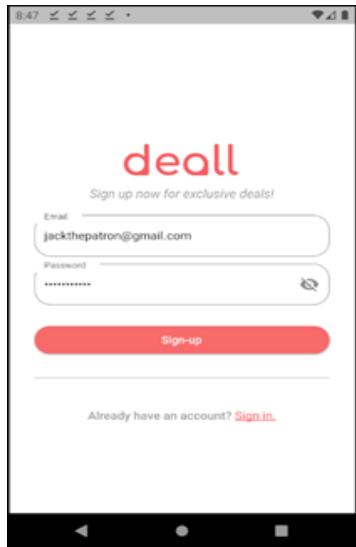
However, with the inclusion of the ‘favourites’ feature for users, we have made it a requirement for consumers to sign up for an account. The information required to sign up for consumers and retailers are different and thus, we have made it intuitive for the users by specifying “Consumers” or “Retailers” sign up.

As both user types will need to sign in in order to use the features in the app, the welcome page was removed, and users will be brought to the sign in page immediately after opening the app.



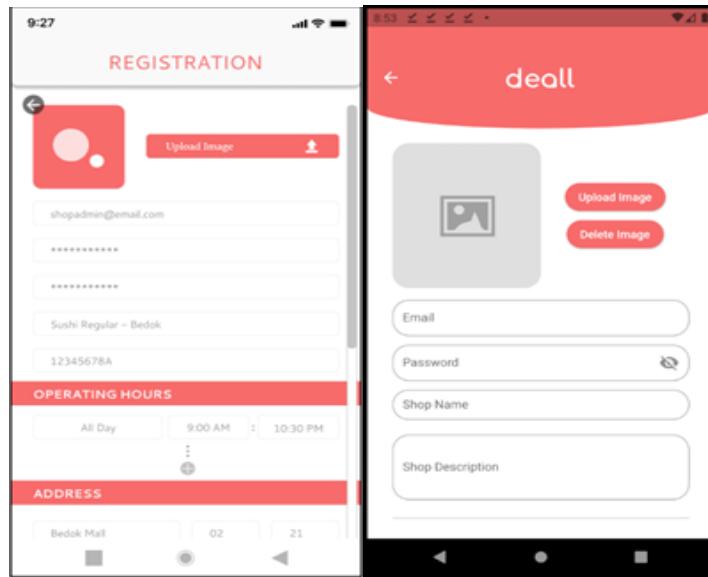
Design iterations for sign up page(consumer):

The Hi-Fi prototype does not consist of any sign up for consumers and thus does not have a design iteration for it. We ensured that primary actions such as the sign up button stand out and that the color used matches the color theme of the app. The final design is shown below:



Design iterations for sign up page(retailer):

The app bar for the final design of this page is changed to standardise the design and color theme of app bars. The design for every field in the sign-up form is also fixed to simply having a text field with the hint text indicating what that field is for. The operating hour field is also changed to a single text field where users will enter the details instead of the one designed in prototype. We have decided to let users decide the format of how they want to enter the details on their operating hours.

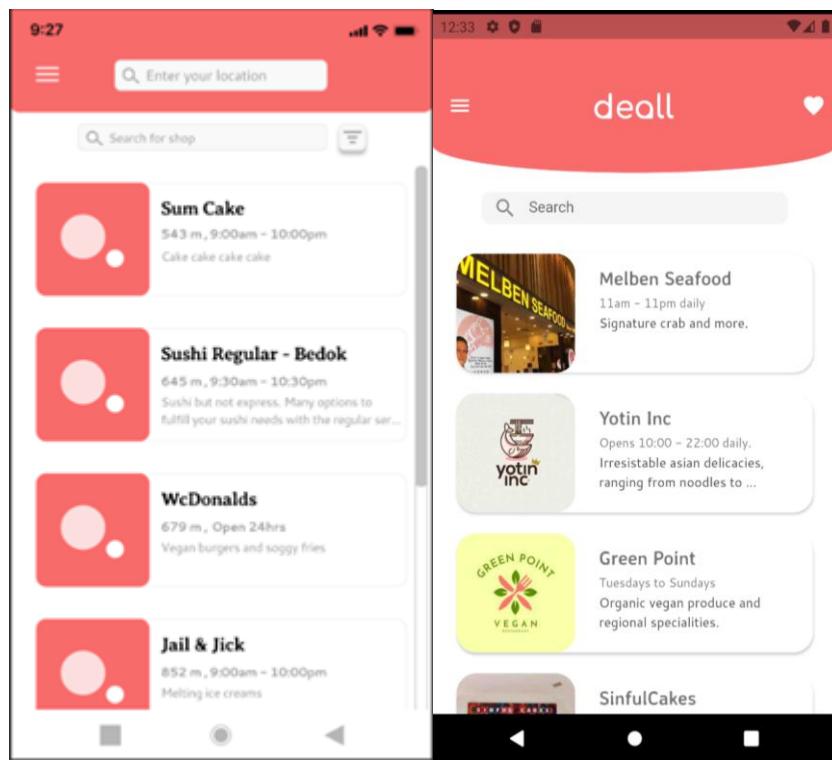


Design iterations for consumer home page(retailer list):

The initial design of the Lo-Fi prototype for this page consists of a bottom navigation bar but was later removed to reduce clutter. In order to navigate to the favourites list page in the Hi-Fi, the user needs to press the drawer menu on the top left and press the Favourites selection in the menu. In the new design iteration, they will have an additional option of pressing the favourite

icon on the top right corner which will directly bring them to the favourites page. This is an application of the UI design pattern of progressive disclosure.

The current design of the page will also replace the location search bar with the app logo as the distance and location feature will not be included. The filter button beside the shop search bar is also removed as the initial purpose was to allow users to filter between distance or alphabetical order. The app bar was given this specific shape and size for the application of the design concept of hand position controls, which is further explained in the design justifications of this report. This allows the search bar and retailer list to sit in a comfortable position where users can interact without stretching their thumbs too much. This app bar design is also applied throughout the app in other pages.



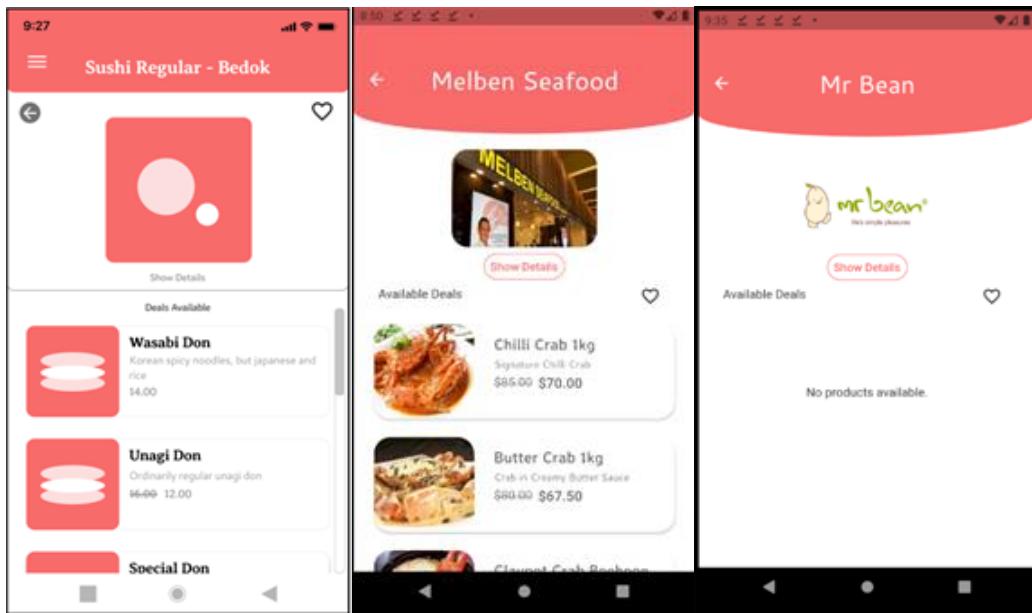
Design iterations for consumer product list page(product list of selected retailer):

Some notable changes in the Hi-Fi prototype from the final design are the positioning of back and favourite button, show details design and the position of deal availability.

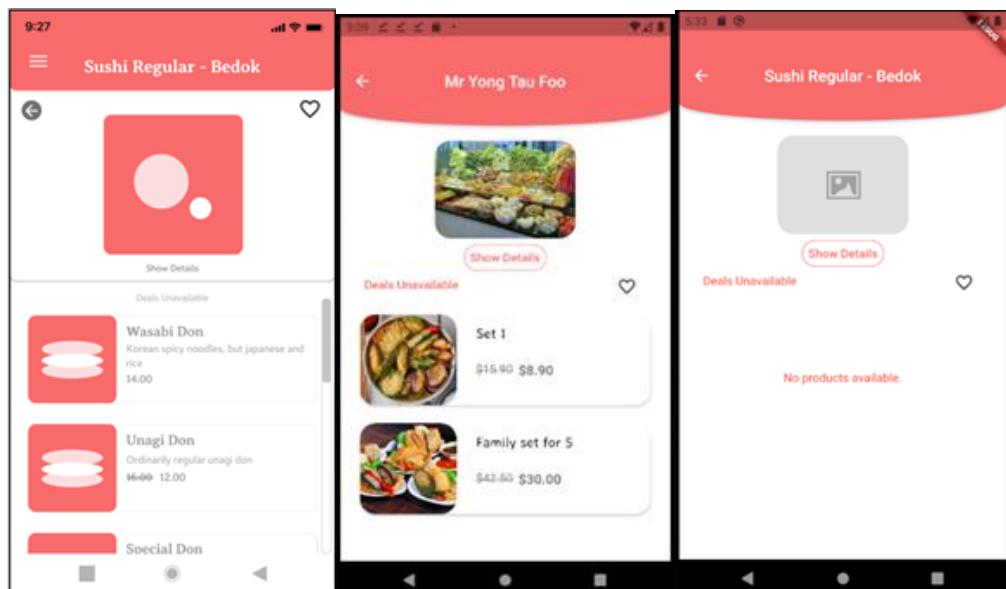
The back button replaced the drawer menu to restrict navigation through the home page only. The favourites button was lowered to make it easier for users to favourite the shop and borders were added to "show details" to make it more intuitive that it can be interacted with. This is the solution to the problem of progressive disclosure which is further discussed in the UI/UX patterns of this report. We have also moved the "Available Deals"(visibility) of the shop to be aligned with the favourite button, followed by the products list below it. If no products are available, we simply

display a centered text. If visibility is off, “Available Deals” text will change to “Deals Unavailable”. The stated changes are shown below.

Visibility On (L -> R: Hi-Fi v2, visibility on with products, visibility on without products):



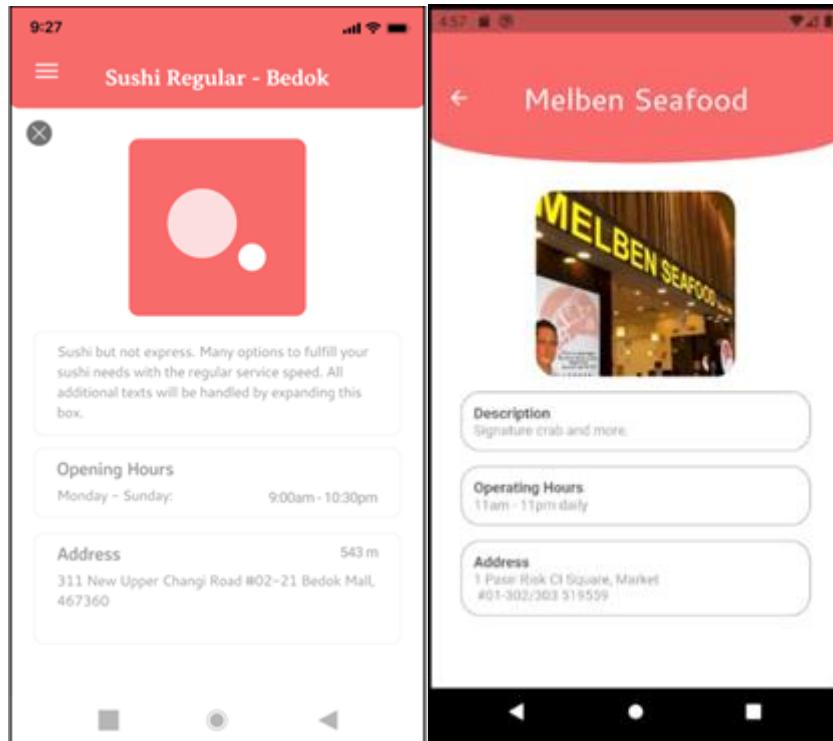
Visibility Off (L -> R: Hi-Fi v2, visibility off with products, visibility off without products):



Design iterations for consumer - retailer detail page(detail page of selected retailer):

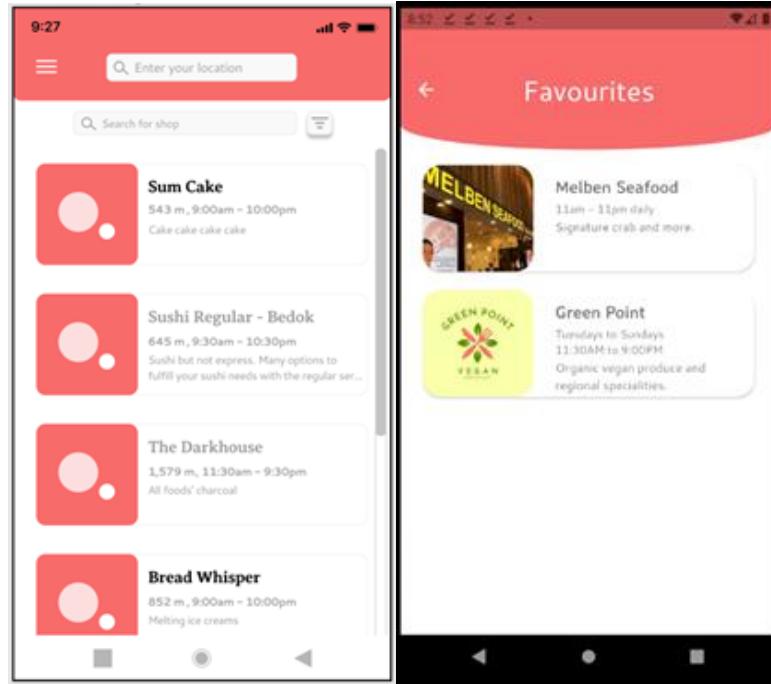
The final design of the retailer detail page does not differ too much from the prototype. The back button replaced the drawer menu which is similar to the product list page, an additional 'Description' header and darker borders were added.

The distance text was removed as the distance feature will not be included and the format of the operating hours text display was changed to be exactly as how the retailer user enters the operating hours.



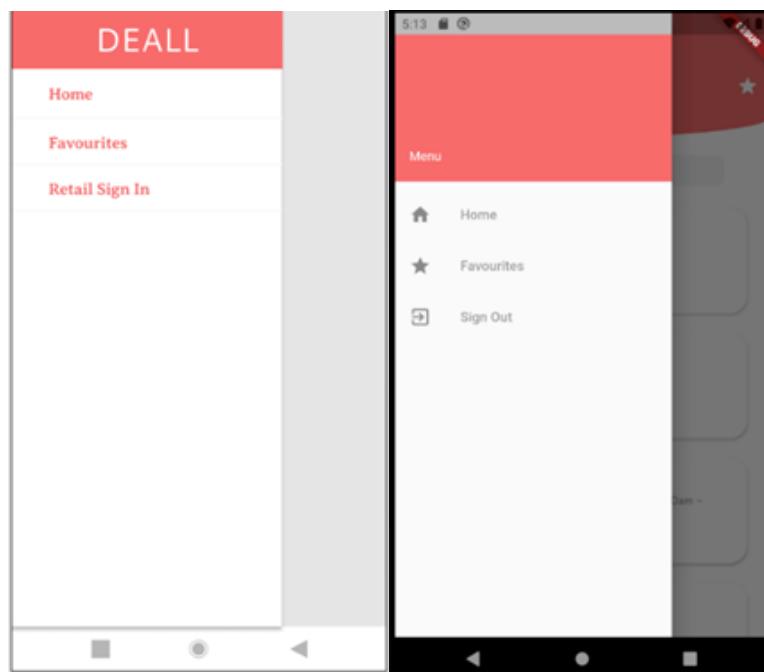
Design iterations for consumer favourites page:

The final design for consumer favourites page retailer does not have the search bars and only the retailers that were added to favourites previously will be displayed. The drawer menu is also replaced by a back button similar to other pages. Retailers who are added to the favourites list will be displayed regardless of the visibility of the shop.

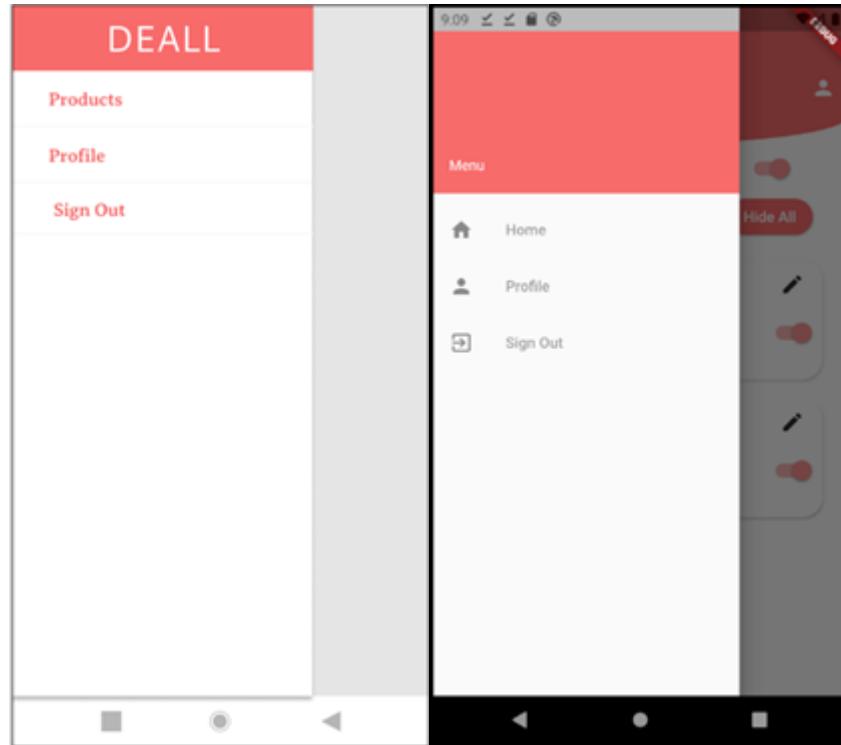


Design iterations for consumer and retailer drawer menu:

The drawer menu for consumer users were also changed to accommodate for the new sign in feature of consumers. A user will have to sign out from the consumer account and sign in again if they want to use the device to log into another consumer/retailer account. Icon was also included to make the navigation much more intuitive. Similarly, the app bar of the menu was enlarged to accommodate for more ease of pressing the elements in the menu.



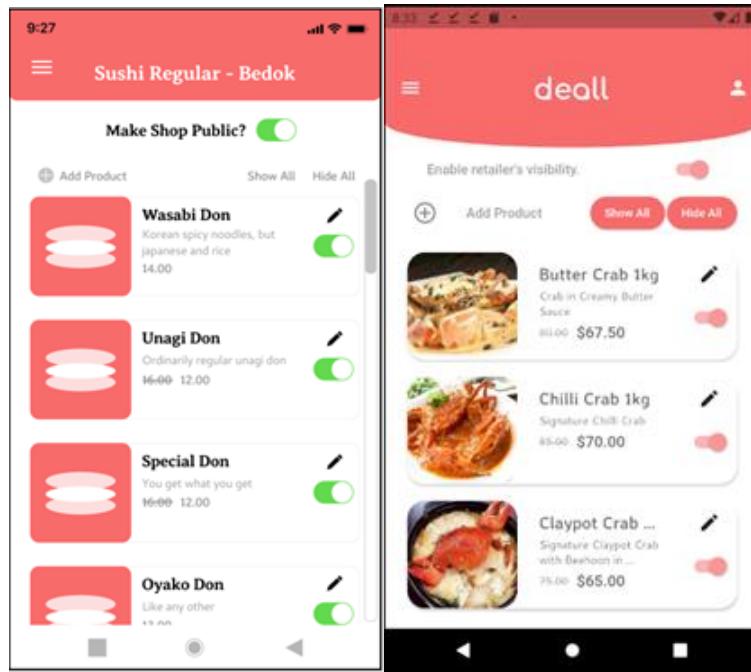
The only changes to retailer users' drawer menu are that 'Products' is renamed to 'Home' and icons were added similar to the consumer drawer.



Design iterations for retailer home page(products list page):

The first changes made was that the app bar title was fixed as the app name. Since the user logging into the retailer will naturally know the shop name, there is no need for the shop name to be viewed on the app bar. If any changes need to be made, they can go to the profile page to edit it.

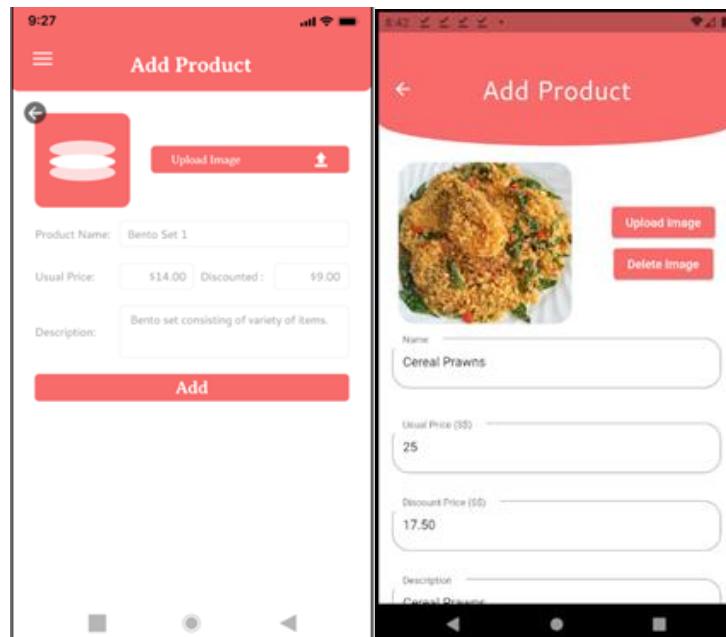
The 'Make Shop Public?' text was replaced with 'Enable retailer's visibility' and the show and hide all buttons were given a border and elevation. The profile icon was added to the top-right-hand corner for quicker navigation to the profile page which can still be navigated to through the drawer menu. The switch colour was also changed to fit the colour theme of the app.

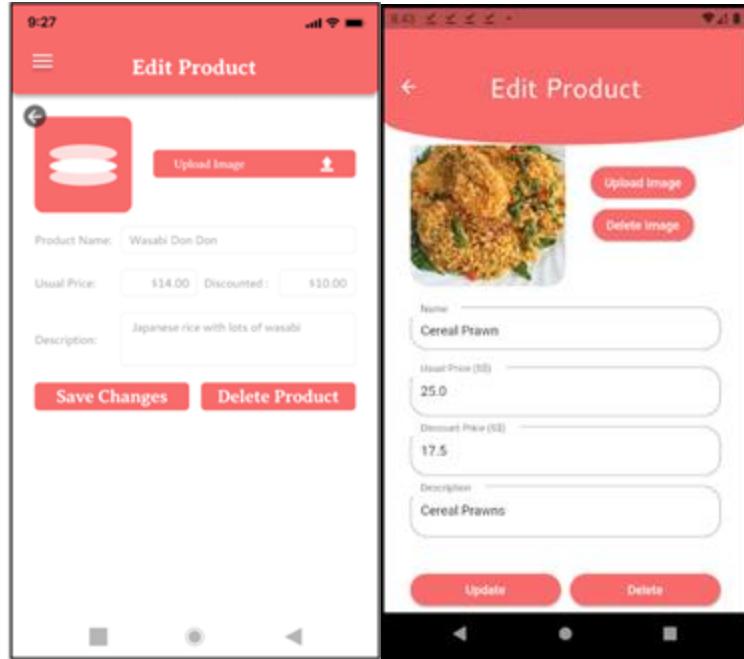


Design iterations for retailer add and edit product page:

The design for add and edit product pages are very similar. The difference in design compared to the Hi-Fi prototype for the add product page is that we enlarged the image, reduced the size of the upload image button, and added an additional delete image button. The design of the fields was changed to the ones available on flutter where the label text of the field is used to show the field names.

The difference for the edit product page is similar to the above and the additional difference is that the buttons “Save Changes” and “Delete Product” are changed to “Update” and “Delete”.

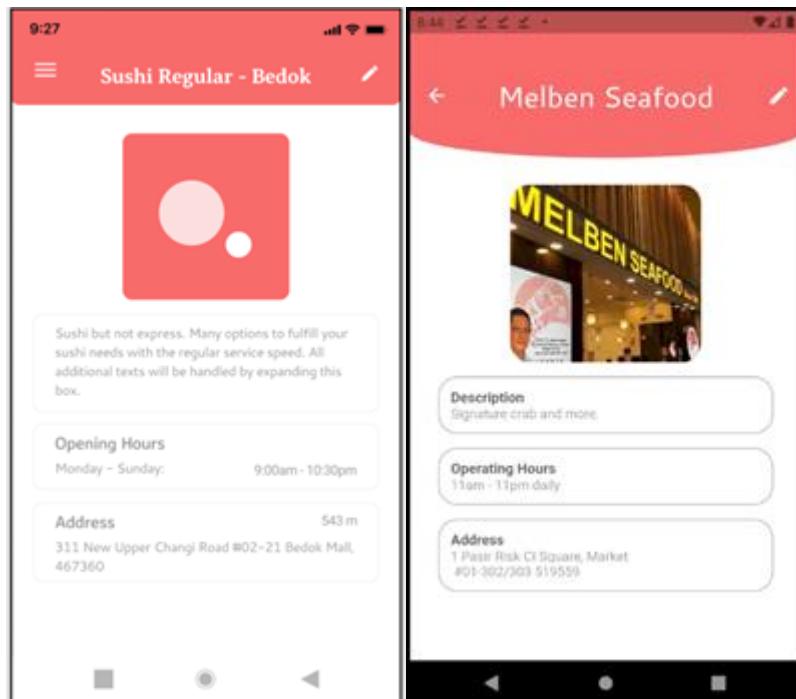




Design iterations for retailer profile page:

The final design for the retailer profile page is very similar to the consumer – retailer detail page design and has the same differences from the prototype as that page.

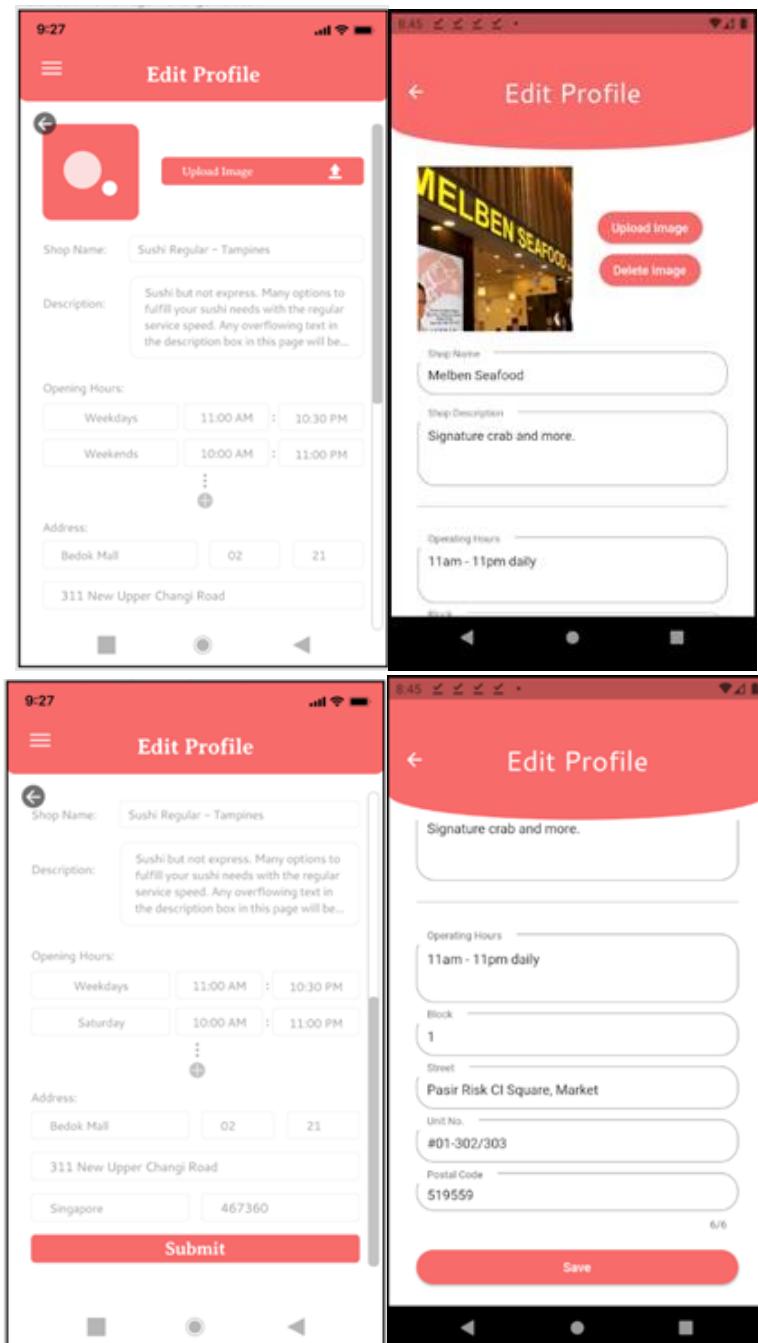
However, in the retailer profile page, there is an edit icon button on the top right hand button that will bring users directly to the edit profile page.



Design iterations for retailer edit profile page:

Similar to the other pages, the final design of the retailer edit profile page differs from the prototype in that the image is enlarged with the additional delete button. Fields are also redesigned to use the ones available on flutter with a label text and the field value.

Similar to the retailer sign up page, the operating hours is simply changed to a text field and instead of simply displaying address along with the fields, we have added a label text for all respective fields (block, street, unit no., postal code). 'Submit' text is also replaced with 'Save'.



4.0 Final Design

Specifications

High fidelity prototypes - designing principles and techniques used:

- CRAP (Contrast, Repetition, Alignment, Proximity)
- visual hierarchy
- visual journey

Final design - UI/UX patterns with multiple design principles:

- Progressive disclosure/Content prioritisation
- Clear primary actions
- Intuitive mobile app design
- Hand position controls
- Legible fonts and texts
- Color theories

Main Font: Cantarell

Sub Font: Averia serif libre

- **Cantarell:** <https://fonts.google.com/specimen/Cantarell>
- **Averia serif libre:** <https://fonts.google.com/specimen/Averia+Serif+Libre>
- **Colour Schemes:** Light and soft colour scheme. Soft red, light gray and white

UI/UX Patterns

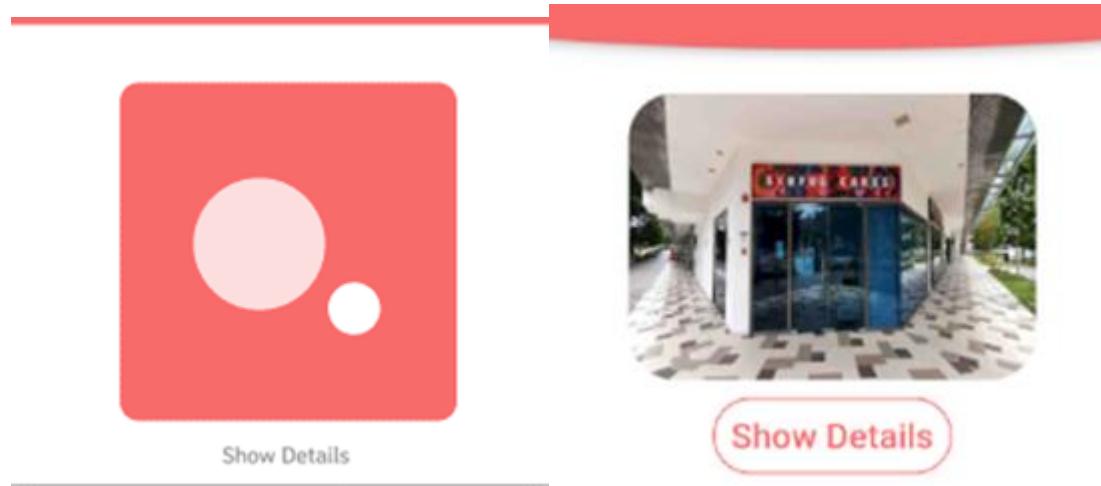
Progressive disclosure

One of the UI design patterns used is progressive disclosure (What are User Interface (UI) Design Patterns?, 2021). Progressive disclosure is the concept of hiding details from users until they are requested or required. This allows users to focus and spend more time on important features, helping them to understand the use of the app much easier (Babich, N, 2019, August).

One example of practising this concept is the change of the bottom navigation bar, to a hamburger menu, so that it will appear only when users want to navigate to a different page.

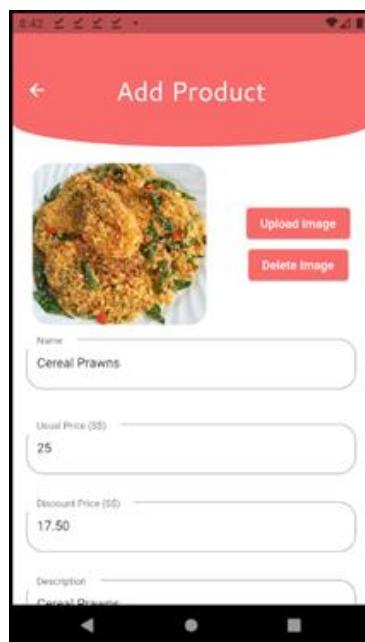
One other example is that the consumer home page displays only the basic information of each retailer instead of all details. If users are interested, they can press any of the retailers, and more information such as products list or shop detail will be displayed.

However, with progressive disclosure comes the problem of discoverability where contents that are hidden should still be easily accessed by users with sufficient contextual details. As such, the show detail text button was changed from a single text button to a text button with coloured borders.



Clear primary actions

The other UI design pattern used is clear primary action. Clear primary action is the concept of making buttons stand out so that users know what can be done. The practice of this concept can be seen in the buttons throughout the app. One example is the add product page where all pressable buttons are coloured with borders and texts:



Design Justification

Some design principles (Experience, A., 2018 December) that are used in our project are:

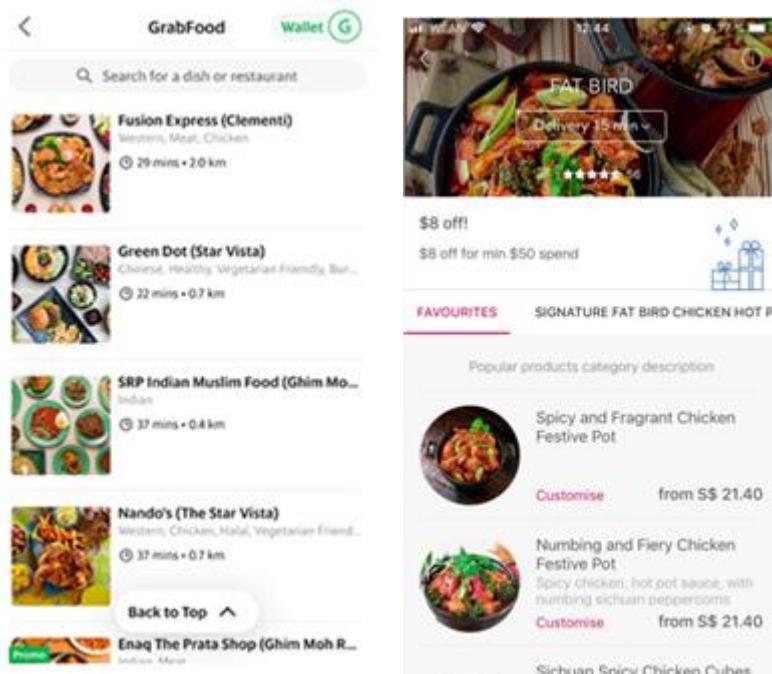
1. Content prioritization:

Removed bottom navigation bar to minimise clutter or interface elements to keep users engaged. All secondary contents are made accessible using the drawer menu.

2. Intuitive Mobile App Design:

Having a similar application interface design to popular and commonly used applications will allow users to learn and use the app with relative ease and comfort as it provides some familiarity.

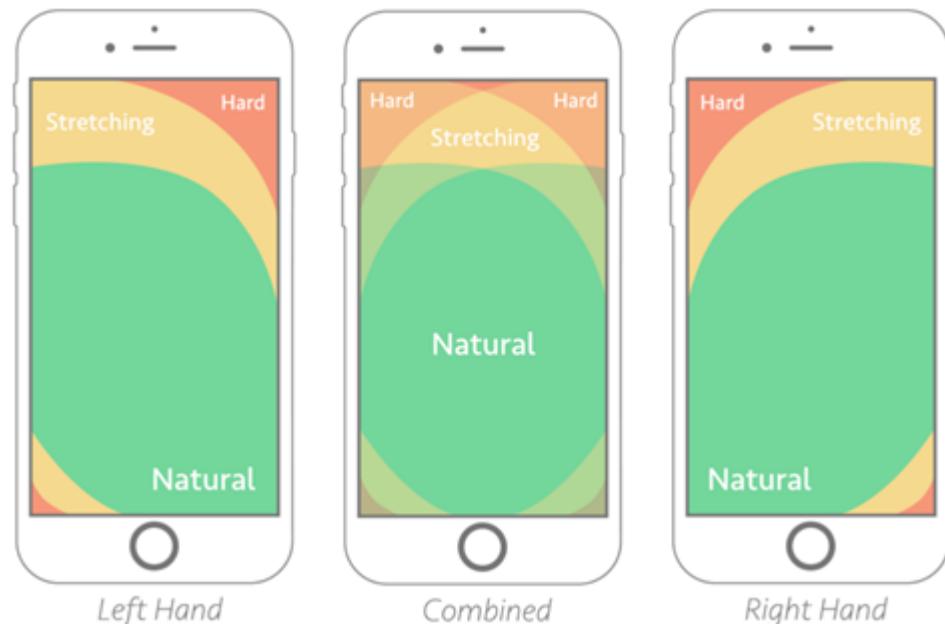
References used in the design of the DEALL app are from popular food delivery apps in Singapore such as GrabFood and FoodPanda. As DEALL and food delivery apps are related to the food industry, we can expect some or most of the user base to have similarities. Image on the left shows the restaurant list for the GrabFood app while the image on the right shows the shop detail page of the FoodPanda app.



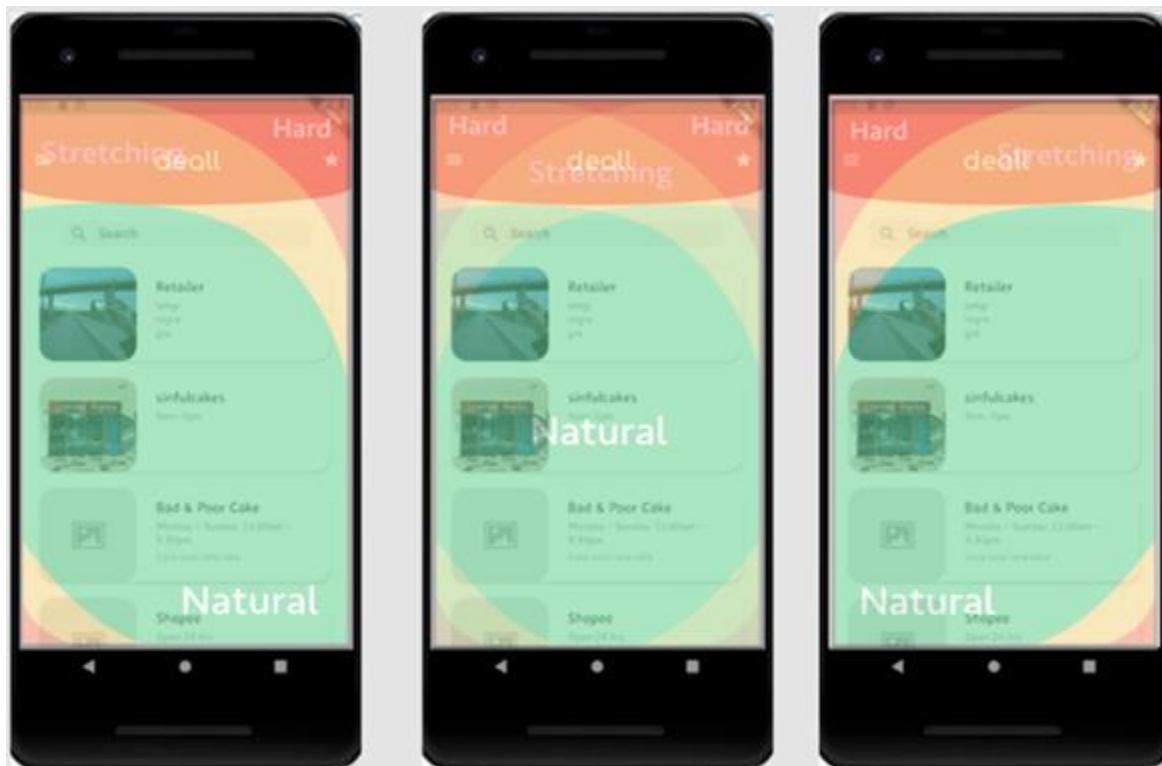
3. Hand Position Controls

One important aspect in the design of mobile apps is the position of interface elements such that users are able to interact comfortably. In the figure below, it shows the thumb

zone of an average user(Kolieboi, A., 2019, April). Even though people differ in the way they hold and use their devices, it still provides us with a good reference for the majority of the users.



An example of applying this concept to our app's design is the consumer home page - retailer list as shown in the figure below.



The retailer list and search bar sits comfortably within the natural thumb movement region of users of different usage habits. As consumers will typically be browsing through the retailers and interacting with these elements for more details such as products sold by the retailer or retailer detail(address, opening hours, etc), the list would best be within the natural region. The drawer menu may require users to stretch their thumbs to access but it follows the typical design of virtually every app, having the menu on the app bar. However, consumer users may also access the hamburger menu by sliding the screen from the left to the right in the home page if stretching is uncomfortable for them, and the favourite page can also be accessed through the hamburger menu if the favourite icon is too uncomfortable or difficult to press.

4. Legible fonts and texts

Cantarell is a font under the sans-serif font family. It was selected as the main font as most modern fonts of that font family typically have a much better readability and they scale much more easily (Best fonts for mobile app design., 2020, April).

Averia serif libre, of the serif font family was selected as the sub font as this font is still highly readable and it is compatible with the other elements and design of the app.

5. Color theories

In the color wheel basics, colors separated in the center can be grouped to warm(red, orange, yellow) and cool(blue, green, purple) colors (Decker, K., 2020, July). As warm colors are usually associated with energy, brightness and action, we have chosen a warm color, red as the primary color for the DEALL app, encouraging users to take action and visit the listed retailers on the app.

Features Justification

Features for consumers

Features	Details
Browse retailers (consumer)	When a user(consumer) uses the DEALL app, they will want to browse for available deals from retailer shops that offer discounted food products. The retailer list will enable that by displaying all the retailers available.

Search for retailers	Allow users to search for a specific retailer when they are looking for a specific shop or when the available retailer list is too long to browse through all of them. One other reason is that retailers who have off their visibility can still be searched by interested consumers.
View retailer details	When users(consumers) are interested in a retailers' products, they will want to view the details of that retailer such as opening hours and address.
View available surplus food	When the users(consumers) first tap the retailer that they are interested in, they will want to see the list of available food products that are on discount which is why the product list is used to display them.
Mark retailer as favourite	Allow users(consumers) to mark multiple retailers as favourites so that they can easily access their favourite retailers through the favourite page without having to use the search bar to search for those retailers repeatedly.
Sign up & login(consumers)	Allow users(consumers) to sign up so that their favourite retailers' data can be stored for that specific user. This allows the users that have multiple devices or when they switch to a new device, they will still retain all the favourite retailers in that account. If no login feature is implemented for them and cache is used, it only stores the data on that device but is not transferable to other devices.

Features for retailers

Features	Details
Create/Edit/Delete product:	Allow users(retailers) to add, edit or delete products from their shop's product list which the users(consumers) will browse.
Toggle shop visibility:	If the user's(retailer) shop is closed on that day, they can toggle the visibility off so that users(consumer) will know that they are not open.
Select food availability:	Retailers will often have a wide variety of products and some may not be available. It would be troublesome for them to remove products that are not available today, only to add them again tomorrow when they are available. Therefore, we added the availability feature for products.
Edit profile:	As retailers operate their businesses, there may be changes to the location of their branch, operating hours or even the description of their shop. Thus, they will be able to make changes to those stated details in their profile anytime that they want to.
Sign up & login(Retailers):	Retailers will have to sign up for an account to display their shop's information. Retailers are the only user type that is able to make limited changes to the database. They will have to be logged in to make changes to their shop's availability, product list, or profile.

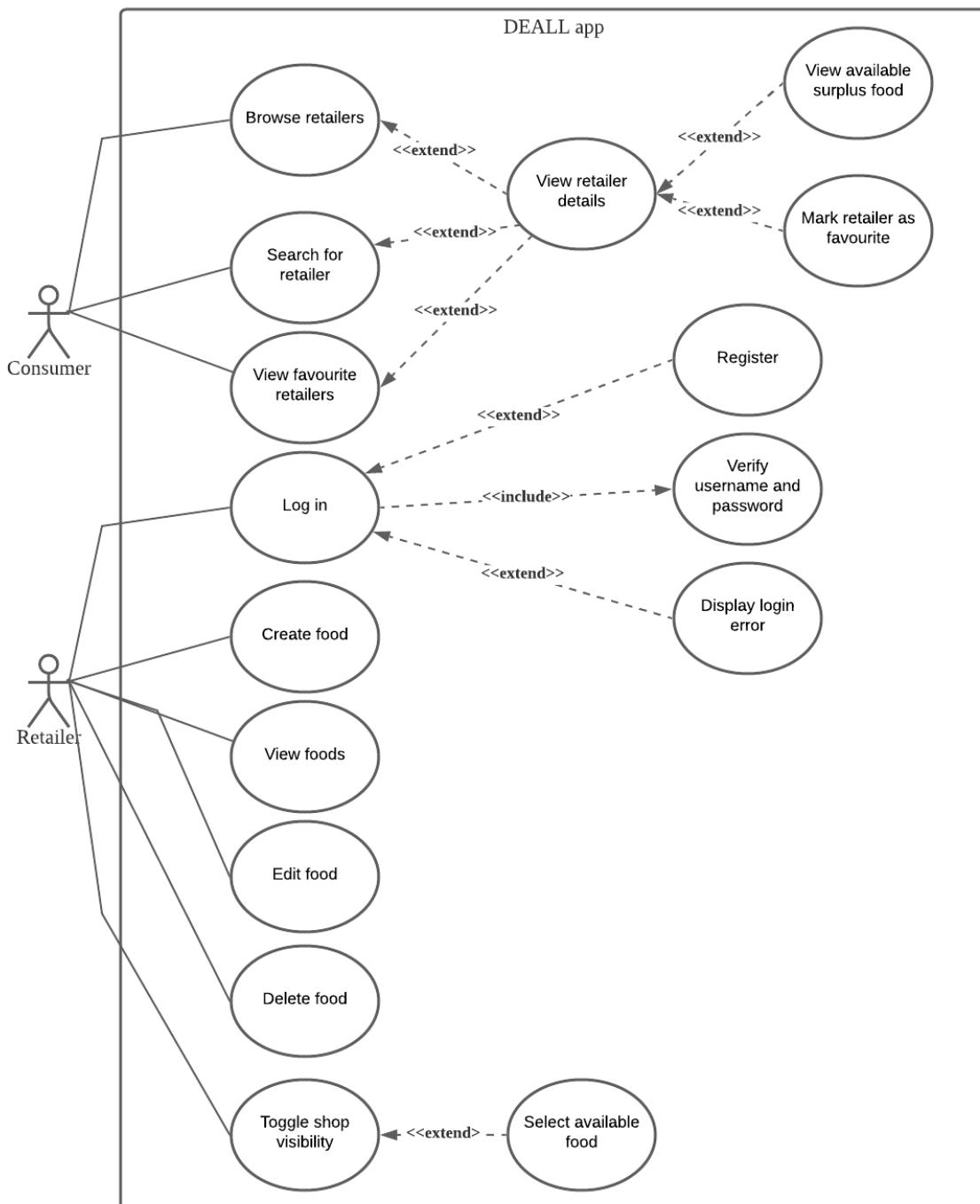
5.0 Requirements

Functional requirements

1. Retailers should be able to login to the system.
 - a. The system should validate the retailer's accounts.
2. Retailers should be able to register for a retail account.
 - a. Retailers should be able to enter their email as their username.
 - b. Retailers should be able to enter the password for the account.
 - c. Retailers be able to upload an image of their shop
 - d. Retailers should be able to enter the name of their shop.
 - e. Retailers should be able to enter their business UEN.
 - f. Retailers should be able to enter the address of their shop.
 - g. Retailers should be able to enter the operating hours of their shop.
 - h. Retailers should be able to enter the description of their shop.
3. Retailers should be able to edit their profile details
 - a. Retailers should be able to edit the name of their shop
 - b. Retailers should be able to edit the address of their shop
 - c. Retailers should be able to edit the operating hours of their shop
 - d. Retailers should be able to reupload a new image of their shop
 - e. Retailers should be able to edit the description of their shop
4. Retailers should be able to create food details on the home page.
 - a. Retailers should be able to upload an image of the food (optional).
 - b. The system should use a placeholder image as the food image if the retailer does not upload an image of the food.
 - c. Retailers should be able to enter the name of the food.
 - d. The system should prompt an error message if the name of the food is empty or more than 50 characters.
 - e. Retailers should be able to enter the description of the food (optional).
 - f. Retailers should be able to enter the usual and discounted price of the food.
 - g. The system should only allow retailers to enter the discounted price in a range from 0.01 to less than usual price.
5. Retailers should be able to edit the food details in the edit food page.
 - a. Retailers should be able to reupload a new image of the food.
 - b. The system should use a placeholder image as the food image if the retailer decides to delete the current image.
 - c. Retailers should be able to edit the name of the food.

- d. The system should prompt an error message if the name of the food is empty or more than 50 characters.
 - e. Retailers should be able to edit the description of the food.
 - f. Retailers should be able to edit the usual and discounted price of the food.
 - g. The system should only allow the retailer to enter the discounted price in a range from 0.01 to less than usual price.
6. Retailers should be able to delete the food in the edit food page
7. Retailers users should be able to toggle the visibility of their shop to the public.
8. Retailers should be able to select food that are available on that day in the home page
- a. The system should display a list of predefined foods in the home page of retail account
 - b. Retailers should be able to toggle surplus food that are available within the list of predefined foods
9. Consumers should be able to login to the system.
- a. The system should validate the inputs.
10. Consumers should be able to register for a consumer account.
- a. Consumers should be able to enter their email as their username.
 - b. Consumers should be able to enter the password for the account.
11. Consumers should be able to view the list of retailers that are available for the discount.
- a. The details such as the logo, name, description and distance of the retailer should be displayed to the consumer.
12. Consumers should be able to search for a retailer by retailer's name.
13. Consumers should be able to view the details of the retailers
- a. The details such as the logo, name, address, products, operating hours and description of the retailers should be displayed to the consumers.
14. Consumers should be able to view the list of foods that are available for discount
- a. The details such as the image, name, description and price of the food should be displayed to the consumers.
15. Consumers should be able to mark a retailer as favourite
16. Consumers should be able to view the list of favourite retailers

6.0 Use cases



Use Case Name:	Browse through retailers
----------------	---------------------------------

Description:	This use case showcases how the consumer can browse through the list of retailers that have discounts for the surplus food.
Actors:	Consumer
Preconditions:	<ul style="list-style-type: none"> - Consumer has to be logged in successfully into his/her account
Main flow:	<ol style="list-style-type: none"> 1. The system retrieves a list of retailers sorted alphabetically by name and filter by visibility 2. The system displays the list of retailers to the consumer 3. (Extension point: view Retailers details) 4. The use case ends
Postconditions:	The consumer will be able to browse through the list of retailers on the home page
Alternative flows:	<p>1a. The system unable to retrieve the list of retailers because there is no Internet connection</p> <ol style="list-style-type: none"> 1. The system display no connection page to the user 2. The use case ends
Priority:	High

Use Case Name:	Search for retailers
Description:	This use case showcases how the consumer can search for a retailer by name.
Actors:	Consumer
Preconditions:	<ul style="list-style-type: none"> - Consumer has to be logged in successfully into his/her account
Main flow :	<ol style="list-style-type: none"> 1. The consumer enters the name of the retailer 2. The consumer submits the name of the retailer to the system 3. The system validates the input 4. The system retrieves the retailer where the name contains the consumer's input 5. The system displays the retailer/s to the user 6. (Extension point: view Retailers details) 7. The use case ends

Postconditions:	The consumer will be able to find the retailer that they want.
Alternative flows:	-
Priority:	Medium

Use Case Name:	View retailers details
Description:	This use case allows the consumer to view the details of the retailers
Actors:	Consumer
Extends use cases	<ol style="list-style-type: none"> 1. Browse through retailers 2. Search for retailers 3. View favourite retailers
Preconditions:	<ul style="list-style-type: none"> - Consumer has to be logged in successfully into his/her account - At least one retailer must be created
Main flow:	<ol style="list-style-type: none"> 1. The consumer selects a retailer from the list 2. The system retrieves the details of that retailer 3. The system displays the details of the retailer to the consumer 4. (Extension point: view available surplus food) 5. (Extension point: mark retailer as favourite) 6. The use case ends
Postconditions:	The consumer will be able to view the details of the retailer such as the name, description, operating hours, and so on.
Alternative flows:	-
Priority:	High

Use Case Name:	View available surplus food
Description:	This use case allows the consumer to view the available surplus food of a retailer for that day

Actors:	Consumer
Extends use cases	1. View retailers details
Preconditions:	<ul style="list-style-type: none"> - Consumer has to be logged in successfully into his/her account - At least one retailer must be created
Main flow:	<ol style="list-style-type: none"> 1. The system retrieves the available surplus food of the retailer from the database 2. The system displays the information of the available surplus food to the consumer 3. The use case ends
Postconditions:	The consumer will be able to view the details of the available surplus food such as the image, name, original price and discounted price.
Alternative flows:	-
Priority:	Medium

Use Case Name:	Mark retailer as favourite
Description:	This use case allows the consumer to mark a retailer as their favourite retailer
Actors:	Consumer
Extends use cases	1. View retailers details
Preconditions:	<ul style="list-style-type: none"> - Consumer has to be logged in successfully into his/her account - At least one retailer must be created
Main flow:	<ol style="list-style-type: none"> 1. The user submits the request to mark a retailer as favourite 2. The system processes the request and updates the database 3. The use case ends
Postconditions:	The system will mark the retailer as the favourite retailer of that consumer
Alternative flows:	1a The user submits the request to remove a retailer as

	<p>favourite</p> <ol style="list-style-type: none"> 1. The system processes the request and updates the database 2. The use case ends
Priority	Low

Use Case Name:	View favourite retailers
Description:	This use case allows the consumer to view list of their favourite retailers
Actors:	Consumer
Preconditions:	<ul style="list-style-type: none"> - Consumer has to be logged in successfully into his/her account - At least one retailer must be created - At least one retailer has been marked as favourite
Main flow:	<ol style="list-style-type: none"> 1. The system retrieves a list of favourite retailers from the database 2. The system displays the list of favourite retailers to the consumer 3. (Extension point: view Retailers details) 4. The use case ends
Postconditions:	The consumer will be able to browse through the list of favourite retailers
Alternative flows:	-
Priority:	Low

Use Case Name:	Log in
Description:	This use case allows the retailers to login to the system to access the relevant function based on their role.
Actors:	Retailer
Preconditions:	<ul style="list-style-type: none"> - The retailer must already have a valid retail account
Main flow:	<ol style="list-style-type: none"> 1. The retailer enters registered email and password 2. The retailer submits the input to the system

	<ol style="list-style-type: none"> 3. The system validates the email and password 4. The system verifies the registered email and password 5. The system displays the retailer's homepage 6. The use case ends
Postconditions:	The system displays relevant homepage to the retailer
Alternative flows:	<p>3a. The system detects empty email and/or password</p> <ol style="list-style-type: none"> 1. The system displays error message 2. The system prompts for a valid email and password 3. Use case resumes at main flow step 1 <p>3b. The system detects invalid email format</p> <ol style="list-style-type: none"> 1. The system displays error message 2. The system prompts for a valid email 3. Use case resumes at main flow step 1 <p>4a. The system detects invalid email and/or password</p> <ol style="list-style-type: none"> 1. The system displays error message 2. The system prompts for email and password 3. Use case resumes at main flow step 1
Priority:	High

Use Case Name:	Sign up
Description:	This use case allows consumers to sign up an consumer account
Actors:	Consumer
Preconditions:	-
Main flow:	<ol style="list-style-type: none"> 1. Consumer enters email and password 2. Consumer submits the input to the system 3. The system validates the email and password 4. The system displays consumer home page 5. The use case ends
Postconditions:	The system displays relevant homepage to the consumer
Alternative flows:	<p>3a. The system detects empty email and/or password</p> <ol style="list-style-type: none"> 1. The system displays error message

	<p>2. The system prompts for a valid email and password</p> <p>3. Use case resumes at main flow step 1</p> <p>3b. The system detects invalid email format</p> <ol style="list-style-type: none"> 1. The system displays error message 2. The system prompts for a valid email 3. Use case resumes at main flow step 1 <p>4a. The system detects invalid email and/or password</p> <ol style="list-style-type: none"> 1. The system displays error message 2. The system prompts for email and password 3. Use case resumes at main flow step 1
Priority:	High

Use Case Name:	Sign up
Description:	This use case allows retailer to sign up an retailer account
Actors:	Retailer
Preconditions:	-
Main flow:	<ol style="list-style-type: none"> 1. Retailer enters email and password and shop information 2. Retailer submits the input to the system 3. The system validates the inputs 4. The system displays retailer home page 5. The use case ends
Postconditions:	The system displays relevant homepage to the retailer
Alternative flows:	<p>3a. The system detects any empty email, password, shop name, UEN, or postal code.</p> <ol style="list-style-type: none"> 1. The system displays error message 2. The system prompts for a valid inputs 3. Use case resumes at main flow step 1 <p>3b. The system detects invalid email format, password, UEN, or postal code</p> <ol style="list-style-type: none"> 1. The system displays error message 2. The system prompts for a valid input 3. Use case resumes at main flow step 1

Priority:	High
-----------	------

Use Case Name:	Create food
Description:	This use case allows retailers to create a food that will be displayed to the consumer
Actors:	Retailer
Preconditions:	<ul style="list-style-type: none"> - Retailer has to be logged in successfully into his/her retail account
Main flow:	<ol style="list-style-type: none"> 1. The retailer enters food details into the form 2. The retailer submits the food details to the system 3. The system validates the inputs in the form 4. The system stores the details of the food in the database 5. The system prompts a successful message to the retailer 6. The use case ends
Postconditions:	The system created the food in the database and the food will be displayed on the shop profile
Alternative flows:	<p>3a. The system detects empty food name, usual price and/or discounted price</p> <ol style="list-style-type: none"> 1. The system displays error message 2. The system prompts for empty food name, usual price and/or discounted price 3. Use case resumes at main flow step 1 <p>3b. The system detects the discounted price is higher than usual price</p> <ol style="list-style-type: none"> 1. The system display error message 2. The system prompts for discounted price 3. Use case resumes at main flow step 1 <p>3c. The system detects the discounted price is lower than 0.01 and more than 9999.99</p> <ol style="list-style-type: none"> 1. The system display error message 2. The system prompts for discounted price 3. Use case resumes at main flow step 1 <p>3d. The system detects the usual price is lower than 0.02 and more than 9999.99</p>

	<ol style="list-style-type: none"> 1. The system displays error message 2. The system prompts for usual price 3. Use case resumes at main flow step 1
Priority:	High

Use Case Name:	View foods
Description:	This use case allows retailers to view the details of the foods that they have created
Actors:	Retailer
Preconditions:	<ul style="list-style-type: none"> - Retailer has to be logged in successfully into his/her retail account - Food must be created
Main flow:	<ol style="list-style-type: none"> 1. The system retrieves a list of foods that the retailer created from the database 2. The system displays the list of food to the retailer 3. The use case ends
Postconditions:	The retailer will be able to browse through the list of foods that are displayed by the system
Alternative flows:	-
Priority:	High

Use Case Name:	Edit food
Description:	This use case allows retailers to edit the details of the food that they have previously created
Actors:	Retailer
Preconditions:	<ul style="list-style-type: none"> - Food must be created - Retailer has to be logged in successfully into his/her retail account - Retailer has selected a food from the list of foods
Main flow:	<ol style="list-style-type: none"> 1. The system retrieves the details of the food selected by the retailer

	<ol style="list-style-type: none"> 2. The system displays the details of the food as the default value of the text fields to the retailer 3. The retailer modifies the details of the food 4. The retailer submits the details of the food to the system 5. The system validates the input from the retailer 6. The system updates the details of the foods in the database 7. The system navigate the user back to home page 8. The use case ends
Postconditions:	The system updates the details of the food and the retailer and consumer will be able to see the updated details of the food
Alternative flows:	<p>5a. The system detects empty food name, usual price and/or discounted price</p> <ol style="list-style-type: none"> 1. The system displays error message 2. The system prompts for empty food name, usual price and/or discounted price 3. Use case resumes at main flow step 3 <p>5b. The system detects the discounted price is higher than usual price</p> <ol style="list-style-type: none"> 1. The system display error message 2. The system prompts for discounted price 3. Use case resumes at main flow step 3 <p>5c. The system detects the discounted price is lower than 0.01 and more than 9999.99</p> <ol style="list-style-type: none"> 1. The system display error message 2. The system prompts for discounted price 3. Use case resumes at main flow step 3 <p>5d. The system detects the usual price is lower than 0.02 and more than 9999.99</p> <ol style="list-style-type: none"> 1. The system display error message 2. The system prompts for usual price 3. Use case resumes at main flow step 3
Priority:	Medium

Use Case Name:	Delete food
Description:	This use case allows retailers to delete a food from their account

Actors:	Retailer
Preconditions:	<ul style="list-style-type: none"> - Food must be created - Retailer has to be logged in successfully into his/her retail account - Retailer has selected a food from the list of foods
Main flow:	<ol style="list-style-type: none"> 1. The retailer submits the request to delete a food 2. The system deletes the food from the database 3. The system navigate the user back to home page 4. The use case ends
Postconditions:	The system deletes the food from the list and will not be recoverable anymore
Alternative flows:	-

Use Case Name:	Toggle shop visibility
Description:	This use case allows the retailer to toggle the visibility of their shop to the public
Actors:	Retailer
Preconditions:	<ul style="list-style-type: none"> - Retailer has to be logged in successfully into his/her retail account
Main flow:	<ol style="list-style-type: none"> 1. The retailer toggles the visibility of their shop on 2. The system processes the request and displays the shop in the home page of the public 3. (extension point: select available food) 4. The use case ends
Postconditions:	The system toggles the visibility of the shop to the public
Alternative flows:	1a. The retailer toggles the visibility of their shop off 1. Use case resumes at main flow step 2
Priority:	High

Use Case Name:	Select available food
Description:	This use case allows the retailer to select the surplus food of the day. The selected food will be displayed to the consumer.

Actors:	Retailer
Extend use cases:	1. Toggle shop visibility
Preconditions:	<ul style="list-style-type: none"> - At least a food must be created - Retailer has to be logged in successfully into his/her retail account
Main flow:	<ol style="list-style-type: none"> 1. The system retrieves the list of foods from the database 2. The system displays the details of the foods including the previous visibility state of each food 3. The retailer toggles the visibility of each food on/off 4. The system processes the requests and updates the visibility of the foods in the database 5. The use case ends
Postconditions:	The consumer will be able to see the selected surplus food of the retailers
Alternative flows:	-
Priority:	High

7.0 System Development

Technologies

Language and framework

Flutter SDK and Dart programming language work well together and we have chosen them to develop our application. They allow fast development of prototypes and minimal viable product (MVP) by providing ready-made widgets for fast user interface(UI) coding. Other than that, there are tons of open source packages available for us to facilitate the development of the app. Therefore, they are great for our project because we focus heavily on user-centered design. We need to be able to build each feature in a short timeframe so that we can gather feedback from the users and make changes based on the feedback.

Lastly, they allow building of cross-platform applications. Even though we are building an Android app for this assignment, we might want to build our application on IOS as well in the future.

Minimum device requirement

- Device running android operating system version 5.1 and above
- Minimum of 4gb RAM is recommended

Authentication

Firebase authentication is chosen as our authentication service because:

- It has a robust and popular library made for flutter application development by the Google firebase team.
- It leverages industry standards which is OAuth 2 but provides simpler implementations than the OAuth 2 library out there.
- It supports email and password authentication which is what we are looking for in our application

Database

Firebase cloud firestore

Cloud firestore is a cloud hosted, NoSql database mainly used to store common data types such as numbers, strings, lists, timestamps, and maps in documents. The reasons why we decided to use cloud firestore as our main database is:

- It provides a serverless solution. Cloud firestore allows us to quickly develop prototypes and iterations without needing to set up an intermediary server. This is a huge benefit for us because we focus heavily on user centered design where iterations are crucial.

- It offers free quotas for storing and querying data. The quota that cloud firestore offers is sufficient for our project because our active user base is limited as we haven't published and promoted our application. Even if we exceed the limit, cloud firestore is so scalable that we just need to pay on demand. The image below shows the quota that cloud firestore offers for free. (Usage and limits | Firebase Documentation, n.d.)

Free tier	Quota
Stored data	1 GiB
Document reads	50,000 per day
Document writes	20,000 per day
Document deletes	20,000 per day
Network egress	10 GiB per month

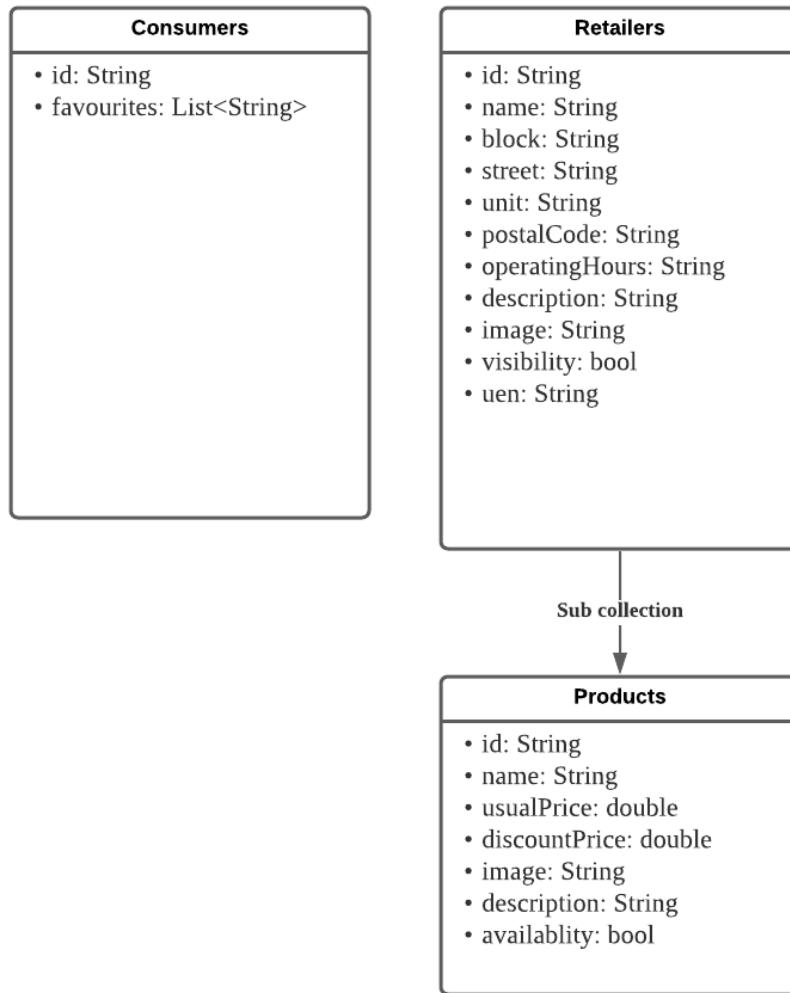
- It provides both real time and offline support. This further facilitates the development of our application as we do not have to implement another local storage solution for local caching
- It works well with services such as cloud storage and firebase authentication. For example, we can write security rules on cloud firestore such that the user have to be authenticated through firebase authentication before they can access the data stored in cloud firestore

Firebase cloud storage

We chose cloud storage as the object storage service to store and retrieve images such as profile pictures and product images. Some of the reasons are:

- It can store user generated content directly without needing to do manual conversion which simplifies the software development process
- It offers free quotas. All of the firebase service provides free quotas which is good for school projects.
- It works well with other firebase services.
- We can set the security rules based on authenticity of the user which are managed by firebase authentication

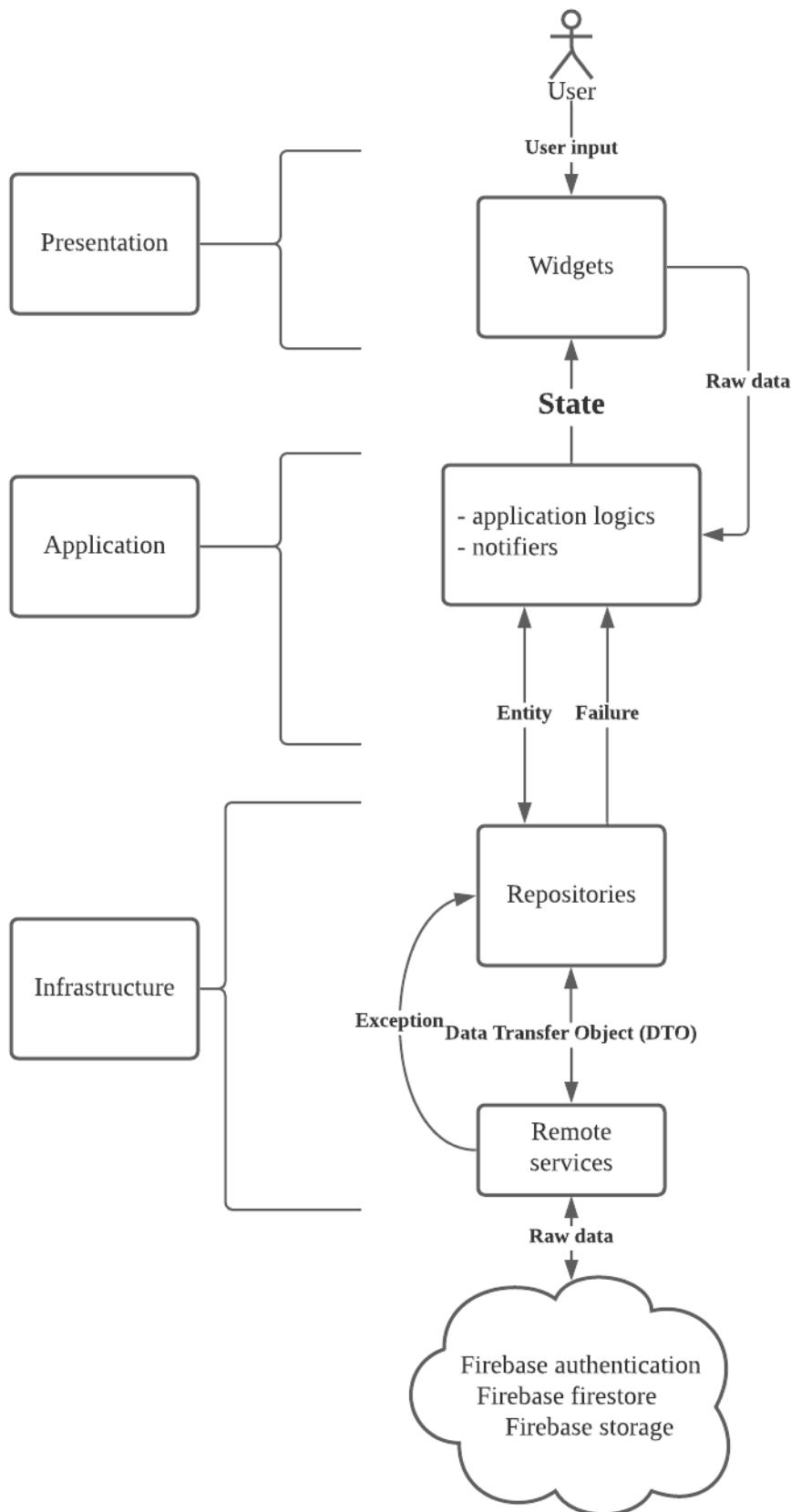
Database Structure



(database structure)

The image above shows the entities used in our project. In firebase firestore, each entity is called collection. We have three collections in the project, they are, consumers, retailers, and products. Products is the sub-collection of the retailers which means it is dependent on retailers. A retailer can have zero or more products but a product belongs to only one retailer. Each record is called a document in this context and it holds attributes about the entity.

Software Architecture



Presentation layer

Presentation layer contains widgets which make up the user interface to the user and the user can interact with some of the widgets like buttons, text fields, etc. Widgets communicate with the notifier in the application layer by sending raw data such as strings, integers and so on into the function held in the notifier. The notifier will respond with states which will change the user interface. For example, the application may display a loading indicator to the user when it is in loading state.

Application layer

Entities are pure business classes that do not rely on flutter framework so that they can be reused in any dart program.

Failures are union classes that contain all possible failures that could happen to the application. For example, DatabaseFailure union class may contain subclasses such as NoConnection, Unauthenticated, Unexpected, and so on.

Notifiers communicate with the presentation layer by returning proper state based on the value returned by the repository, the value returned by the repository could be either entity or failure union class. Other than that, it will communicate with repositories by passing in data such as entities into the repository.

Infrastructure layer

Data transfer object (DTO) is a copy of entity class, but is not pure as it may contain any kind of data such as raw data from databases. DTO also contains extra functionality such as converting to/from JSON and to/from entity. DTO is mainly used in communication between repositories and remote services.

Repositories communicate with notifiers by responding back either entities or failures to the notifiers. Repositories also communicate with remote services by passing data such as DTO as argument. Other than that, repositories are responsible for handling all possible exceptions coming from remote services by converting the exception into failures object which will then be returned back to the notifiers so that the user will be displayed with the error message.

Remote services communicate with API and databases and they will respond with data such as DTO back to the repository. Remote services will not handle the exception thrown by API and databases, instead, it will either rethrow a custom exception or the default one back to the repository to handle.

Example of data flow

One of the important features in our app is retrieving the information of the retailer from the database when the user signs in. This feature will be used to explain the flow of the data from user input to the database and vice versa.

```
@override
Widget build(BuildContext context) {
    return Scaffold(
        appBar: AppBar(
            title: const Text('DEALL'),
        ), // AppBar
        drawer: const RetailerDrawer(),
        body: ref.watch(retailerNotifierProvider).when(
            initial: () {
                // returns empty screen
            },
            loading: () {
                // returns loading indicator
            },
            loaded: (retailer, hasConnection) {
                // returns the UI of the home page if loaded
            },
            failure: (retailerFailure) {
                //returns error screen if fail to retrive data
            },
        ),
    ); // Scaffold
}
```

(code snippet of retailer home page in presentation layer)

The build function above is used to build the user interface of the retailer home page. This function builds different types of UI depending on the state. In this case, the possible state could be initial, loading, loaded or failure state. The state of the UI will be controlled by the retailer notifier.

```

Future<void> getRetailer() async {
    state = const RetailerNotifierState.loading();
    final failureOrRetailer = await _retailerRepository.getRetailer();
    failureOrRetailer.fold(
        (f) => state = RetailerNotifierState.failure(f),
        (retailer) {
            state = RetailerNotifierState.loaded(
                retailer,
                hasConnection: true,
            );
        },
    );
}

```

(code snippet of getRetailer function in retailer notifier)

When the user signs in to the application, the application will trigger the getRetailer() function in the notifier shown above before building the home page. Retrieving the retailer's details from the database is asynchronous, therefore, the function will assign the state to be a loading state before calling the asynchronous function. This results in the user seeing a few seconds or milliseconds of loading indicator in the UI while waiting for the asynchronous function to complete.

The getRetailer() function from the repository will return either a retailer entity or failure object which will then be handled by the notifier by assigning either a loaded state or failure state depending on the return value. These states will be used in the presentation layer to determine the proper UI that should be displayed to the user.

```

Future<Either<RetailerFailure, Retailer>> getRetailer() async {
    if(! await _internetConnectionChecker.hasConnection) {
        return left(const RetailerFailure.noConnection());
    }

    try {
        final retailerDTO = await _retailerRemoteService.getRetailer();
        return right(retailerDTO.toDomain());
    } on FirebaseAuthException catch (e) {
        return left(RetailerFailure.authentication('${e.code}: ${e.message}'));
    } on FirebaseException catch (e) {
        if (e.code == 'not-found') {
            return left(const RetailerFailure.notFound());
        } else {
            return left(RetailerFailure.unexpected('${e.code}: ${e.message}'));
        }
    }
}

```

(code snippet of getRetailer function in retailer repository)

The return type of getRetailer() function in the repository shown above is either RetailerFailure or Retailer object. This function will call the getRetailer() function from the remote service. Then, it will catch all possible exceptions thrown by returning different types of RetailerFailure back to the notifier to handle. If no exception is thrown, the function will get back the Retailer Data Transfer Object(DTO) from the remote service which will then be converted into a Retailer object.

```
Future<RetailerDTO> getRetailer() async {
    final userId = _firebaseAuth.currentUser!.uid;
    final documentSnapshot =
        await _firebaseFirestore.collection('retailers').doc(userId).get();
    final retailerJson = documentSnapshot.data()!;

    retailerJson['id'] = documentSnapshot.id;

    return RetailerDTO.fromJson(retailerJson);
}
```

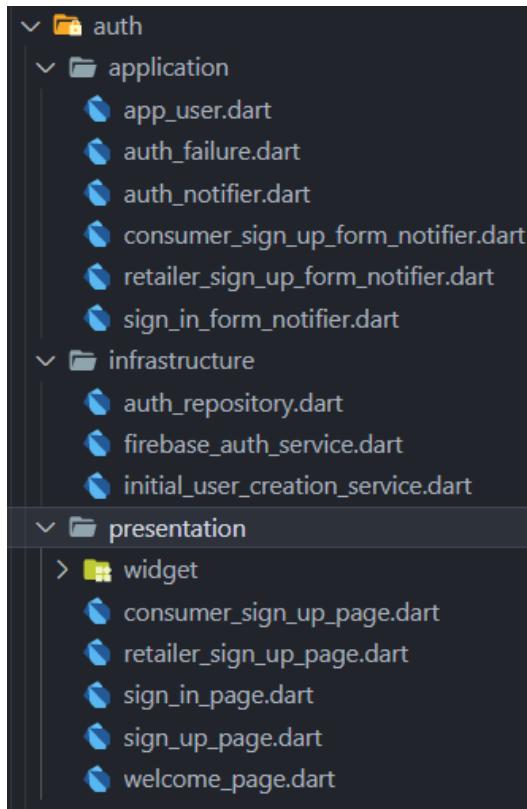
(code snippet of getRetailer function in retailer remote service)

The getRetailer() function in the remote service shown above communicates with Firebase Firestore to retrieve the raw data of JSON type from the database. If successful, this function will convert the raw data into RetailerDTO which will then be returned to the caller in the repository.

Summary

To summarise, by following this software architecture in our software development, we achieved separation of concern (SOC). Presentation code, business logic, and logic for communicating with databases or APIs are separated in different layers. Other than that, we achieved a clean and maintainable codebase as the structure of the code is clear and easy to understand. This can greatly help us in the future when we try to add new features.

File structure



(screenshot of file structure of authentication feature)

The image above shows the file structure of a feature. Our application consists of multiple features and authentication is one of the features that will be used to explain the file structure. Application folder consists of entities, failures and notifiers. Infrastructure layer consists of repositories and remote services. Presentation layer consists of pages displayed to the user which are made up of widgets.

Error and exception handling

```
Future<Either<FirestoreFailures, Unit>> updateFavouriteRetailerList(  
    List<Retailer> retailerList) async {  
    if (!await _internetConnectionChecker.hasConnection) {  
        return const Left(FirestoreFailures.noConnection());  
    }  
    final retailerIdList = retailerList.map((retailer) => retailer.id).toList();  
  
    try {  
        await _favouriteListRemoteService.updateFavouriteRetailerList(  
            retailerIdList: retailerIdList,  
            userId: _authRepository.getUserId(),  
        );  
        return right(unit);  
    } on FirebaseException catch (e) {  
        if (e.code == 'not-found') {  
            return left(const FirestoreFailures.objectNotFound());  
        } else {  
            return left(const FirestoreFailures.unknown());  
        }  
    }  
}
```

(updateFavouriteRetailerList function in repository)

Most of the exceptions and errors are handled in repositories because they call functions from the remote services. Remote services may throw different kinds of exceptions and errors as they are in charge of communicating with third party resources such as databases.

From the image above, the function catches all possible exceptions and errors by returning a failure object as the result. The failure object will be used by notifiers to determine the state of the application.

Official documentation

```
/// Updates retailer data with the provided [retailer] data
///
/// If device has no internet connection, [RetailerFailure.noConnection()] will be returned.
///
/// If user is not authenticated, [FirebaseAuthException] will be thrown and
/// [RetailerFailure.authentication(errorMessage)] will be returned.
///
/// If no existing retailer with the provided id is found, [FirebaseException] will be thrown
/// and [RetailerFailure.notFound()] will be returned.
///
/// If other errors are found, [FirebaseException] will be thrown and
/// [RetailerFailure.unexpected(errorMessage)] will be returned.
Future<Either<RetailerFailure, Unit>> updateRetailer(
    Retailer retailer) async {
  if (!await _internetConnectionChecker.hasConnection) {
    return left(const RetailerFailure.noConnection());
  }
  try {
    await _retailerRemoteService
      .updateRetailer(RetailerDTO.fromDomain(retailer));
    return right(unit);
  } on FirebaseAuthException catch (e) {
    return left(RetailerFailure.authentication('${e.code}: ${e.message}'));
  } on FirebaseException catch (e) {
    if (e.code == 'not-found') {
      return left(const RetailerFailure.notFound());
    } else {
      return left(RetailerFailure.unexpected('${e.code}: ${e.message}'));
    }
  }
}
```

(*updateRetailer function located in retailer repository*)

Other than using unit tests as part of our documentation, we have included official comments in repositories so that we can refer to them when developing other layers of the application. The image above demonstrates an example of official documentation written in the application.

External resources

```
auto_route: ^2.2.0
cached_network_image: ^3.1.0
cloud_firestore: ^2.4.0
connectivity: ^3.0.6
cupertino_icons: ^1.0.2
dartz: ^0.10.0-nullsafety.1
firebase_auth: ^3.0.1
firebase_core: ^1.4.0
firebase_storage: ^10.0.1
flutter:
  sdk: flutter
flutter_native_splash: ^1.2.0
flutter_riverpod: ^1.0.0-dev.5
flutter_screenutil: ^5.0.0+2
freezed_annotation: ^0.14.1+3
google_fonts: ^2.1.0
image_picker: ^0.8.2
internet_connection_checker: ^0.0.1+2
json_annotation: ^4.0.1
path: ^1.8.0
rxdart: ^0.27.1
```

(Third party packages imported in DEALL)

We have used some third party packages in our application to improve our software development. Some of the packages worth mention are:

- image_picker
 - This package allows users to pick an image from the local gallery.
- firebase_auth
 - This package allows us to integrate firebase authentication into the application to authenticate the user

- firebase_storage
 - This package allows us to integrate firebase storage so that we can stores and retrieves images from the cloud easily
- cloud_firestore
 - This package allows us to integrate firebase firestore so that we can store information into the database.
- internet_connection_checker & connectivity
 - These packages allow us to check whether the user's device is connected to the Internet before performing actions that require Internet connection.
- auto_route
 - This package simplifies the routing in the app by generating boilerplate code for us

Black box testing

Black box testing for each feature is done after each sprint as well as after the app is fully developed. They are conducted by us to make sure the features work as intended in the user's perspective. Decision table testing and test cases are done for all features. However, equivalence partitioning and boundary value analysis are done on selected applicable features

Techniques:

Equivalence partitioning → boundary value analysis → decision table testing → test case table.

Browse through retailers (consumer)

Decision Table			
Conditions	Case 1	Case 2	Case 3
Consumer has logged in to the system?	T	F	T
At least one retailer with visibility on?	T	-	F
Action: browse through retailer list?	T	F	F

Test Table						
Test No.	Test Cases	Preconditions	Steps	Expected Results	Actual Results	Pass/Fail

1	No retailer turns on visibility of their shop	1. Consumer must be logged in	1. Consumer logged in successfully	No retailer will be shown in the home page	No retailer will be shown in the home page	Pass
2	At least one retailer turns on visibility of their shop	1. Consumer must be logged in	1. Consumer logged in successfully	All available retailers will be shown in the home page	All available retailers will be shown in the home page	Pass

Search for retailers (consumer)

Decision Table			
Conditions	Case 1	Case 2	Case 3
Consumer has logged in to the system?	T	F	T
At least one retailer name contains the user input?	T	-	F
Action: able to get a list of retailer/s	T	F	F

Test Table						
Test No.	Test Cases	Preconditions	Steps	Expected Results	Actual Results	Pass/Fail
1	User enters name of existing retailer	1. Consumer must be logged in	1. Consumer logged in successfully 2. Enter 'test'	The system displays a list of retailer where the name contains the user input	As expected	Pass

			retailer' into search bar			
2	User enters name of non-existing retailer	1. Consumer must be logged in	1. Consumer logged in successfully 2. Enter '!@#\$%^&*' into search bar	The system displays an empty page with no result found message to the user	As expected	Pass

View retailer details (consumer)

Decision Table			
Conditions	Case 1	Case 2	Case 3
Consumer has logged in to the system?	T	F	T
At least one retailer has been created?	T	-	F
Action: Consumer able to view retailer's details?	T	F	F

Test Table						
Test No.	Test Cases	Preconditions	Steps	Expected Results	Actual Results	Pass/Fail
1	Consumer should be able to view retailer's details	1. Consumer must be logged in 2. At least 1 retailer turns the shop visibility on	1. Consumer logged in successfully 2. Click on a random retailer list tile in the	The system displays the detail of the selected retailer	As expected	Pass

			home page			
--	--	--	-----------	--	--	--

View available surplus food (consumer)

Decision Table		
Conditions	Case 1	Case 2
Consumer has logged in to the system?	T	T
At least one retailer has been created?	T	T
At least one food is available?	T	F
Action: consumer able to see the list of surplus foods?	T	F

Test Table						
Test No.	Test Cases	Preconditions	Steps	Expected Results	Actual Results	Pass/Fail
1	Select retailer with no surplus food available	1. Consumer must be logged in 2. At least 1 retailer turns the shop visibility on	1. Consumer logged in successfully 2. Click on a retailer where no surplus food is available	The system display empty list with no food available message	As expected	Pass
2	Select retailer with at least 1 surplus available	1. Consumer must be logged in 2. At least 1 retailer turns the shop visibility on	1. Consumer logged in successfully 2. Click on a retailer where at least one	The system display a list of available surplus food to the user	As expected	Pass

			surplus food is available			
--	--	--	------------------------------	--	--	--

Mark retailer as favourite (consumer)

Decision Table			
Conditions	Case 1	Case 2	Case 3
Consumer has logged in to the system?	T	F	T
At least one retailer has been created?	T	-	F
Action: mark retailer as favourite?	T	F	F

Test Table						
Test No.	Test Cases	Preconditions	Steps	Expected Results	Actual Results	Pass/Fail
1	Mark retailer as favourite	1. Consumer must be logged in 2. At least 1 retailer is created	1. Consumer logged in successfully 2. Click on any retailer created 3. Click on favourite icon	The system stores the selected retailer into the consumer's favourite list. The favourite button filled with colour	As expected	Pass
2	Remove retailer as favourite	1. Consumer must be logged in 2. At least 1 retailer is marked as favourite	1. Consumer logged in successfully 2. Navigate to favourite retailer list	The system remove the selected retailer from the favourite list The	As expected	Pass

			3. Click on any favourite retailer 4. Click on favourite icon	favourite button become outlined shape		
--	--	--	--	--	--	--

View favourite retailers (consumer)

Decision Table		
Conditions	Case 1	Case 2
Consumer has logged in to the system?	T	T
At least one retailer has been created?	T	T
At least one retailer is marked as favourite?	T	F
Action: able to view favourite retailers?	T	F

Test Table						
Test No.	Test Cases	Preconditions	Steps	Expected Results	Actual Results	Pass/Fail
1	View favourite retailer with at least one retailer marked as favourite	1. Consumer must be logged in 2. At least 1 retailer is created 3. At least 1 retailer is marked as favourite	1. Consumer logged in successfully 2. Navigate to favourite retailer list	The system should display a list of retailers that has been marked as favourite	As expected	Pass
2	View	1. Consumer	1. Consumer	The system	As	Pass

	favourite retailer with no retailer marked as favourite	must be logged in 2. At least 1 retailer is created 3. No retailer has been marked as favourite	logged in successfully 2. Navigate to favourite retailer lis	should display an empty page with no result found message to the user	expected	
--	---	---	---	---	----------	--

Login (consumer & retailer)

Decision Table			
Conditions	Case 1	Case 2	Case 3
valid email?	T	F	T
Valid password?	T	-	F
Action: home page is displayed	T	F	F

Test Table						
Test No.	Test Cases	Preconditions	Steps	Expected Results	Actual Results	Pass/Fail
1	The user enters the following email and password combination . Note: I means invalid, V means valid	1. User must have a registered account	1. Enter email and password combination 2. Click on login button	Note: E means the system will show error message to the user, S means the user is authenticated and will be navigated back to the home page		
1.1	V, V			S	S	Pass

1.2	F, T			E	E	Pass
1.3	T, F			E	E	Pass

Sign up (consumer)

Decision Table			
Conditions	Case 1	Case 2	Case 3
valid email?	T	F	T
Valid password?	T	-	F
Action: home page is displayed	T	F	F

Test Table						
Test No.	Test Cases	Preconditions	Steps	Expected Results	Actual Results	Pass/Fail
1	The user enters the following email and password combination . Note: I means invalid, V means valid	-	1. Enter email and password combination 2. Click on login button	Note: E means the system will show error message to the user, S means the user is authenticated and will be navigated back to the home page		
1.1	V, V			S	S	Pass
1.2	F, T			E	E	Pass
1.3	T, F			E	E	Pass

Sign up (retailer)

Equivalence Partitioning			
Test Case	Invalid	Valid	Invalid
Number of character for the shop name	0	$1 \leq x \leq 50$	> 50
Number of character for the UEN	0	> 0	-
Number of digits for postal code	< 6	6	> 6

Boundary Value			
Test Case	Invalid	Valid	Invalid
Number of character for the shop name	0	1, 50	51
Number of character for the UEN	0	> 0	-
Price range of discounted price	0.00	0.01 ,Usual price - 0.01	Usual price

Decision Table						
Conditions	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
Valid email?	T	F	T	T	T	T
Valid password?	T	-	F	T	T	T
Valid shop name?	T	-	-	F	T	T
Valid UEN?	T	-	-	-	F	T
Valid Postal?	T	-	-	-	-	F

Action: sign up successfully and navigate to the home page?	T	F
---	---	---

Test Table						
Test No.	Test Cases	Preconditions	Steps	Expected Results	Actual Results	Pass/Fail
1	The user enters the following email, password, shop name, UEN, and postal code combination . Note: I means invalid, V means valid	-	1. Enter combination 2. Click on sign up button	Note: E means the system will show error message to the user, S means the user account is successfully created and will be navigated back to the home page		
1.1	V, V, V, V, V			S	S	Pass
1.2	F, T, T, T, T			E	E	Pass
1.3	T, F, T, T, T			E	E	Pass
1.4	T, T, F, T, T			E	E	Pass
1.5	T, T, T, F, T			E	E	Pass
1.6	T, T, T, T, F			E	E	Pass

Create food (retailer)

Equivalence Partitioning

Test Case	Invalid	Valid	Invalid
Number of character for the name of the food	0	$1 \leq x \leq 50$	> 50
Price range of usual price	< 0.02	$0.02 \leq x \leq 9999.99$	> 9999.99
Price range of discounted price	< 0.01	$0.01 \leq x <$ usual price	\geq usual price

Boundary Value			
Test Case	Invalid	Valid	Invalid
Number of character for the name of the food	0	1, 50	51
Price range of usual price	0.01	0.02,9999.99	10000.00
Price range of discounted price	0.00	0.01 ,Usual price - 0.01	Usual price

Decision Table				
Conditions	Case 1	Case 2	Case 3	Case 5
Food name valid?	T	F	T	T
Price range of the usual price valid?	T	-	F	T
Price range of the discounted price valid?	T	-	-	F
Action: create food?	T	F		

Test Table						
Test No.	Test cases	precondition	Steps	Expected results	Actual results	Pass/Fail
1	User input the following name, usual price and discounted price combination. Note: I means invalid, V means valid	1. Must be logged in as a Retailer	1. Retailer logged in successfully 2. Click on add food icon in the home page 3. Enter the fields based on the combination 4. Press 'add food'	Note: E means the system will show error message to the user, S means the system successfully created the food in the database and the user will be navigated back to the home page		
1.1	V, V, V			S	S	Pass
1.2	I, V, V			E	E	Pass
1.3	V, I, V			E	E	Pass
1.4	V, V, I			E	E	Pass

View foods (retailer)

Decision Table		
Conditions	Case 1	Case 2
Retailer has logged in to the system?	T	T
At least one food has been created?	T	F
Action: able to view food list?	T	F

Test Table						
Test No.	Test cases	precondition	Steps	Expected results	Actual results	Pass/Fail
1	View the list of food with at least one food created	1. Must be logged in as a retailer 2. At least one food has been created	1. Retailer log in successfully	The system display a list of food in the home page	As expected	Pass
2	View the list of food with no food created	1. Must be logged in as a retailer 2. No food has been created	1. Retailer log in successfully	The system display a no food message to the user	As expected	Pass

Edit food (retailer)

Equivalence Partitioning			
Test Case	Invalid	Valid	Invalid
Number of character for the name of the food	0	$1 \leq x \leq 50$	> 50
Price range of usual price	< 0.02	$0.02 \leq x \leq 9999.99$	> 9999.99
Price range of discounted price	< 0.01	$0.01 \leq x < \text{usual price}$	$\geq \text{usual price}$

Boundary Value			
Test Case	Invalid	Valid	Invalid
Number of character for the name of the food	0	1, 50	51
Price range of usual price	0.01	0.02,9999.99	10000.00
Price range of discounted price	0.00	0.01 ,Usual price - 0.01	Usual price

Decision Table				
Conditions	Case 1	Case 2	Case 3	Case 5
Food name valid?	T	F	T	T
Price range of the usual price valid?	T	-	F	T
Price range of the discounted price valid?	T	-	-	F
Action: create food?	T		F	

Test Table						
Test No.	Test cases	precondition	Steps	Expected results	Actual results	Pass/Fail
1	User input the following name, usual price and discounted price combination. Note: I means invalid, V means valid	1. Must be logged in as a Retailer 2. At least a food must be	1. Retailer logged in successfully 2. Click on edit button of a food in	Note: E means the system will show error message to the user, S means		

		created	the home page 3. Enter the fields based on the combination 4. Press 'update' button	the system successfully updated the food in the database and the user will be navigated back to the home page		
1.1	V, V, V			S	S	Pass
1.2	I, V, V			E	E	Pass
1.3	V, I, V			E	E	Pass
1.4	V, V, I			E	E	Pass

Delete food (retailer)

Decision Table		
Conditions	Case 1	Case 2
Retailer has logged in to the system?	T	T
At least a food has been created?	T	F
Action: delete food?	T	F

Test Table						
Test No.	Test cases	precondition	Steps	Expected results	Actual results	Pass/Fail
1	Delete food from food list	1. Retailer has logged in to the system	1. Retailer logged in successfully	The system deleted the food in the database	As expected	Pass

		2. At least one food must be created	2. Click on edit button of the selected food 3. Click on delete button	and navigate back to home page		
--	--	--------------------------------------	---	--------------------------------	--	--

Toggle shop visibility (retailer)

Decision Table		
Conditions	Case 1	Case 2
Retailer has logged in to the system?	T	F
Action: toggle shop visibility?	T	F

Test Table						
Test No.	Test cases	precondition	Steps	Expected results	Actual results	Pass/Fail
1	Toggle shop visibility	1. Retailer has logged in to the system	1. Retailer logged in successfully 2. Click button to toggle visibility of the shop 3. Log into consumer account 4. Check if the retailer	The system update the database The user should be able to see the retailer appear in the list after switching to consumer account	As expected	Pass

			appear in the home page			
--	--	--	-------------------------	--	--	--

Select available food (retailer)

Decision Table		
Conditions	Case 1	Case 2
Retailer has logged in to the system?	T	T
At least one food has been created?	T	F
Action: toggle food availability	T	F

Test Table						
Test No.	Test cases	precondition	Steps	Expected results	Actual results	Pass/Fail
1	Toggle food availability	1. Retailer has logged in to the system 2. At least one food must be created	1. Retailer logged in successfully 2. Click button to toggle the availability of the selected product 3. Log into consumer account 4. Search for the retailer	The system update the database The user should be able to see the food appear in the retailer detail page after switching to consumer account	As expected	Pass

			5. Click on the retailer 6. Check if the food appear in retailer detail page			
--	--	--	--	--	--	--

White box testing

```

v ✓ validatePassword 4/4 passed, 44ms
  ✓ should return ValueFailure.invalidPassword if the input is empty 32ms
  ✓ should return ValueFailure.invalidPassword if the input length < 8 4ms
  ✓ should return ValueFailure.invalidPassword if the input length >= 8 but does not contain numeric 4ms
  ✓ should return Unit if the input contain at least 1 numeric char and length of 8 or above 4ms

```

(unit test log for validating password fields)

Unit testing is the main technique that we use for white box testing. They are written after the implementation of each feature so that we are more confident with our logics. Unit tests are written such that all possible paths in a function are tested and work as intended. The image above shows the unit test log for validatePassword() function. Various test inputs are used to test the robustness of the function.

The images below are the unit test logs for every sprint:

```
✓  ✓  auth_notifier_test.dart 3/3 passed, 138ms
  >  ✓  checkIfUserIsAuthenticated 2/2 passed, 112ms
  >  ✓  signOut 1/1 passed, 26ms
✓  ✓  auth_repository_test.dart 13/13 passed, 154ms
  >  ✓  getFirebaseUser 3/3 passed, 61ms
  >  ✓  signIn 4/4 passed, 54ms
  >  ✓  consumerSignUp 4/4 passed, 29ms
  >  ✓  signOut 2/2 passed, 10ms
✓  ✓  retailer_list_repository_test.dart 5/5 passed, 11...
  >  ✓  getRetailerList 5/5 passed, 113ms
✓  ✓  value_validator_test.dart 11/11 passed, 80ms
  >  ✓  validateEmail 7/7 passed, 66ms
  >  ✓  validatePassword 4/4 passed, 14ms
✓  ✓  retailer_dto_test.dart 1/1 passed, 33ms
  >  ✓  toDomain 1/1 passed, 33ms
```

(sprint 1: 66 tests)

```
✓ auth_notifier_test.dart 3/3 passed, 144ms
  > checkIfUserIsAuthenticated 2/2 passed, 120ms
  > signOut 1/1 passed, 24ms
✓ auth_repository_test.dart 18/18 passed, 296ms
  > getFirebaseUser 3/3 passed, 98ms
  > signIn 4/4 passed, 67ms
  > consumerSignUp 4/4 passed, 30ms
  > retailerSignUp 5/5 passed, 59ms
  > signOut 2/2 passed, 42ms
✓ retailer_list_repository_test.dart 5/5 passed, 12...
  > getRetailerList 5/5 passed, 121ms
✓ product_list_notifier_test.dart 2/2 passed, 141ms
  > getProductStream 2/2 passed, 141ms
✓ value_validator_test.dart 27/27 passed, 192ms
  > validateEmail 7/7 passed, 82ms
  > validatePassword 4/4 passed, 19ms
  > validatePostal 2/2 passed, 9ms
  > validateNotEmpty 2/2 passed, 8ms
  > validateUsualPrice 5/5 passed, 31ms
  > validateDiscountedPrice 7/7 passed, 43ms
✓ retailer_dto_test.dart 1/1 passed, 50ms
  > toDomain 1/1 passed, 50ms
✓ retailer_repository_test.dart 10/10 passed, 116ms
  > getRetailer 5/5 passed, 83ms
  > updateRetailer 5/5 passed, 33ms
✓ product_repository_test.dart 9/9 passed, 104ms
  > getProductList 4/4 passed, 80ms
  > generateNewProductId 1/1 passed, 4ms
  > addProduct 4/4 passed, 20ms
✓ add_product_form_notifier_test.dart 10/10 passed
  > prodNameChanged 1/1 passed, 53ms
  > prodUsualPriceChanged 1/1 passed, 5ms
  > prodDiscountedPriceChanged 1/1 passed, 5ms
  > prodImageChanged 1/1 passed, 5ms
  > prodDescriptionChanged 1/1 passed, 4ms
  > prodAvailabilityChanged 1/1 passed, 4ms
  > pickImage 2/2 passed, 27ms
  > deleteImage 2/2 passed, 10ms
```

(sprint 2: 85 tests)

		✓ ✓ retailer_edit_profile_form_notifier_test.dart 14/1
	>	✓ initialiseRetailer 1/1 passed, 69ms
	>	✓ nameChanged 1/1 passed, 11ms
	>	✓ blockChanged 1/1 passed, 5ms
	>	✓ streetChanged 1/1 passed, 8ms
	>	✓ unitChanged 1/1 passed, 6ms
	>	✓ postalCodeChanged 1/1 passed, 6ms
	>	✓ operatingHoursChanged 1/1 passed, 5ms
	>	✓ descriptionChanged 1/1 passed, 6ms
	>	✓ imageChanged 1/1 passed, 10ms
	>	✓ imageStringChanged 1/1 passed, 6ms
	>	✓ pickImage 2/2 passed, 42ms
	>	✓ deleteImage 2/2 passed, 13ms
✓	✓	retailer_repository_test.dart 10/10 passed, 222ms
	>	✓ getRetailer 5/5 passed, 183ms
	>	✓ updateRetailer 5/5 passed, 39ms
✓	✓	product_repository_test.dart 17/17 passed, 169...
	>	✓ getProductList 4/4 passed, 80ms
	>	✓ generateNewProductId 1/1 passed, 5ms
	>	✓ addProduct 4/4 passed, 23ms
	>	✓ updateProduct 4/4 passed, 26ms
	>	✓ updateProductList 4/4 passed, 35ms
✓	✓	add_product_form_notifier_test.dart 10/10 pas...
	>	✓ prodNameChanged 1/1 passed, 62ms
	>	✓ prodUsualPriceChanged 1/1 passed, 4ms
	>	✓ prodDiscountedPriceChanged 1/1 passed, 4ms
	>	✓ prodImageChanged 1/1 passed, 6ms
	>	✓ prodDescriptionChanged 1/1 passed, 4ms
	>	✓ prodAvailabilityChanged 1/1 passed, 5ms
✓	✓	retailer_dto_test.dart 1/1 passed, 68ms
	>	✓ toDomain 1/1 passed, 68ms

(sprint 3: 107 tests)

		✓ retailer_edit_profile_form_notifier_test.dart 14/14 passed, 146ms
	>	✓ initialiseRetailer 1/1 passed, 79ms
	>	✓ nameChanged 1/1 passed, 11ms
	>	✓ blockChanged 1/1 passed, 5ms
	>	✓ streetChanged 1/1 passed, 7ms
	>	✓ unitChanged 1/1 passed, 6ms
	>	✓ postalCodeChanged 1/1 passed, 6ms
	>	✓ operatingHoursChanged 1/1 passed, 6ms
	>	✓ descriptionChanged 1/1 passed, 5ms
	>	✓ imageChanged 1/1 passed, 6ms
	>	✓ imageStringChanged 1/1 passed, 6ms
	>	✓ pickImage 2/2 passed, 40ms
	>	✓ deleteImage 2/2 passed, 12ms
✓	✓	✓ retailer_repository_test.dart 10/10 passed, 146ms
	>	✓ getRetailer 5/5 passed, 111ms
	>	✓ updateRetailer 5/5 passed, 35ms
✓	✓	✓ product_repository_test.dart 21/21 passed, 162ms
	>	✓ getProductList 4/4 passed, 66ms
	>	✓ generateNewProductId 1/1 passed, 5ms
	>	✓ addProduct 4/4 passed, 22ms
	>	✓ updateProduct 4/4 passed, 22ms
	>	✓ deleteProduct 4/4 passed, 29ms
	>	✓ updateProductList 4/4 passed, 19ms
✓	✓	✓ add_product_form_notifier_test.dart 10/10 passed, 146ms
	>	✓ prodNameChanged 1/1 passed, 64ms
	>	✓ prodUsualPriceChanged 1/1 passed, 4ms
	>	✓ prodDiscountedPriceChanged 1/1 passed, 6ms
	>	✓ prodImageChanged 1/1 passed, 6ms
	>	✓ prodDescriptionChanged 1/1 passed, 4ms
	>	✓ prodAvailabilityChanged 1/1 passed, 5ms
	>	✓ pickImage 2/2 passed, 26ms
	>	✓ deleteImage 2/2 passed, 9ms
✓ auth_notifier_test.dart 3/3 passed, 117ms		
	>	✓ checkIfUserIsAuthenticated 2/2 passed, 93ms
	>	✓ signOut 1/1 passed, 24ms
✓	✓	✓ auth_repository_test.dart 18/18 passed, 265ms
	>	✓ getFirebaseUser 3/3 passed, 83ms
	>	✓ signIn 4/4 passed, 69ms
	>	✓ consumerSignUp 4/4 passed, 37ms
	>	✓ retailerSignUp 5/5 passed, 58ms
	>	✓ signOut 2/2 passed, 18ms
✓	✓	✓ retailer_list_repository_test.dart 5/5 passed, 178ms
	>	✓ getRetailerList 5/5 passed, 178ms
✓	✓	✓ product_list_notifier_test.dart 2/2 passed, 153ms
	>	✓ getProductStream 2/2 passed, 153ms
✓	✓	✓ value_validator_test.dart 27/27 passed, 193ms
	>	✓ validateEmail 7/7 passed, 87ms
	>	✓ validatePassword 4/4 passed, 18ms
	>	✓ validatePostal 2/2 passed, 13ms
	>	✓ validateNotEmpty 2/2 passed, 11ms
	>	✓ validateUsualPrice 5/5 passed, 29ms
	>	✓ validateDiscountedPrice 7/7 passed, 35ms
✓	✓	✓ retailerDto_test.dart 1/1 passed, 68ms
	>	✓ toDomain 1/1 passed, 68ms

(sprint 4: 121 tests)

Sprints

Trello is the tool we chose to collaborate during the development of our application. It allows us to organise and distribute development tasks easily. Other than that, it allows us to see the tasks at a glance and we can access the board anytime as long as we have a device to access the website.

We have multiple section in which a task can be located:

- Backlog
 - This is where all tasks are located initially
- Sprint backlog
 - Tasks that we have decided to implement in the next sprint will be shifted to this section
- In progress
 - Each of us can pick any task from sprint backlog to implement and the task will be shifted to this section
- Blocked
 - If any of the member get stuck when implementing some tasks, the tasks will be shifted to this section and we will discuss and find a way to solve the problem on the next standup meeting
- Sprint completed (with due date)
 - A task will be shifted to this part after we agreed with the implementation during the standup meeting

Sprint log sorted by sprints

28/7/2021 - Sprint completed		
(feature) implement project wide firebase storage logic to store images		
①	<input type="radio"/>	SG
(feature) sign in logic for consumer	<input checked="" type="checkbox"/> 4/4	SG
(feature) sign up logic for consumer	<input checked="" type="checkbox"/> 4/4	SG
consumer home page (retailer list)	<input checked="" type="checkbox"/> 5/5	KH
consumer and retailer drawer menu	<input checked="" type="checkbox"/> 2/2	KH
(feature) sign out logic	<input checked="" type="checkbox"/> 4/4	SG
(feature) authentication routing logic	①	SG
(feature) welcome page UI for consumer	<input type="radio"/>	LH
(feature) sign in page UI for consumer		
(feature) consumer home page UI		
(feature) improved drawer widget placeholder UI		

(sprint 1)

04/08/2021 - Sprint completed

...

(feature) sign in logic for retailer

⌚ 4/4

SG

(feature) sign up logic for retailer

⌚

SG

UI for retailer sign up page

W

(feature) retailer toggle visibility

⌚

SG

(feedback) toggle password field visibility

⌚

SG

(feature) add product

KH

UI for add product page

W

(feature) display product list in retailer home page

⌚

SG

(feature) added in splash screen UI

LH

(UI) app theme data (color swatch, google fonts, text styles etc)

3/3

LH

(feature) added responsive values to UI

LH

(UI) product list page(retailer)

LH

(sprint 2)

11/8/2021 - Sprint completed

...

(feature) shop detail page(consumer)

3/3

KH

(feature) shop product list
page(consumer)

KH

(feature) profile page(retailer)

◎

SG

(feature) edit retailer information
(retailer)

◎

SG

(feature) toggle product availability

◎

SG

(feature) update product information

◎

SG

(feature) toggle all products on/off

◎

SG

(bug) fix bug where field values
becomes '0' before form submission
finishes

KH

(feature) retailer home page UI

LH

(feedback) combined sign in page UI
for both consumer and retailer

LH

(UI) shop product list
page(consumer)

LH

(UI) profile page ui(retailer)

J

(UI) edit retailer information(retailer)

J

(UI) shop detail page(consumer)

J

(sprint 3)

18/8/2021 - Sprint completed

...

(feedback) include availability status
in shop detail page (consumer)

KH

(feature) favourite/unfavourite a
retailer

◎

SG

(feature) delete product

◎

SG

(feedback) when the retailer go to
edit profile screen, when he delete
the current image without
reuploading a new one, the image is
not getting deleted in the database

◎

SG

(feedback) image error when the
retailer(without profile picture) go to
his profile or edit his profile

◎

SG

(feedback) image not getting deleted
properly when editing profile

◎

SG

(feature) search shop(consumer)

◎

SG

(feature) display favorites retailer
(consumer)

◎

SG

(UI) consolidated responsive values
and standardise design for all pages

LH

(sprint 4)

25/08/2021 - Sprint completed ...

(feedback) trim text fields input

◎

SG

(bug) fix bug for 'home' button in consumer drawer widget

KH

(feedback) description field missing

◎

SG

(bug) fix bug for missing placeholder image when retailer profile image is deleted

J

(sprint 5)

Sprint log sorted by members

Completed - Sin Hau	...	(feedback) toggle password field visibility
(feature) implement project wide firebase storage logic to store images	<input checked="" type="checkbox"/> SG	
(feature) sign in logic for consumer	<input checked="" type="checkbox"/> 0/4 SG	(feature) display product list in retailer home page
(feature) sign up logic for consumer	<input checked="" type="checkbox"/> 0/4 SG	(feature) profile page(retailer)
(feature) sign out logic	<input checked="" type="checkbox"/> 0/4 SG	(feature) edit retailer information (retailer)
(feature) authentication routing logic	<input checked="" type="checkbox"/> SG	(feature) toggle product availability
(feature) sign in logic for retailer	<input checked="" type="checkbox"/> 0/4 SG	(feature) update product information
(feature) sign up logic for retailer	<input checked="" type="checkbox"/> SG	(feature) toggle all products on/off
(feature) retailer toggle visibility	<input checked="" type="checkbox"/> SG	

(feature) favourite/unfavourite a retailer

SG

(feature) delete product

SG

(feedback) when the retailer go to edit profile screen, when he delete the current image without reuploading a new one, the image is not getting deleted in the database

SG

(feedback) image error when the retailer(without profile picture) go to his profile or edit his profile

SG

(feedback) image not getting deleted properly when editing profile

SG

(feature) search shop(consumer)

SG

(feature) display favorites retailer (consumer)

SG

(feedback) trim text fields input

SG

(feedback) description field missing

◎

SG

(task completed by Sin Hau)

Completed - Lawrence

...

welcome page UI (consumer)

LH

sign in page UI for (consumer)

LH

consumer home page UI

LH

improved UI drawer widget
placeholder

LH

splash screen UI

LH

combined signn in page UI
(consumer & retailer)

LH

added responsive values to UI

LH

product list page UI (retailer)

LH

retailer home page UI

LH

shop product list page (consumer)

LH

consolidated responsive values and
standardize design for all pages

LH

(task completed by Lawrence)

Completed - Jia Xiang ...

(UI) profile page ui(retailer)

J

(UI) edit retailer information(retailer)

J

(UI) shop detail page(consumer)

J

(bug) fix bug for missing placeholder
image when retailer profile image is
deleted

J

(task completed by Jia Xiang)

Completed - Kian Hui

...

consumer home page (retailer list)

5/5

KH

consumer and retailer drawer menu

0/2

KH

(feature) add product

KH

(feature) add product

KH

(feature) shop detail page(consumer)

0/3

KH

(feature) shop product list
page(consumer)

KH

(bug) fix bug where field values
becomes '0' before form submission
finishes

KH

(feedback) include availability status
in shop detail page (consumer)

KH

(bug) fix bug for 'home' button in
consumer drawer widget

KH

(task completed by Kian Hui)

Completed - Bao Wen

...

(UI) for Add product page [retailer]

W

(UI) for edit product page [retailer]

W

(UI) for retailer sign up page [retailer]

W

combine placeholder and buttons in column

W

solve additional placeholder issue

W

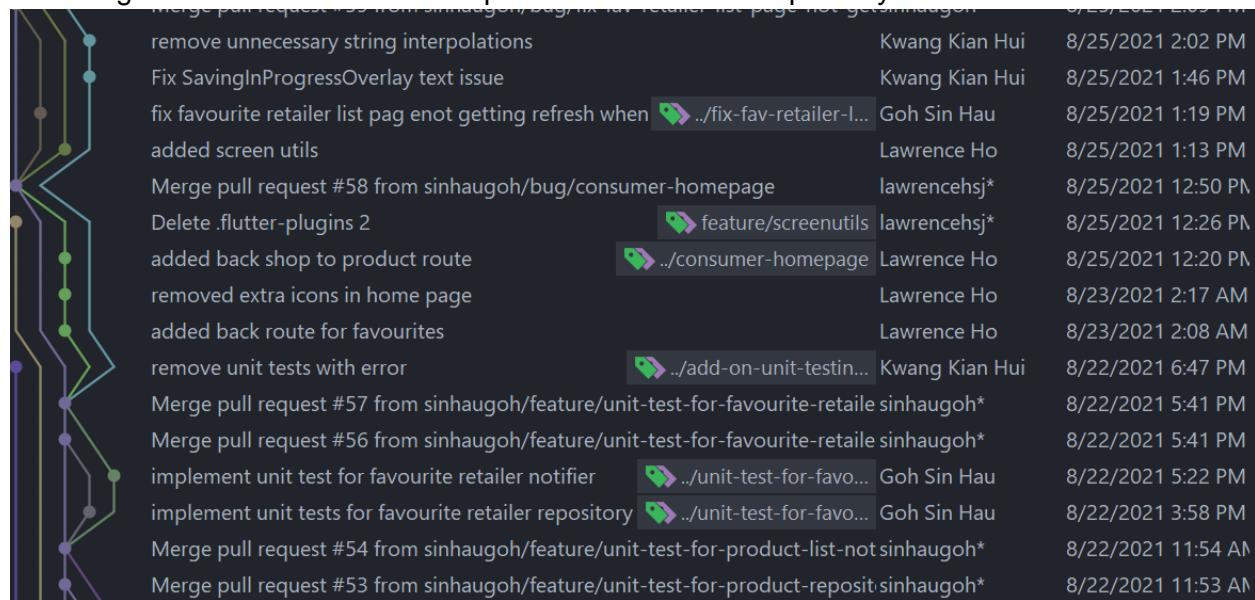
(task completed by Bao Wen)

Version control tool

Git is used as the version control tool while Github is used to host our project repository online. Our team decided to follow certain rules to make collaboration easier. These rules are:

- Main branch should be always working.
 - To accomplish this, a new branch will be created from the remote repository whenever any of us want to make changes to the project. After making desired changes to the branch, a pull request will be created and some of us will be requested to review the code. Lastly, the branch will be merged into the main branch if the branch is approved by the reviewers.
- Branch naming convention:
 - feature/{branch name}
 - Is created when any of us want to add in new feature to the application
 - optimisation/{branch name}
 - Is created when any of us want to make some improvement or optimisation to the application
 - bug/{branch name}
 - Is created when any of us want to fix bugs
 - junk/{branch name}
 - Is created when any of us want to experiment with some feature or third party packages

The image below shows some example of commits in our repository:



Commit Message	Author	Date
Merge pull request #59 from sinhaugoh/bug_fix_fav_retailer_list_page_not_getting...	Kwang Kian Hui	8/25/2021 2:03 PM
remove unnecessary string interpolations	Kwang Kian Hui	8/25/2021 2:02 PM
Fix SavingInProgressOverlay text issue	Kwang Kian Hui	8/25/2021 1:46 PM
fix favourite retailer list pag enot getting refresh when/fix-fav-retailer-l...	Goh Sin Hau	8/25/2021 1:19 PM
added screen utils	Lawrence Ho	8/25/2021 1:13 PM
Merge pull request #58 from sinhaugoh/bug/consumer-homepage	lawrencehsj*	8/25/2021 12:50 PM
Delete .flutter-plugins 2	feature/screenutils	8/25/2021 12:26 PM
added back shop to product route/consumer-homepage	8/25/2021 12:20 PM
removed extra icons in home page	Lawrence Ho	8/23/2021 2:17 AM
added back route for favourites	Lawrence Ho	8/23/2021 2:08 AM
remove unit tests with error/add-on-unit-testin...	Kwang Kian Hui
Merge pull request #57 from sinhaugoh/feature/unit-test-for-favourite-retail...	sinhaugoh*	8/22/2021 5:41 PM
Merge pull request #56 from sinhaugoh/feature/unit-test-for-favourite-retail...	sinhaugoh*	8/22/2021 5:41 PM
implement unit test for favourite retailer notifier/unit-test-for-favo...	Goh Sin Hau
implement unit tests for favourite retailer repository/unit-test-for-favo...	Goh Sin Hau
Merge pull request #54 from sinhaugoh/feature/unit-test-for-product-list-not...	sinhaugoh*	8/22/2021 11:54 AM
Merge pull request #53 from sinhaugoh/feature/unit-test-for-product-reposit...	sinhaugoh*	8/22/2021 11:53 AM

(example of commit from android studio version control integration)

-o-	Commits on Aug 25, 2021
	<p>Fix SavingInProgressOverlay text issue ...  Kwang-Kian-Hui committed 24 days ago</p> <p>fix favourite retailer list pag enot getting refresh when the connect...  sinhaugoh committed 24 days ago</p> <p>added screen utils ...  Lawrence Ho committed 24 days ago</p> <p>Merge pull request #58 from sinhaugoh/bug/consumer-homepage ...  lawrencehsj committed 25 days ago</p> <p>added back shop to product route  Lawrence Ho committed 25 days ago</p>
-o-	Commits on Aug 23, 2021
	<p>edited custom card to accomodate product  Lawrence Ho committed 27 days ago</p> <p>removed extra icons in home page ...  Lawrence Ho committed 27 days ago</p> <p>added back route for favourites ...  Lawrence Ho committed 27 days ago</p>
-o-	Commits on Aug 22, 2021
	<p>remove unit tests with error  Kwang-Kian-Hui committed 27 days ago</p> <p>Merge pull request #57 from sinhaugoh/feature/unit-test-for-favourite...  sinhaugoh committed 27 days ago</p> <p>Merge pull request #56 from sinhaugoh/feature/unit-test-for-favourite...  sinhaugoh committed 27 days ago</p>

(example of commit from github website)

To see the full commits of our project, refer to appendix section: A5 - Full Github Commit or check out: <https://github.com/sinhaugoh/DEALL>

8.0 User Testing

User interface evaluation (online):

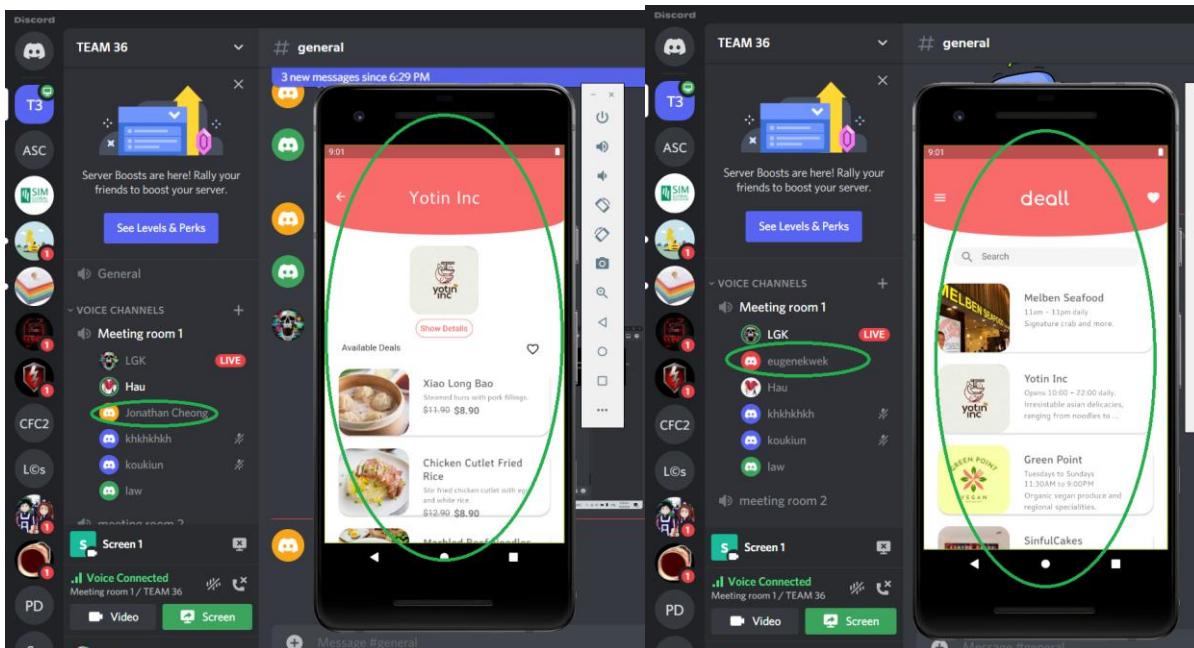
Due to the 2019 coronavirus pandemic, the Singapore's government has come up with preventive measures to curb the rapid boom in the numbers of infections. Therefore, physical meetings were restricted to groups of two to three, therefore we had to conduct our user testing via an online platform.

The platform that we have chosen for holding the interview was through discord, as we are able to share our screens with our user and show them the user interface of our application, while complying with the safety measures implemented by the Singapore government.

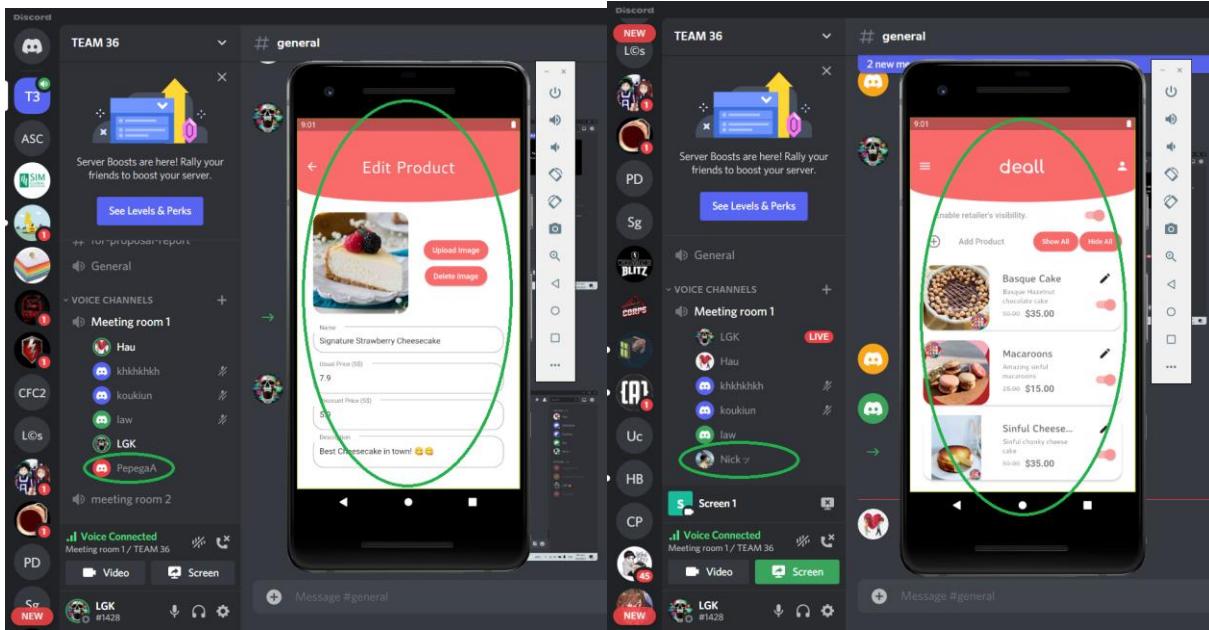
Users will share with us their thoughts on the presentation and the layout of our application. With the input from the user, we will tweak and improve the presentation to their suggestion before reviewing the updated application with them.

Another reason why we did not choose a face to face approach was because the data collected from the user was just their opinion and their thoughts with regards to our mobile application, thus it was not necessary to conduct a meet up interview session with them.

Consumer:



Retailer:



Usability testing:

The execution of the usability test was a face to face approach, as we wanted to observe their facial expression and identify the body language they are having when using our application. This will enable us to have a good gauge if our application is simple enough for them to operate and navigate through the various tasks that we have assigned them with to test our mobile application.

System Usability Scale(SUS):

The reason why we chose system usability scale was due to the simple scale that any user will be able to comprehend and understand. The 10 questions guide also enables both the user and developer to have both a gauge on the application that they are using or developing. With SUS, it allows developers to analyse if their application that they have created are within the passing means of the average score of 68. (Time Budgets for Usability Sessions, 2021).

B	C	D	E	F	G	H	I	J	K	L	M
I think I w I found t I though I think tr I found t I thought I would ir I found th I felt very I needed											SUS score:
4	1	4	1	3	2	4	1	4	2	32	80
5	2	5	1	5	1	5	1	5	1	39	97.5
5	2	5	1	5	1	5	1	5	1	39	97.5
5	1	5	1	5	1	5	1	5	1	40	100
5	2	5	2	4	2	5	2	4	1	34	85
4	2	5	1	4	1	4	1	5	1	36	90
5	1	4	2	4	1	4	2	5	1	35	87.5
4	2	5	1	4	2	4	1	4	2	33	82.5
5	1	4	1	5	1	5	1	5	1	39	97.5
3	2	4	1	5	1	4	1	3	1	33	82.5
											Average SUS score:
											90

From our SUS results, we can safely confirm that our results are better than the average SUS score of 90. With the sample size of 10, we are able to have assurance that our mobile application will be intuitive, simple and easy to use for new users. Although sample size can go as low as 2 user testers, we decided that 10 will be a better number as it will give us a better figure to gauge and it will be easy for us to do calculations (How Many Test Users in a Usability Study?, 2021).

Additional usability testing:

First Sprint:

Consumer		
Tasks	Able to complete (Yes/No)	Time taken to complete the task
User is able to sign up as a consumer		
User is able to sign in as a consumer		
User is able to sign out of their account		

Our ten consumer user testers for our first sprint were all able to complete the task that was set above in the table above. However, four out of ten had difficulty trying to enter their password. As they could not remember their password due to the obscure feature that we have implemented in the password section.

The testers also suggested having a toggle button so as to toggle the hiding function of the password. We have found out that with the consumer and retailer sign in pages, timing for completion of each individual task consumes more time, thus we decided to combine both sign in pages for the retailer and the consumer. It can also be observed that the users are facing troubles

navigating, as there will be occasional scratching of head, frowning and some frustration. With the input from the user, we will be implementing the suggestions into the next implementation.

Second Sprint:

In the second sprint, our team started to work on the features for the retailer's perspective which has features similar to that of the consumer, with the additional features like adding products to their respective retailer home page, toggling restaurant visibility.

After implementing the improvements from the first sprint, we reran the tests with the users to get their feedback on the improvements that have been made, by optimizing the sign in and sign up page for users under the consumer category.

Users felt that the process was smoother and easier to go through the sign up process after implementing the toggle function for obscuring the password.

Consumer (improvement testing)		
Tasks	Able to complete (Yes/No)	Time taken to complete the task
User is able to sign up as a consumer		
User is able to sign in as a consumer		

Retailer		
Features (retailer)	Able to complete (Yes/No)	Time taken to complete the task
User is able to create an account		
User is able to sign into their account created		
User is able to add a product		
User is able to toggle the visibility of their shop		
User is able to sign out safely from their account		

Users for both retailer and consumer are able to disable the obscure function by toggling the obscure symbol.

Users on the retailer are able to add to their products and they should be able to toggle the visibility of each individual food, enabling the consumer to view the discounted foods. Most of the features were relevant to them and mostly intuitive for them to navigate around.

However, users also pointed out that it was cumbersome for them to have a sign in page for consumer and another sign in page for retailer and they suggested if it was possible to combine the both of the features into one, so as to minimize confusion.

Third Sprint:

In our third sprint, we have combined both the consumer and retailer sign in page and improved the flow of the authentication process, by having the user retest the process of sign in into their respective accounts, before updating their feedback with regards to the process of signing up and signing in.

Users are pleased with the upgraded process and feel that the process shortens the time taken to complete each process and it is better than the previous version.

Consumer / Retailer (improvement testing)		
Task	Able to complete (Yes/No)	Time taken to complete the task
User is able sign up as a (consumer / retailer)		
User is able sign in as a (consumer / retailer)		
User is able sign out as a (consumer / retailer)		

Consumer		
Task	Able to complete (Yes/No)	Time taken to complete the task
User is able to select a restaurant and view the available products		
User is able to navigate to the retailer's details		

Retailer		
----------	--	--

Task	Able to complete (Yes/No)	Time taken to complete the task
User is able to view their profile page		
User is able to navigate to edit profile page		
User is able to toggle their product availability		
User is able to navigate to edit product page		

On top of that, we have included additional features for the consumer users, from viewing the restaurant's available products to obtaining the restaurant's address. This is to ensure the consumers are able to locate the address of the respective restaurants. Users (consumers) will be able to view the updated menu the retailer's have uploaded onto their restaurant page.

Features were implemented for the retailers, ranging from viewing of their profile page, toggling the visibility of their individual dishes in the menu to editing their account's products and their profile page.

With the implementation of the toggle switches for both the visibility of the shop and the foods, users (retailers) found it confusing on which toggle switch to use and requested some form of text as guidance.

Fourth Sprint:

During the fourth sprint, we asked the users (retailers) to look into the toggle switches again, after the implementation of the guides and all of the users claimed that it was better with the guided text on each use of the toggle switch.

Retailer (improvement testing)		
Task	Able to complete (Yes/No)	Time taken to complete the task
User is able to toggle visibility of their shop		
User is able to toggle availability of individual food		
User is able to view their profile page		

Consumer

Task	Able to complete	Time to complete
User is able to check availability status of a restaurant		
User is able to favourite/unfavourite a restaurant		
User is able to search for a restaurant (example: SinfulCakes)		
User is able to navigate to favourite retailer page to check for their favourite retailers		

Retailer		
Task	Able to complete	Time to complete
User is able to delete a product		

We added more features allowing consumers to check the availability of the products in the retailer's menu and enabling the users (consumers) to have a library of favourite restaurants that they would like to patronize consistently, and implement a search function to accelerate the process of finding their desired restaurants.

As for the retailer, we added the delete product function, allowing retailers to remove any unwanted product off their menu.

Through the various sprints that the two different groups of users that were testing our mobile application DEALL, we realised a few minor issues with the applications.

They are namely:

- image not being deleted when user have deleted the current image on their profile / menu and user was not able to change the photo
- image error issue when user did not upload any images to the image field
- image not deleted properly when the user edits their profile page

We have resolved all these issues that we have observed during the user testing phase and have also consolidated responsive values and standardise design for all pages.

The team have also did some minor touch up like:

- Trim text fields
- resolving missing description field
- Images in forms should be enlarged to allow them to see more clearly

Summary:

In this summary test, we will be testing our completed application and collating all the response time of the reaction of the user, where they are timed while completing the task assigned. Through this activity, we will be able to determine if our DEALL mobile application is intuitive by meeting the standard reaction time of two to three seconds, inclusive of completion of task instructed. (Todd DeCapua, n.d.)

Some examples of the test would be authentication flow like sign up, sign in and sign out (consumer / retailer), if you are a consumer we might have you navigate to a certain restaurant , finding a restaurant or even adding the restaurant that you like to your favourites list.

As a retailer, we may get the user to create an account with us, set up their account by adding some products in their menu or just simply do some editing of their profile description, while timing them.

Consumer			
Tasks	Initial page	Completion rate (%)	Average Time to complete (second)
User is able to sign up as a consumer	Welcome page	100	15
User is able to sign in as a consumer	Welcome page	100	10
User is able to sign out of their account	Home page	100	4
User is able to select a restaurant and view the available products	Home page	100	3
User is able to navigate to the retailer's details	Home page	100	5
User is able to check availability status of a restaurant	Home page	100	3
User is able to favourite/unfavourite a restaurant	Home page	100	4
User is able to search for a restaurant (example: SinfulCakes)	Home page	100	2
User is able to navigate to favourite retailer page to check for their favourite retailers	Home page	100	2

Retailer			
Tasks	Initial page	Completion rate (%)	Average Time to complete (second)
User is able to create an account	Welcome page	100	150
User is able to sign into their account created	Welcome page	100	10
User is able to add a product	Home page	100	35
User is able to toggle the visibility of their shop	Home page	100	1
User is able to sign out safely from their account	Home page	100	3
User is able to view their profile page	Home page	100	2
User is able to navigate to edit profile page	Home page	100	4
User is able to toggle their product availability	Home page	100	2
User is able to navigate to edit product page	Home page	100	2
User is able to delete a product	Home page	100	4

Findings

In this report, we did not implement the user guide for our application, as our findings in our usability test reflected that all of our test users were able to complete the task that was allocated to them. This shows that our application is intuitive, simple to understand and easy for any user to pick up quickly.

Our aim for task completion rate is about 78% completion, based on our research, benchmark for average completion rate is about 78% and the minimum sample size for our usability test will be at least 8 persons or more. Hence our team will be using the 78% completion rate benchmark for our completion testing (Jeff Sauro, 2021).

Our team will also be timing the result of the user's time to complete each task, so as to track if there are any improvements in timing if improvements have been implemented based on the user's feedback, after every test on the respective features.

Taking performance into consideration, we have also found out that in order to have efficient and productive feedback from our users, interaction has to be kept to a minimum of up to one hour per session. As research has shown that if any unit testing occurs over an hour, it will be too tiring for both the user and the developer, reducing the quality of the test. Therefore we are aiming to conduct the test within one hour, to ensure optimum focus for both the user and the developers. (Time Budgets for Usability Sessions, 2021)

9.0 Evaluation

Teamwork Processes

During the early ideation phase, the team decided to use an agile-oriented view on the project management process. A specific agile framework was not yet selected, so the team arranged weekly meetings through the use of Discord and each member would provide updates and discuss the next step of action to take. Throughout the week, short queries and discussion were also being engaged within the group's Telegram chat. The pace of work began at a steady pace and intensified when more research and requirements were gathered. Stakeholder engagement was slightly impeded due to the pandemic but we managed to conduct adequate physical as well as online interviews. Weekly meetings increased to twice a week and daily meetings were conducted in the days before the deadline. A couple of weeks before the submission, the team selected Scrum as a specific toolset to implement agile concepts in the software development process. Trello was selected to implement Scrum boards for visualisation of the Scrum process. Scrum boards were set up in preparation of sprints. We decided to implement these tools after the midterm submission, where the software development would actually commence.

During the software development phase, the team selected certain hallmarks of Scrum to aid in our agile-oriented management, such as sprints and daily meetings. Our sprints were planned roughly a week each to enable a User Centred Design(UCD) focus on fewer features that can be completed and presented to user/stakeholder for testing. Our daily meetings however were reduced to alternate days, as every member also had academic workload from other degree modules so it was not feasible to expect substantial progress from a single day. Discord continued to be used for meetings. Scrum boards visualised the entire process with Trello being a flurry of activity updates daily. As with the earlier phase, Telegram was also used for quick clarifications and updates daily. Before each sprint, a sprint meeting was held. The product backlog was reviewed and high priority items selected to be placed into the upcoming sprint's backlog. During the sprint, features will be pulled into a board for items in progress, features that encountered problems in development were placed into another board for blocked progress. Such issues were discussed in Scrum meetings throughout the sprint. During each Scrum meeting, each member would give updates based on the following two questions, "What have I done?" and "What am I going to do next?". Finished items at the end of sprints were moved into boards for completed sprints . A final review would take place after every sprint and discuss potential improvements. Github was used as a version control tool to manage the software development. In-between sprints, UCD was highly utilised as user/stakeholder feedback was continuously received through user/stakeholder testing. This assisted in the iterative improvements of features with a user-oriented focus.

Challenges Faced

There were time constraints as the team also had to fulfill responsibilities for other degree modules. This was further underscored by some of the members' unfamiliarity of the technology stack selected to develop the software. As Dart and Flutter were completely new to 3 of the 5 members, additional time had to be allocated for the learning process. One of the members experienced with Flutter and Dart had created an introductory guide to the stack. The 2 more experienced members with the stack took up infrastructure and backend tasks while the 3 inexperienced members learnt on the go, mostly working on the frontend UI. This arrangement ensured that each member was able to perform optimally to his ability. Sometimes, timings for meetings also had to be adjusted according to each member's schedule. Physical meetings were planned, however, due to the evolving state of pandemic, changing government restrictions meant that meetings had to be held online. This also added some element of complexity in communication. However we managed to have some instances of physical communication, such as liaising with users/stakeholders for feedback.

Product feature variations from original proposal

Most of the features from the original proposal were implemented, with variations. As user requirements evolved, a few functional requirements and use cases were altered. A couple of features were removed from development due to time limitations. They include the sorting via distance feature and the distance feature. However, trimming features from its original plan are not necessarily detrimental as retaining resource heavy features may sometimes bog down application performance. In this case of the distance feature, substantial computational power is required to measure distance between user and retailers. This may strain the application's ability to run at an optimal level had it been included. As the sorting via distance feature requires measurement of distances to multiple retailers, the processing load on the application is rather significant.

Application Evaluation(User)

To thoroughly evaluate an application's user facing quality, usability and accessibility metrics are deployed. Using Jakob Nielsen's usability heuristics(10 Usability Heuristics for User Interface Design, 2020),

1: Visibility of system status

The application implemented feedback such as error messages during wrong input and displaying loading animations for images. This could be further improved by adding notifications to inform users of more events happening in the application.

2: Match between system and the real world

The application used terms that were universally understood by all. From the Findings / Summary subsection in the User Testing section that there was full task completion by users. This shows that users understood terms in the application well enough in order to navigate.

3: User control and freedom

There are buttons to return to previous pages in the application. However there are no universal home page buttons included in every page as a quick exit feature.

4: Consistency and standards

As shown by the full user completion rate of tasks in the Findings / Summary subsection under User Testing section, users are able to consistently utilise features throughout the application. This reflects the consistency of design that is present in the application.

5: Error prevention

There are warning messages that appear during wrong input, in order to inform the user of the correct action. However there are no confirmation options before users perform an action. Including confirmation options may help to prevent errors from occurring as the user might reconsider their actions.

6: Recognition rather than recall

Most of the features in the application can be accessed from a few buttons or icons. The icons' graphics are selected with universally understood symbols, for example a heart shaped icon accesses the favourite feature. This ensures that the user retains the information intuitively.

7: Flexibility and efficiency of use

There is more than a way to use features in the application.

For inexperienced users, they might wish to use buttons labelled with terms, for experienced users, unlabeled icons are available for the same function. This is to ensure maximum efficiency in the user experience.

8: Aesthetic and minimalist design

The UI of the application has been designed with minimalism and attractiveness as factors.

9: Help users recognize, diagnose, and recover from errors

Error messages will appear due to wrong input or leaving a required field blank. The messages are displayed in natural language and coloured red to highlight attention to it.

10: Help and documentation

The application did not include a user guide as the results from user testing implied that a guide was not necessary. However, documentation can still be useful as there may be outliers.

Accessibility is important as people with disabilities might not be able to properly utilise the services of the application. However, the application did not test for accessibility nor considered accessibility as part of the design.

Potential Application improvements

After evaluating the completed application, the following are some features that could be considered for development in the future as enhancements.

Application services:

- A distance feature that sorts retailers by distance so that consumers can further customise the way our application displays the available retailers. As a result, the value of our application will increase due to improvement of user experience.
- A push notifications feature to inform the user(consumer side) when their favourite retailer has discounts available. This will add value to the user experience in terms of convenience. Other than that, user engagement and retention can be improved by having push notifications that are beneficial for the user (Giampedraglia, 2020).
- Allow users to change the password for their account. To protect against misuse of passwords compromised in a breach, consumers should promptly change affected passwords (Bhagavatula, Sruti & Bauer, Lujo & Kapadia, Apu, 2020). This is crucial in the context of user account security.
- Create an IOS version of the application. Currently the application is being developed for Android, with an IOS version the user base can be expanded. As Dart and Flutter are cross-platform, some configuration can be done to publish the application for IOS.
- An automatic toggle feature of retailer profile visibility based on set timings. Retailers might not wish to have their profiles be visible during periods when discounts are not being offered, as it may create misunderstandings. This is also to streamline processes to aid in convenience instead of having retailers to manually toggle visibility when required.
- A mechanism to filter retailers based on distance and price range. This will be convenient to users(consumer side) as they can use it to extract retailers that meet their requirements.
- An administrator for the application provider to verify Unique Entity Number(UEN) of retailers. This is crucial in the context of establishing the application as a legitimate platform with verified entities.
- Secure E-Payment methods through the app to automate and smoothen the process of the transaction. This will allow consumers flexibility and options to further their transactions with the retailers.

Application UI/UX:

- Introduce animations for page transitions and icons. This will create a more appealing experience to attract users. Through appealing animations, the user can wait patiently while the page gets loaded. This approach will at least engage the user in the app while reducing their frustration (Importance of Animation in App Design, 2021).

SWOT ANALYSIS

Strengths:	<ul style="list-style-type: none">• Only application in Singapore's market, that supports the environment (food wise) and provides business opportunities• Easy and convenient for users to use, both Retailers owners and consumers (mobile application)• Reduces food waste• Enable F&B owners to still profit• Enables F&B owners to cut losses from perishable food waste / surplus food made• Customers can enjoy delicious food at discounted rates from F&B stalls within their area• Have a better and cleaner environment with better food management
Weakness:	<ul style="list-style-type: none">• Customers must provide their own transportation for food collection• Online based application• No ordering from the application• Food availability
Opportunity:	<ul style="list-style-type: none">• Reduce food waste from the surplus food or cancelled orders• Enable restaurant owner to clear backlog inventory items• Allowing the food industry to be able to still profit• First application in the Asia market, specifically Singapore
Threats:	<ul style="list-style-type: none">• Pre-existing similar popular competitors in market (grabfood, foodpanda, deliveroo)• Pre-existing applications in the playstore market (gomkt, toogoodtogo, makan rescue)

STEEPLE Analysis

Social:	<ul style="list-style-type: none">• Covid19 introduced restrictions that affects crowd flow, affecting businesses• Technology has boomed significantly and everyone owns a mobile device, thus it's easier to engage with them through this application• An opportunity for us to introduce this app to boost sales of restaurant, cafe and food stall owners
Technology:	<ul style="list-style-type: none">• Design an app for android OS, apple IOS, webpage/website• Use of Flutter framework for our application (cross platform and easy to use), visual studio, jupyter notebook, brackets• Language used: Dart, C++, C#, python
Economics:	<ul style="list-style-type: none">• Implementation of advertisement in our app• Charge the restaurant a monthly subscription fee, after 3 free month trial
Environmental:	<ul style="list-style-type: none">• Management of surplus food• Increase CO2 emissions, contributing to global warming• Increase in landfill stowage to cope with food waste management
Political:	<ul style="list-style-type: none">• Pilot projects (2016–2018) cope with food waste in wet market• 2019 Year Towards Zero Waste campaign• Personal Data Protection Act (PDPA) to protect user's data
Legal:	<ul style="list-style-type: none">• Abide to all NEA rules and regulations with regards to both food and waste management• Abide to PDPA policies to ensure the safety of the user's data
Ethical:	<ul style="list-style-type: none">• Ethical to support efforts to reduce food waste which will contribute to global warming• Win-win situation for both the F&B owner and paying customer

Market comparison

Software/Company Name	Description
GoMkt	<ul style="list-style-type: none">• An online platform that allows retailers to post their unsold inventory for sale with business partners ranging from restaurants, cafes, and coffee shops, hosting discounts up to 75% less.• An inclusive function of gift cards and bonuses that boosts customer retention and partnership opportunities.
Toogoodtogo	<ul style="list-style-type: none">• A mobile application that provides an overview of shops and retailers that have surplus food based on the user's location, and the user can order and pay through the application.• User profile with list of favourite retailers.• Simple and direct UI that is not overly cluttered. Functionality is distinct right from the get-go.
Makan Rescue	<ul style="list-style-type: none">• A mobile application that notifies users when there are free food or food surpluses nearby that they can claim.• However, the usability is limited to certain university campuses, which is a very specific target audience.
Foodpanda/ Grafood	<ul style="list-style-type: none">• A food ordering application that allows users to order food within their vicinity and be delivered to their doorstep• Has a favourite / like system to allow users to have a collection of their favourite foods
Favepay	<ul style="list-style-type: none">• Payment system for food• Customers can enjoy discount and cashback through their app• Rebates on spending• Updates on the discounts shops are giving or the cashback they have

Key takeaway

The information we gathered during market analysis assisted us in refining our concept and ideation. We implemented certain features that we found largely successful such as:

- List of favorite retailers feature
- Simple and clean UI that shows distinct functionalities for each page.

Our application reaches out to all possible retail partners, with a single purpose in mind, which is to connect consumers and retailers. We aim to keep the functions as simple and clean as possible based on these successful case studies.

Our position in the market

Our app offers a simplistic approach to providing FnB promotions to consumers. As compared to similar apps on the market, we are definitely lacking in many features given the time constraints of the project. However, should the concept kick off, we feel that there is potential to further explore additional features to be implemented.

Setting our limitations aside, we analysed and consolidated limitations in the dominant apps within the market. We implemented a favourite page feature that allows easy access and reference for preferred outlets. This will be a convenient function for users to bypass scrolling through or searching for their preferred retailer. It also works along with our design standard of being simple and intuitive.

Comparatively, our app's unique concept is based on timely promotions (nearing closing hours) that provide a unique business opportunity that competitive food delivery applications do not have. This will be a first of its kind in Singapore's context.

10.0 User Guide

We will not be including an in-app user guide as our design standard inclines towards simplicity and intuitiveness. We have combed down on the learning curve to use our application, and the similarity towards other apps in the market will provide an easy transition for users to experience. Furthermore, our SUS test results are highly positive, proving that the app is straightforward enough to not require a comprehensive user guide.

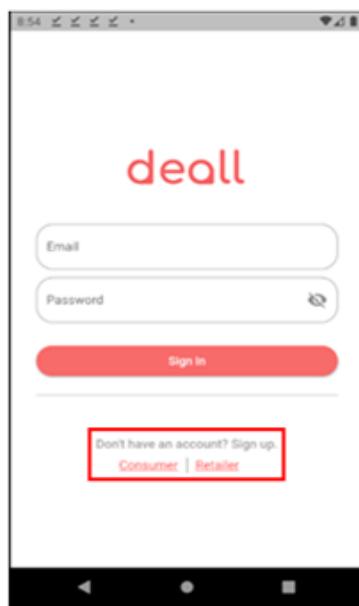
Instead, we will have a visual representation of our user guide to explain the application's structure and features:

Sign in page:

When launching the app for the first time, users will be brought to the sign in page.

Users may sign in by clicking the sign in button only after they have entered the required details.

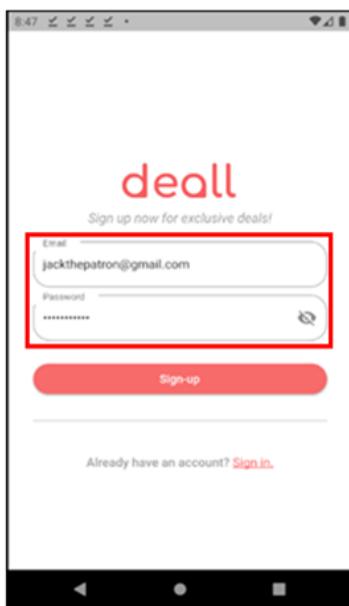
*Accounts already made available for testing can be found in appendix - section A1



Users may sign up for an account by clicking the respective user type that they want.

Consumer Sign Up Page:

When signing up as a consumer, only an email and password will be required but there are requirements to be met for them to be eligible



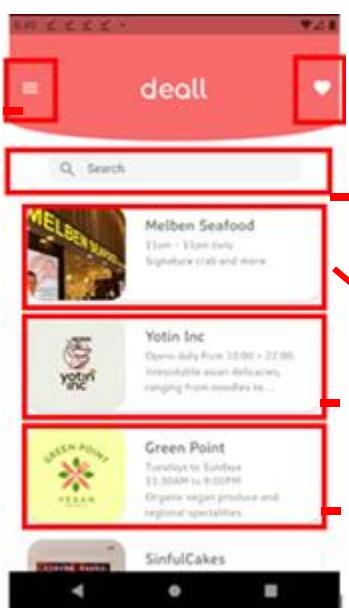
Users may press the sign up button to submit the details for a new account

Users may also press the sign in button to exit this page and proceed to sign in

Consumer Home Page (Retailer List):

After signing in or through a successful sign up for a consumer account, the user will be brought to the home page.

Users may navigate to all available pages or sign out from the hamburger menu. This menu can also be accessed by sliding the screen from the left to the right.



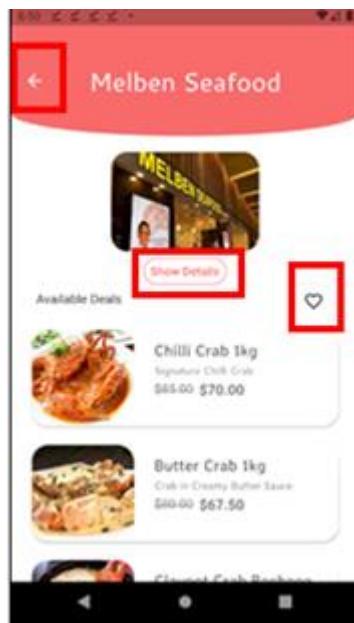
Users may navigate to the favourites page by clicking this icon or through the hamburger menu
Users may search for any retailer and any matches will be displayed regardless of the shop's visibility

Users may press any of these elements to view products offered by the respective retailer

Consumer Product List Page(Product list of selected retailer):

When a user selects a retailer item, they will be brought to the product list page.

Users may press the back arrow to navigate back to the previous page.



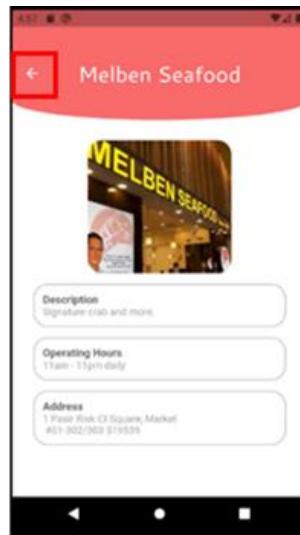
Users may press the show details button to navigate to the detail page showing the details of the retailer

Users may press the favourites icon to add this retailer to favourites which will be listed in the favourites page

Consumer - Retailer Detail Page (Selected retailer's detail):

When the show detail button is pressed in the product list page, the user will be navigated to the following detail page. This page does not have any elements that can be interacted with other than the back arrow icon button.

Users may press the back arrow to navigate back to the previous page.



Consumer Favourites Page:

When the user navigates to the favourites page, all favourited retailers will be displayed regardless of their visibility.

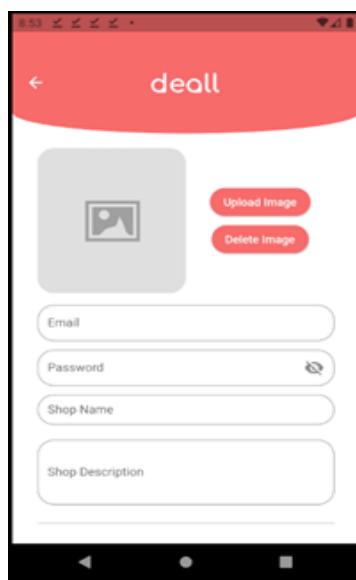
Users may press the back arrow to navigate back to the previous page.



Users may press any of these elements to view products offered by the respective retailer

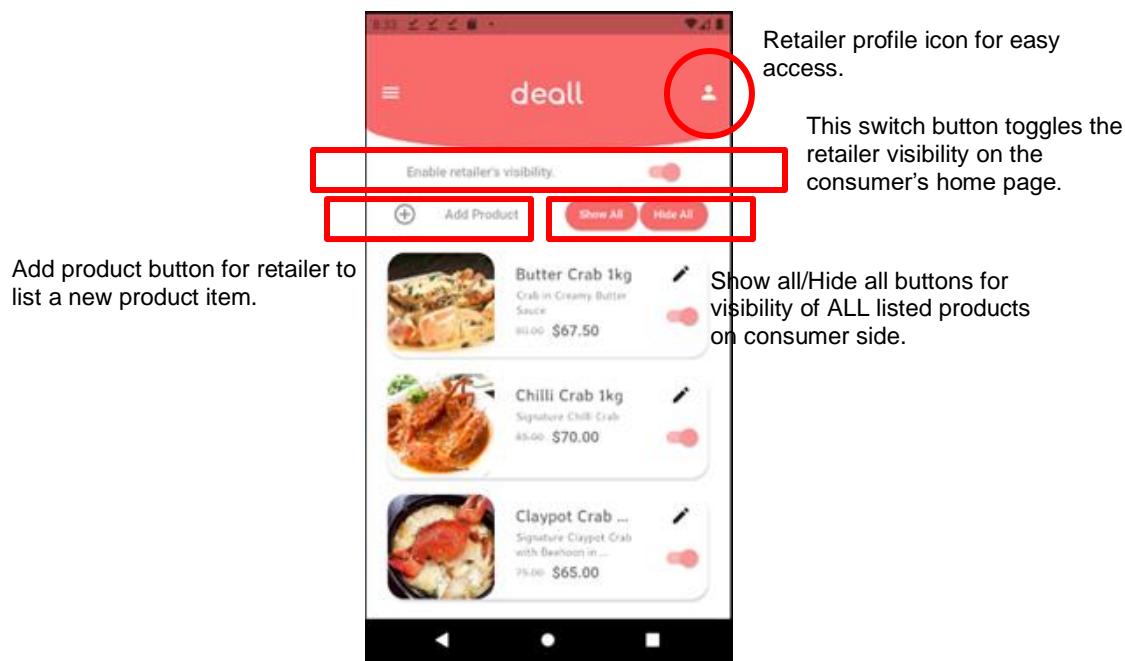
Retailer Sign Up Page:

The retailer sign up page is relatively straightforward and comparable to other applications with a sign-up interface. Simply fill up the necessary text fields and then click on the 'Sign-up' button at the bottom of the page.

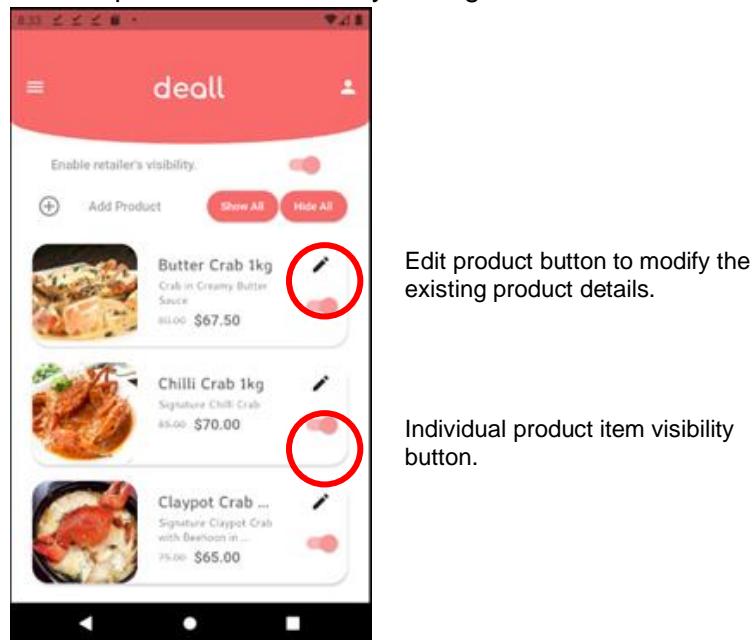


Retailer Home Page:

In the retailer's home page, the top section consists of options pertaining to the retailer's profile and product listing visibility. There are numerous buttons with labels that indicate their purpose, and they correlate to the visibility settings for the consumer side. There is also an icon at the top right hand corner to provide an easy and convenient access to the retailer's profile.

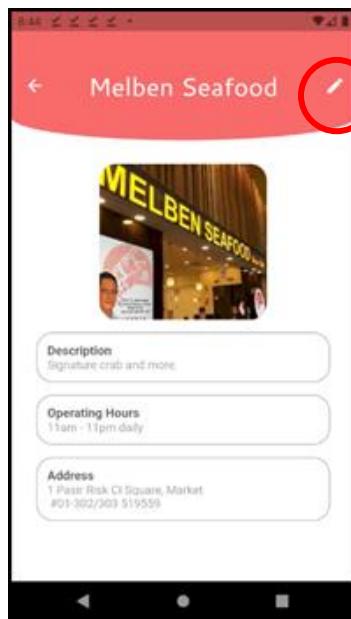


For the individual product cards, there are options for them as well. These options are not as comprehensible as the others, but once understood, it is actually really simple as well. Similar to the options in the top of the page, it corresponds to the visibility settings in the consumer end.



Retailer Profile Page:

The retailer's profile page displays the retailer's details, along with the edit profile button.

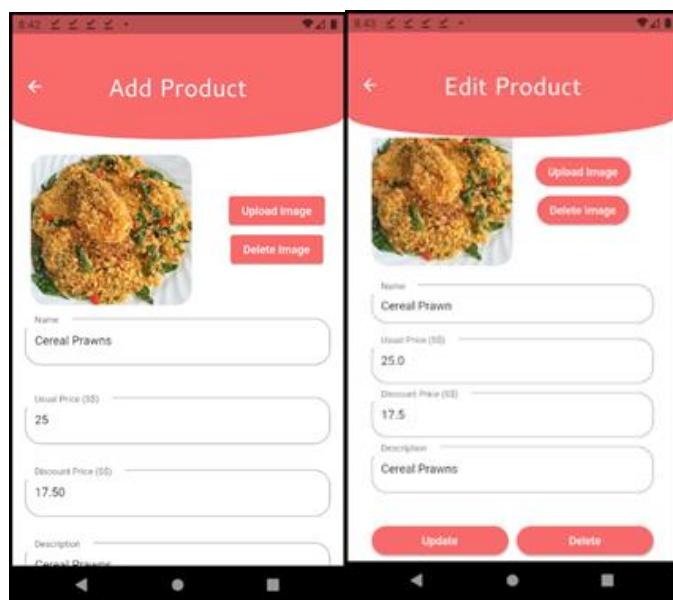


Edit profile button to modify the existing profile details.

Retailer Add Product/Edit Product Page:

Input fields to add product listing accordingly.

Input fields to edit existing product listings accordingly.



Regardless, we have provided a demo video stored under Appendix - section A2.

11.0 Conclusion

Life cycle beyond delivery stage

Our software has a very specific window of business, and for the most part, serves as a channel for consumers and retailers. With regards to determining the success of the app, we can only observe and look out for users' needs in due time and make adjustments accordingly. The given time constraints limit the amount of features we were able to implement. As a result, there may be implications on the simplicity and lack of features within the application, and However, should the concept be well received, we feel that with the addition of more substantial features, it is highly probable for the app to succeed and be able to compete with larger corporations within the market.

The region that we are based in, Singapore, has not had a successful launch of similar applications, hence we feel that it would definitely have an impact in the f&b industry. It will be a unique approach to clearing food items.

In terms of the software development, we can conclude that our app has been primarily successful based on the success metrics used to evaluate our findings. Throughout the development, we have integrated secure programming to handle exceptions and errors, especially with the database. As a result, even without continuous updates, the integrated database and application should be robust enough to be able to sustain without any prolonged maintenance.

Application of software

Our app targets a niche ideology of clearing food stocks whilst tackling food waste.

In today's context, with the uncertainty of the pandemic, food delivery and distributor services have become much more prevalent. More and more people are resorting to purchasing and having food delivered to them.

Food businesses have to face uncertainty on how much food to purchase, as unprecedented closures may occur. When faced with such a dilemma, naturally any form of business will be beneficial. Through application of our concept, we are providing a channel for consumers to be able to purchase discounted fnb items, in an effort to support businesses as well as creating an outlet to reduce the likelihood of food waste. Likewise, even with all other fnb retailers, it is an additional outlet of communication and opportunity of business and enjoyment for both ends of the deal.

With the simplicity and intuitiveness that we worked towards, it should not be a tough concept and interface to grasp, and the low learning curve presents itself positively.

Impact of software

Our application caters for everyone, and does not have any age or maturity restrictions. We aim to reach out to as many people as possible through this, and provide these green food management opportunities.

The concept of integrating food promotions and a resolve to reduce food waste will be our solution to answering global food waste. We seek to impact not just people who use our application, but also the environment as well. We hope to see a demand of users patronizing our app, which in return will raise more awareness on the topic of food waste.

Appendix

A1 - Test accounts

Consumers

Username: test@gmail.com

Password: abcd1234

Username: greenpointe@gmail.com

Password: qwerty123

Username: jackthepatron@gmail.com

Password: jackjack123

Username: admin@admin.com

Password: admin123

Username: wsl@gmail.com

Password: qwerty12345

Retailers

Username: hau@gmail.com

Password: abcd1234

Username: greenpoint@gmail.com

Password: greenpoint1

Username: melbenseafood@gmail.com

Password: melbensea123

Username: mrbean@gmail.com

Password: beanybean123

Username: sinfulcakes@gmail.com

Password: 12345qwerty

Username: yotin@yotin.com

Password: yotin123

A2 - Demonstration video link

Consumer perspective

Youtube link: <https://youtu.be/uCt3ScmO1s0>

Retailer perspective

Youtube link: <https://youtu.be/jK1BPayk9rU>

A3 - Github link

Link to our github repository: <https://github.com/sinhuagoh/DEALL>

A4 - Other Technical details:

Version of language and framework that we used to build our application:

- Flutter 2.2.3
- Dart 2.13.4

The source code that we have uploaded will not include the dependencies files because they take up too much space. In order to run the code, please use the following commands:

1. “flutter pub get”
 - a. to download all dependencies
2. “flutter run”
 - a. runs the code.

Within our project, there will be two folders that hold our code implementation. The “lib” folder will hold the codes for UI and logics that we have implemented. The “test” folder will hold all the unit tests that we have done to ensure our application is robust.

A5 - Full git commits

commit a853ca703c2c5ff02404f01169462c8f6961a1b9

Merge: c180591 102e98f

Author: sinhuagoh <shgoh025@mymail.sim.edu.sg>

Date: Sun Sep 19 20:29:43 2021 +0800

Merge pull request #85 from sinhuagoh/optimisation/disable-landscape-mode

disable landscape mode

commit 102e98f1467ec2f9b7695f4806a7a9c8fa4c1d47

Author: sinhaugoh <sinhaugoh456@hotmail.com>
Date: Sun Sep 19 20:27:56 2021 +0800

disable landscape mode

commit c180591c71d678e6baee89cb377a63c1dbc72e9b
Merge: 6520122 d69d2a0
Author: sinhaugoh <shgoh025@mymail.sim.edu.sg>
Date: Sun Sep 19 01:38:10 2021 +0800

Merge pull request #84 from sinhaugoh/optimisation/remove-warnings

removes warnings

commit d69d2a0a63efbd6a8fc212e50412c1febc663a1b
Author: sinhaugoh <sinhaugoh456@hotmail.com>
Date: Sun Sep 19 00:42:09 2021 +0800

removes warnings

commit 652012205b31f7ab5f2a112a7270f55ce7dfffff
Merge: 23be543 d530c39
Author: lawrencehsj <58553029+lawrencehsj@users.noreply.github.com>
Date: Sat Sep 18 23:37:04 2021 +0800

Merge pull request #83 from sinhaugoh/bug/responsive-improvements

Bug/responsive improvements

commit d530c39d065bbc55528a9412cbecd82753057661
Author: Lawrence Ho <lawrencehsj@gmail.com>
Date: Sat Sep 18 18:44:38 2021 +0800

bump

commit 7c230857b7dbe0de72e2bb24895447ce7fa78a7d
Author: Lawrence Ho <lawrencehsj@gmail.com>
Date: Sat Sep 18 18:30:28 2021 +0800

adjusted values in card

commit 8a5b52deba7f87ae65c842151e58f64ef4009171
Author: Lawrence Ho <lawrencehsj@gmail.com>
Date: Sat Sep 18 18:22:19 2021 +0800

add spacing for smaller devices at the bottom of screen for consumer sign in and sign up form

commit 9ed265b19f0b24c1f68778fd9b0e105eacdec4db

Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Sat Sep 18 18:17:19 2021 +0800

adjusted constraints for sign_in_form and consumer_sign_up_form

commit 1efe1e85aaa65e5a822855ec39c16f01d5b2ab19

Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Sat Sep 18 18:13:58 2021 +0800

updated the constraints for retailer form_text_field

commit ec36cf5277bfd13a2a94de0ca289323deb0f86b4

Author: Kwang Kian Hui <2k1h297@gmail.com>

Date: Sat Sep 18 17:37:25 2021 +0800

implement new appbar for add product page

commit 541e89e0367401b0d3661e2c957fbf32397d1f5c

Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Sat Sep 18 14:50:42 2021 +0800

remove extra comments

commit 856b5942c4680ba910b6d4bb551fc654539d12b7

Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Sat Sep 18 14:45:18 2021 +0800

remove unused imports and other problems

commit c9e35bba9db8085bc1e49d4c52b737b15c01a6d3

Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Sat Sep 18 14:32:31 2021 +0800

added back appbar logo

commit 11691ff64411c777b82170b81f4bd014a5f772a7

Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Sat Sep 18 14:25:23 2021 +0800

bump

commit 80a61b678e030c1dbd18b86c672601d0cc317189

Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Sat Sep 18 13:35:21 2021 +0800

adjust appbar font size

commit 3ffffdfa2123bc0494bf0ec27a0e4cee500e731de

Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Sat Sep 18 12:58:54 2021 +0800

bump

commit 3a3eb059091d0fef3fb21000a034e452c55e9d09

Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Sat Sep 18 12:56:24 2021 +0800

bump

commit 77e64c604dbe2f11acc4b435c531fd25f28ad798

Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Sat Sep 18 12:33:27 2021 +0800

bump

commit 41138b9539ad683368108932d448b5499de610f4

Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Sat Sep 18 12:32:23 2021 +0800

bump

commit c0de6820b79682eebf07dec966478ecd42cbdeb7

Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Sat Sep 18 12:28:50 2021 +0800

bump

commit 3b8637e25f09de365fe1481c683acb4ee55931dd

Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Sat Sep 18 11:39:29 2021 +0800

bump

commit 6fdb905bcf8c3a6826870777be406e09ae7053eb
Author: Lawrence Ho <lawrencehsj@gmail.com>
Date: Sat Sep 18 11:31:36 2021 +0800

edited style_theme and bumped values

commit 70a48b5c14d47653b543d98e0b41999314c659a7
Author: Lawrence Ho <lawrencehsj@gmail.com>
Date: Sat Sep 18 10:05:37 2021 +0800

bump

commit 2f4bda90578f51f224908e93fb4117581de5f56a
Author: Lawrence Ho <lawrencehsj@gmail.com>
Date: Sat Sep 18 09:48:07 2021 +0800

adjusted values

commit 09e4ef0a48b0da4f920b3e583ebb549ca1ec2bf2
Author: Lawrence Ho <lawrencehsj@gmail.com>
Date: Sat Sep 18 09:43:25 2021 +0800

bump

commit 3b74eb91c82894776613f1baf6fa873af6b56749
Author: Lawrence Ho <lawrencehsj@gmail.com>
Date: Sat Sep 18 09:34:36 2021 +0800

bump

commit 501c8969517cf64490301eb3a0efda09fbdff5b9
Author: Lawrence Ho <lawrencehsj@gmail.com>
Date: Sat Sep 18 09:32:26 2021 +0800

updated retailer_profile_page values

commit b15ef81cb0c99a99e9e0beb2e96cef0e31c08f30
Author: Lawrence Ho <lawrencehsj@gmail.com>
Date: Sat Sep 18 09:29:39 2021 +0800

updated edit_product_form values

commit dc4adca67ec57b77184bde938651f61f8bbf1195
Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Sat Sep 18 00:36:50 2021 +0800

update values in edit_product_form

commit b42c3876e2a427cd7d86d8db50c4b39ff0c37ff1

Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Sat Sep 18 00:28:44 2021 +0800

updated edit_profile_form values

commit d5cdb845720c958756a4921b0c3650aa51b95b5c

Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Fri Sep 17 23:02:50 2021 +0800

bump

commit c8a322aac1b07575c677cd001edf611da44a11d4

Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Fri Sep 17 22:58:49 2021 +0800

standardized appbar for other pages

commit 23e9457607571230fa4739a0a9bf7838e44fd508

Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Thu Sep 16 18:24:58 2021 +0800

edit custom_card text values

commit 1487396465cb3c8cbf3494af9c51149f8892a36f

Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Thu Sep 16 18:17:52 2021 +0800

adjusted consumer_home_page favourite icon

commit 60e31ace46fd40d71042cd068547e34e4af440cb

Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Thu Sep 16 18:17:13 2021 +0800

adjusted retailer_home_page appbar

commit 59ad658ac23c84e8b7bd2f04d80e189e6a515b74

Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Thu Sep 16 18:09:35 2021 +0800

adjusted UI and responsive values of retailer_sign_up_page

commit 4f64a928130294efbb76d5c19c3845e5470edebe

Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Thu Sep 16 17:41:59 2021 +0800

bump

commit 31615c0ac821232640d4af43fa8ccb2ff4d1eaa1

Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Sun Sep 12 15:10:32 2021 +0800

bump

commit 291d2f788870844a91c5d36977c303fa06551a38

Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Sun Sep 12 15:04:28 2021 +0800

update retailer_sign_up_form responsive values

commit 0feec281da4c00681d3ea60f8e500d8ad84b85d7

Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Sun Sep 12 13:17:01 2021 +0800

update consumer_home_page responsive values

created custom_appbar class

commit 13eadc81e89eba53e3bad499aa96174451951246

Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Sun Sep 12 12:38:37 2021 +0800

bump

commit 2e010c24438be2cf31cdcb73cf3336066924be5

Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Sun Sep 12 12:28:45 2021 +0800

improved consumer_sign_up_page responsive values

commit ab8bd611594f067637073fb33590cb13e9100943

Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Sun Sep 12 12:12:23 2021 +0800

improved sign_in_page responsive values

commit 23be5439c7ba8b610f6420e8c1c5240e68000046

Merge: 8bfb47a 6c743fd

Author: Kwang Kian Hui <2k1h297@gmail.com>

Date: Mon Sep 6 20:59:37 2021 +0800

Merge pull request #82 from sinhaugoh/bug/fix-pricing-text

implement pricing text fix

commit 6c743fd94871f187a3b1bd9ac7705c35776800b7

Author: Kwang Kian Hui <2k1h297@gmail.com>

Date: Mon Sep 6 19:00:26 2021 +0800

implement pricing text fix

commit 8bfb47ad6f0f847c8d61b3520c8d6ff64d6a2144

Merge: 0b90d7e 517b595

Author: Kwang Kian Hui <2k1h297@gmail.com>

Date: Mon Sep 6 17:47:15 2021 +0800

Merge pull request #81 from sinhaugoh/improvements/implement-center-title-for-appbars

Implement 'centerTitle=true' for all pages

commit 517b595346c1f5d9b2fb11c32815654462269ef5

Author: Kwang Kian Hui <2k1h297@gmail.com>

Date: Mon Sep 6 17:46:21 2021 +0800

Implement 'centerTitle=true' for all pages

commit 0b90d7efbc4ade7bfccc77619d33431974cf07df

Merge: 0a3ce3e ca140cb

Author: LETSGOKILLING <77219102+LETSGOKILLING@users.noreply.github.com>

Date: Mon Sep 6 15:43:49 2021 +0800

Merge pull request #79 from sinhaugoh/feature/retailer-edit-product-page

Feature/retailer edit product page

commit 0a3ce3eef10d23301db6d83df3a18e2954302795

Merge: 2e9c997 a45829e

Author: sinhaugoh <shgoh025@mymail.sim.edu.sg>

Date: Mon Sep 6 15:34:50 2021 +0800

Merge pull request #80 from sinhaugoh/improvement/remove-debug-banner

Update app_widget.dart

commit a45829e6262be8cf8589ba854e4b2d4a7b673dcf

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Mon Sep 6 15:34:24 2021 +0800

Update app_widget.dart

commit ca140cb086ab603b546662aff898ec1d97eaae0f

Author: LETSGOKILLING <77219102+LETSGOKILLING@users.noreply.github.com>

Date: Mon Sep 6 13:21:40 2021 +0800

update edit product form

commit d782ec7be33c07f68e5ddf3d67b353b7c08b6b57

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Mon Sep 6 13:05:47 2021 +0800

implement image UI logic

commit fef182415f8efb32cbc38ec2aa21e0ef19275a95

Author: LETSGOKILLING <77219102+LETSGOKILLING@users.noreply.github.com>

Date: Mon Sep 6 11:14:24 2021 +0800

Update retailer edit product page

commit ba06ffd5b9a1fc8cd4f2d823b45961210eb172b

Author: LETSGOKILLING <77219102+LETSGOKILLING@users.noreply.github.com>

Date: Mon Sep 6 10:15:00 2021 +0800

placeholder with image issue

just for troubleshooting

commit 2e9c99754d7e448e611a2908b4c4d355bb8f74b1

Merge: 871e797 0e7d828

Author: LETSGOKILLING <77219102+LETSGOKILLING@users.noreply.github.com>

Date: Mon Sep 6 09:23:35 2021 +0800

Merge pull request #78 from sinhaugoh/feature/retailer-add-product-page

implement ui for add product form

commit 0e7d8282662467422a53f9dec6fa86271afb6e59

Author: LETSGOKILLING <77219102+LETSGOKILLING@users.noreply.github.com>

Date: Mon Sep 6 00:38:43 2021 +0800

implement ui for add product form

commit 871e797fc3100d73951eabf915f9933224dcd0cd

Merge: 3c7ad60 7c7efc5

Author: JXBdr <87599994+JXBdr@users.noreply.github.com>

Date: Sun Sep 5 17:00:26 2021 +0800

Merge pull request #77 from sinhaugoh/bug/retailer-profile-ui

Bug fix

commit 7c7efc581355967e52fac845b6e9ce79cf3bcff4

Author: JXBdr <kohjiaxiang15@gmail.com>

Date: Sun Sep 5 16:58:05 2021 +0800

Bug fix

commit afe3051adf500760d6378f009dfa03f8d4facb38

Author: JXBdr <kohjiaxiang15@gmail.com>

Date: Sun Sep 5 16:02:34 2021 +0800

Bug fix

Fixed missing no image display on Retailer Profile and Edit Profile pages

commit 9aa3c4766b606831329dbab817a098aa7548335f

Author: JXBdr <kohjiaxiang15@gmail.com>

Date: Sun Sep 5 13:30:21 2021 +0800

Bug fix

Added no image display on Retailer Edit Profile page

Reduced size of loading animations across Retailer Profile and Consumer Retailer Details page

commit 3c7ad6062c036b0edcda49761f818c6ccacb5a63

Merge: 96ef83d f184cb5

Author: lawrencehsj <58553029+lawrencehsj@users.noreply.github.com>
Date: Sat Sep 4 23:26:36 2021 +0800

Merge pull request #76 from sinhaugoh/bug/retailer-homepage-textoverflow
fixed retailerhomepage text overflow

commit f184cb5c37d1cb71caaf52c96f9bee6c0d12a8f1
Author: Lawrence Ho <lawrencehsj@gmail.com>
Date: Sat Sep 4 23:25:07 2021 +0800

fixed retailerhomepage text overflow

commit 96ef83dd831bde3be3b82c70c4fb942f16a2fbce
Merge: 0da01a5 e4885a6
Author: JXBdr <87599994+JXBdr@users.noreply.github.com>
Date: Sat Sep 4 23:12:48 2021 +0800

Merge pull request #75 from sinhaugoh/bug/retailer-profile-ui

Resolve bug

commit e4885a64124964e11b239dce22650ac36faabf41
Author: JXBdr <kohjiaxiang15@gmail.com>
Date: Sat Sep 4 23:10:25 2021 +0800

Resolve bug

resolved double display of image in profile edit page

commit 0da01a56799b5b1e9fdcaa99d2cc650905c767bf
Merge: 1ded8d4 62c6b38
Author: JXBdr <87599994+JXBdr@users.noreply.github.com>
Date: Sat Sep 4 23:05:03 2021 +0800

Merge pull request #73 from sinhaugoh/feature/retailer_profile_ui

Feature/retailer profile UI

commit 62c6b385035453977bc84b2dd10704f8fe3b1346
Merge: 6385616 1ded8d4
Author: sinhaugoh <shgoh025@mymail.sim.edu.sg>
Date: Sat Sep 4 23:04:28 2021 +0800

Merge branch 'main' into feature/retailer_profile_ui

commit 1ded8d457e2ed3ace5b6e94a37cfefc2c8c1d552

Merge: 7a7b759 23a75a3

Author: lawrencehsj <58553029+lawrencehsj@users.noreply.github.com>

Date: Sat Sep 4 22:57:49 2021 +0800

Merge pull request #74 from sinhaugoh/bug/consumer-homepage-textoverflow

fixed text overflow in card

commit 23a75a3bbf95c90e2baed12ffca3a1094c732501

Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Sat Sep 4 22:56:29 2021 +0800

fixed text overflow in card

commit 638561677f2843e249540f03c72e98ec87d2402c

Author: JXBdr <kohjiaxiang15@gmail.com>

Date: Sat Sep 4 22:52:28 2021 +0800

UI update

change to cached image for better performance

commit 7a7b759b017129f1b3708a4352d9f06e043ee0aa

Merge: 21c85d0 86bb6a4

Author: JXBdr <87599994+JXBdr@users.noreply.github.com>

Date: Sat Sep 4 22:40:27 2021 +0800

Merge pull request #71 from sinhaugoh/feature/consumer-retailer-profile-ui

Feature/consumer retailer profile UI

commit 21c85d0366d71fd1de29503e10831cdb1fe21a83

Merge: 0a9aa66 c6a7416

Author: LETSGOKILLING <77219102+LETSGOKILLING@users.noreply.github.com>

Date: Sat Sep 4 22:38:40 2021 +0800

Merge pull request #68 from sinhaugoh/testing-retailer_Sign_up-tryout

retailer sign up page

commit 86bb6a4fa8d693190620535589e7303dd6105823

Author: JXBdr <kohjiaxiang15@gmail.com>
Date: Sat Sep 4 22:36:36 2021 +0800

UI update

change to cached image for better performance

commit c6a74168de925a9d2fff4544554b877d58e5e5f0
Author: LETSGOKILLING <77219102+LETSGOKILLING@users.noreply.github.com>
Date: Sat Sep 4 22:35:41 2021 +0800

combined if else into one

commit 6e9f9ecf307bfcb820844ddda7d98dd2dc5bba96
Author: LETSGOKILLING <77219102+LETSGOKILLING@users.noreply.github.com>
Date: Sat Sep 4 22:27:07 2021 +0800

remove empty space line 25 retailer sign up page.dartt

commit 89bd2cb8a9655065e00f9f484f84ce3f251f8873
Author: LETSGOKILLING <77219102+LETSGOKILLING@users.noreply.github.com>
Date: Sat Sep 4 22:24:37 2021 +0800

implement ui for retailer sign up form

commit 0a9aa6604128e34554575d1f8157ffb1191e0251
Merge: a6aa698 633090f
Author: lawrencehsj <58553029+lawrencehsj@users.noreply.github.com>
Date: Sat Sep 4 22:21:33 2021 +0800

Merge pull request #69 from sinhaugoh/feature/retailer-homepage

Feature/retailer homepage

commit 633090f52b651caecf4b2794a79c6fce835e5471
Author: Lawrence Ho <lawrencehsj@gmail.com>
Date: Sat Sep 4 22:12:06 2021 +0800

remove unnecessary code

commit a5ab71406e2d4e9309e8814f422b9c717e07309e
Author: Lawrence Ho <lawrencehsj@gmail.com>
Date: Sat Sep 4 22:10:50 2021 +0800

remove commented lines of code

commit 22785ad8808a56d35ea4295a73ee3b39e34c390b

Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Sat Sep 4 22:08:56 2021 +0800

remove product from customcard

commit fdc936828c44e59367e5deef1cf70197260f573

Author: JXBdr <kohjiaxiang15@gmail.com>

Date: Sat Sep 4 22:03:52 2021 +0800

UI update

commit a6aa698702e3ba0abb2d7a7e72cb029b6455665d

Merge: 06629a6 b8ed344

Author: sinhaugoh <shgoh025@mymail.sim.edu.sg>

Date: Sat Sep 4 17:32:05 2021 +0800

Merge pull request #72 from sinhaugoh/bug/no-description-field-for-retailer-sign-up-and-edit-profile-form

fix by adding the widget into the forms. The logic was there already

commit b8ed3443b3a7d99840d7aae84176646d9ddc3260

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Sat Sep 4 17:29:31 2021 +0800

fix by adding the widget into the forms. The logic was there already

commit 4e3e2dff70819b897315f448b0ce011c355a42bd

Author: JXBdr <kohjiaxiang15@gmail.com>

Date: Sat Sep 4 17:07:08 2021 +0800

UI update

commit 2136304d7717a50389c016c9f71ec1a468362879

Author: JXBdr <kohjiaxiang15@gmail.com>

Date: Sat Sep 4 16:28:50 2021 +0800

UI update

Consumer Retailer Detail Page UI update

commit eaca44b656ea15ed6e242d3b8a0dfa386b485e9c
Author: JXBdr <kohjiaxiang15@gmail.com>
Date: Sat Sep 4 15:11:08 2021 +0800

UI update

Retailer Profile page and Retailer edit Profile page UI update

commit 87239e0595dbb1f0d671df3159e5be3652c1992a
Merge: 04c3cbe 06629a6
Author: Lawrence Ho <lawrencehsj@gmail.com>
Date: Fri Sep 3 23:54:37 2021 +0800

Merge branch 'main' into feature/retailer-homepage

commit 06629a6085b9dfad441f75089686212b506a3a48
Merge: e77e634 5396f23
Author: lawrencehsj <58553029+lawrencehsj@users.noreply.github.com>
Date: Fri Sep 3 23:41:39 2021 +0800

Merge pull request #67 from sinhaugoh/feature/consumerdetailpage

Feature/consumerdetailpage

commit 5396f2381b0f609a8c6e25b2e265455f32d0ed07
Author: Lawrence Ho <lawrencehsj@gmail.com>
Date: Fri Sep 3 23:36:24 2021 +0800

update alignment

commit e77f32e84df24892c975887279c3eadf4ae6e808
Author: LETSGOKILLING <77219102+LETSGOKILLING@users.noreply.github.com>
Date: Fri Sep 3 23:29:17 2021 +0800

retailer sign up page

this is for the retailer sign up page , tested the logic for the image upload and delete function to check if image changes from shop image to "no image found" picture

commit 0e43cec641340073c1a3354e8f667d4b856b3859
Author: Lawrence Ho <lawrencehsj@gmail.com>
Date: Fri Sep 3 23:28:17 2021 +0800

adjusted textbutton (show details) appearance

commit 08cbd0c4293c2f507a5459415ff6c75da8cca0dd

Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Fri Sep 3 23:17:22 2021 +0800

removed favourite route and icon

commit 62cf1e28bfe18cf75263643fd29ad1c5135ff799

Merge: 4d90dca b3adffc

Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Fri Sep 3 22:48:40 2021 +0800

Merge branch 'main' into junk/experiment-consumerdetailpage

commit e77e6342cbdf2f20fdae750637b22e880a3c433b

Merge: b3adffc 861b7ed

Author: Kwang Kian Hui <2k1h297@gmail.com>

Date: Fri Sep 3 22:38:48 2021 +0800

Merge pull request #66 from sinhaugoh/bug/bug-fix-for-consumer-drawer-navigation

Implement bug fix for 'home' button in consumer drawer widget

commit 861b7ede4747928e07db6815bf6ba4f742c8ad8a

Author: Kwang Kian Hui <2k1h297@gmail.com>

Date: Fri Sep 3 22:38:10 2021 +0800

Implement bug fix for 'home' button in consumer drawer widget

commit 4d90dcaabec31c6f8923d98d589468eaad93d0c4

Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Fri Sep 3 22:22:20 2021 +0800

added boxfit.cover to product card

commit 3a28fe4ea1320a01ed2718ab7859ba1b61433002

Author: JXBdr <kohjiaxiang15@gmail.com>

Date: Fri Sep 3 22:03:35 2021 +0800

UI update

Profile UI and Edit Profile pages UI update

commit d88bae6f4d3c5a67fec0be439b9f146c04d6ffd6

Author: Lawrence Ho <lawrencehsj@gmail.com>
Date: Fri Sep 3 22:00:14 2021 +0800

changed placeholder image

commit 04c3cbe93ac3ff485f54402320353e2901701ce4
Author: Lawrence Ho <lawrencehsj@gmail.com>
Date: Fri Sep 3 21:55:10 2021 +0800

added ternary for no product

commit 9768d9d796938ace54e6487a5deb2a052a98326d
Author: Lawrence Ho <lawrencehsj@gmail.com>
Date: Fri Sep 3 21:17:30 2021 +0800

trying to clip image in rounded rectangle

commit 026c6c4008149dfc3caa8f9f63e6ab0b7c33bda0
Author: JXBdr <kohjiaxiang15@gmail.com>
Date: Fri Sep 3 21:15:36 2021 +0800

UI update

commit 3d327c6f07d814cd93cda439218711895ba78110
Author: Lawrence Ho <lawrencehsj@gmail.com>
Date: Fri Sep 3 21:05:14 2021 +0800

update detail page

-center appbar title
-change show details to button

commit 9add3148250511eda1725b89e050f4488bf494d3
Author: Lawrence Ho <lawrencehsj@gmail.com>
Date: Wed Sep 1 17:48:02 2021 +0800

added back edit card and availability

commit 695899d27b4287273a601996dcac904a86aa3a36
Author: Lawrence Ho <lawrencehsj@gmail.com>
Date: Wed Sep 1 16:06:49 2021 +0800

create custom_retailer_card

commit 506316a7f8c091c31ccdb752fa7659aac427b430
Author: Lawrence Ho <lawrencehsj@gmail.com>
Date: Wed Sep 1 15:57:56 2021 +0800

added comments

commit 2fba045806627fc4828732d3bfce0d9ae1359563
Author: Lawrence Ho <lawrencehsj@gmail.com>
Date: Wed Sep 1 15:49:19 2021 +0800

bold and enlarged discounted price

commit 9479822ed1ec983028c02be54573253b383fb1bf
Author: Lawrence Ho <lawrencehsj@gmail.com>
Date: Wed Sep 1 15:45:05 2021 +0800

update price

commit 296f63996144f921fbac7ae5a0033eeb47498771
Author: JXBdr <kohjiaxiang15@gmail.com>
Date: Wed Sep 1 15:29:04 2021 +0800

Test

commit 0952efb4fc2967efb73d616539d48280b7557554
Author: TSCOTech <theshippingcompanyofficial@gmail.com>
Date: Wed Sep 1 15:15:05 2021 +0800

test

commit 26aaa2f7a6bf7efcdc980f85583b5312b9ef1687f
Author: Lawrence Ho <lawrencehsj@gmail.com>
Date: Wed Sep 1 15:04:20 2021 +0800

Update retailer_home_page.dart

commit 2dfaf50551103d46d3206d57e79995be8751d1b1
Author: Lawrence Ho <lawrencehsj@gmail.com>
Date: Wed Sep 1 04:35:25 2021 +0800

adjusted values

commit 3c7d54cee0654d277706b17c088cfa23b1eeaac0
Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Wed Sep 1 04:27:00 2021 +0800

added ternary for no products

commit 7d01dd00807be47b05a272c9e48f0620648376ed

Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Wed Sep 1 04:15:26 2021 +0800

update favourites appbar

commit ec6d830e3212cf369521300d2c82914ce084f609

Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Wed Sep 1 04:01:11 2021 +0800

edited appbar

commit 7493d17c5de954bb48dacf64e0374a0585158122

Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Wed Sep 1 03:37:50 2021 +0800

update ternary

commit 91c80229b8fc04927e3b1b669628c75bc57f3494

Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Wed Sep 1 03:37:06 2021 +0800

added ternary for no product

commit f62aacf908ca731570e5f8e8b5dd6c6126fb8d7d

Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Wed Sep 1 03:26:23 2021 +0800

added ternary for no image

commit 1739da635fb45f68232b188fd39c9744302feef

Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Wed Sep 1 02:32:33 2021 +0800

adjusted layout

commit 028173698e06cbae95c10b5ce8a8099c3547230b

Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Wed Sep 1 01:43:21 2021 +0800

added text for hide all products

commit cc71785833f011aecda64d06bbb0fecaf1d5bfd9
Author: Lawrence Ho <lawrencehsj@gmail.com>
Date: Wed Sep 1 00:28:09 2021 +0800

ternary operator for retailer home page not working

commit b3adffcc1ea1ae935049c9054dfc75897a41e0e2
Merge: 893190b 83c60f4
Author: Kwang Kian Hui <2k1h297@gmail.com>
Date: Tue Aug 31 15:10:55 2021 +0800

Merge pull request #16 from sinhaugoh/feature/add-on-unit-testing-to-retailer-list

Feature/add on unit testing to retailer list

commit b27d4d65683a41a7894813692a54128e3c5940b8
Author: Lawrence Ho <lawrencehsj@gmail.com>
Date: Tue Aug 31 15:05:51 2021 +0800

commented out some border edges

commit 21bd0c81a7655f41f934885e34338b30c70a7dff
Author: Lawrence Ho <lawrencehsj@gmail.com>
Date: Tue Aug 31 14:47:07 2021 +0800

added in card code

commit 7783af048c3d9a9d478d21c05e45c05d7ee99890
Author: Lawrence Ho <lawrencehsj@gmail.com>
Date: Tue Aug 31 12:46:42 2021 +0800

adjusted row alignment

-figuring out a way to add no product display

commit 893190b49af4f94c59c66875237f5be84ef5167d
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Sun Aug 29 15:02:58 2021 +0800

include official documentation in repositories and validators for core, auth and consumer features

commit 3ea0f5f9eb76201b0cdbf5063fb2f23fecd1738b
Merge: 3bb2a57 be370eb
Author: Kwang Kian Hui <2k1h297@gmail.com>
Date: Sat Aug 28 11:08:57 2021 +0800

Merge pull request #62 from sinhaugoh/documentations/retailer-and-product-comments

Added comments for retailer and product repository

commit 3bb2a5750d2725edad9145c66a92debe38ebf6f9
Merge: 670e2c2 80a41c8
Author: Kwang Kian Hui <2k1h297@gmail.com>
Date: Sat Aug 28 11:08:11 2021 +0800

Merge pull request #63 from sinhaugoh/improvement/textfield-length-control

Implement maxlength for form text fields

commit 80a41c8b6ac488f84a92a724fd771ac277e5657b
Author: Kwang Kian Hui <2k1h297@gmail.com>
Date: Fri Aug 27 15:23:02 2021 +0800

Implement maxlength for form text fields

Offstage() to hide counter text

commit cbb0248c5856e7b15fef119acb4d52c00e89ec4f
Author: Lawrence Ho <lawrencehsj@gmail.com>
Date: Thu Aug 26 18:33:02 2021 +0800

test toggle buttons

commit 8de66f6b2b7fa45133bee991040b7ea9373cf24e
Author: Lawrence Ho <lawrencehsj@gmail.com>
Date: Thu Aug 26 17:38:33 2021 +0800

revert to button

commit be370ebcdd26898fd70ddcae741588c4d540418b
Author: Kwang Kian Hui <2k1h297@gmail.com>
Date: Thu Aug 26 16:56:00 2021 +0800

Added comments for retailer and product repository

commit 54e9abf2034836d4671a55f4f4a3828774a25729
Author: Lawrence Ho <lawrencehsj@gmail.com>
Date: Thu Aug 26 15:03:34 2021 +0800

trying out checkboxlisttile

commit be734d2bd385870bfa0574acdf069a778b32774c
Author: Lawrence Ho <lawrencehsj@gmail.com>
Date: Wed Aug 25 17:15:57 2021 +0800

adjusted alignment and paddings

commit 670e2c2fa8b56c1a2212c9cd5ac0573a31e31c23
Merge: 1ebf434 384a11d
Author: Kwang Kian Hui <2k1h297@gmail.com>
Date: Wed Aug 25 15:50:50 2021 +0800

Merge pull request #60 from sinhaugoh/bug/fix-progress-indicator-display-saving-text-everytime

Fix SavingInProgressOverlay text issue

commit 384a11dc6f7bbd7d10a8b050d8c509f7d2ca5a47
Author: Kwang Kian Hui <2k1h297@gmail.com>
Date: Wed Aug 25 15:50:19 2021 +0800

resolve conflict errors from merging

commit 270dd9012b56661fd6b25fa7558bb44ff56e6066
Merge: af291af 1ebf434
Author: Kwang Kian Hui <2k1h297@gmail.com>
Date: Wed Aug 25 15:48:34 2021 +0800

Merge branch 'main' into bug/fix-progress-indicator-display-saving-text-everytime

commit 237522df83006beb55d22f94e1d966c677425a3c
Author: Lawrence Ho <lawrencehsj@gmail.com>
Date: Wed Aug 25 15:11:54 2021 +0800

update retailer drawer

commit 185761adcab203b297abb23ff20d8574bc4f113
Merge: e16eeac 1ebf434
Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Wed Aug 25 14:52:11 2021 +0800

Merge branch 'main' into junk/experiment-retailerhomepage

commit e16eeacd1be2e1aeb6d79a6e3d857a85caaafab4

Merge: ac8990c 0327a22

Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Wed Aug 25 14:49:20 2021 +0800

Merge branch 'main' into junk/experiment-retailerhomepage

commit 1ebf434ae637d59ce6c7d6be897df368f8cb277d

Merge: 0327a22 1683894

Author: lawrencehsj <58553029+lawrencehsj@users.noreply.github.com>

Date: Wed Aug 25 14:48:39 2021 +0800

Merge pull request #61 from sinhaugoh/feature/screenutils

Feature/screenutils

commit 16838941001454d2b36bc47ba51758568f3658fd

Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Wed Aug 25 14:44:11 2021 +0800

adjusted values in consumer home page

commit af291af9c4353336aae6bd012358e5ef420cbe25

Author: Kwang Kian Hui <2k1h297@gmail.com>

Date: Wed Aug 25 14:33:27 2021 +0800

update text for adding product

commit 0327a2219ed4eb40125c77645745a524dc7ffb2b

Merge: 92340d6 13d1c79

Author: sinhaugoh <shgoh025@mymail.sim.edu.sg>

Date: Wed Aug 25 14:09:36 2021 +0800

Merge pull request #59 from sinhaugoh/bug/fix-fav-retailer-list-page-not-getting-refreshed-automatically-when-connection-is-back

fix favourite retailer list pag enot getting refresh when the connect...

commit a0f9f988206d971a3ec5b8947ca2b243e324606d

Author: Kwang Kian Hui <2k1h297@gmail.com>

Date: Wed Aug 25 14:02:17 2021 +0800

remove unnecessary string interpolations

commit 4b734b0654bd9b6e6d27b8b9cfde8abacc130ee9

Author: Kwang Kian Hui <2k1h297@gmail.com>

Date: Wed Aug 25 13:46:02 2021 +0800

Fix SavingInProgressOverlay text issue

also removed unused imports and unnecessary 'flex: 1' as it is the default for Flexible widgets

commit 13d1c799dd10740346feedcef380c67315f6257d

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Wed Aug 25 13:19:42 2021 +0800

fix favourite retailer list pag enot getting refresh when the connection is back

commit 86cf037e4331d5e8aeccea5500da36fe1fb0b17

Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Wed Aug 25 13:13:18 2021 +0800

added screen utils

update sign in page values

commit 92340d6ab5ab037404d30c83a3bf220983c8e779

Merge: a2ed4e6 8e32173

Author: lawrencehsj <58553029+lawrencehsj@users.noreply.github.com>

Date: Wed Aug 25 12:50:53 2021 +0800

Merge pull request #58 from sinhaugoh/bug/consumer-homepage

added back route for favourites

commit 8e32173cca18d3d2e51aaa0468d84a536bb64695

Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Wed Aug 25 12:20:55 2021 +0800

added back shop to product route

commit ac8990c9244a591088aa02ef7242faef3b9ae816

Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Mon Aug 23 03:08:59 2021 +0800

edited custom card to accomodate product

commit c5a2c0d6ff52aa636f7d53fdd0d2364e969ad93

Author: Lawrence Ho <lawrenceehsj@gmail.com>

Date: Mon Aug 23 02:17:35 2021 +0800

removed extra icons in home page

replaced profile with favorites

commit 056025bc2c974089b24ecc83f75737615c597ffd

Author: Lawrence Ho <lawrenceehsj@gmail.com>

Date: Mon Aug 23 02:08:29 2021 +0800

added back route for favourites

removed profile route

removed email display

commit 83c60f4a4951e55fb841404766f9827ee93bc149

Author: Kwang Kian Hui <2k1h297@gmail.com>

Date: Sun Aug 22 18:47:40 2021 +0800

remove unit tests with error

commit a2ed4e6ee3fce098d9d36b5644e0215378d4fdc

Merge: db2310c e33e626

Author: sinhaugoh <shgoh025@mymail.sim.edu.sg>

Date: Sun Aug 22 17:41:32 2021 +0800

Merge pull request #57 from sinhaugoh/feature/unit-test-for-favourite-retailer-notifier

implement unit test for favourite retailer notifier

commit db2310c043e454fc16e5a9d3dce6583c824488d6

Merge: 05bba0d 6db766f

Author: sinhaugoh <shgoh025@mymail.sim.edu.sg>

Date: Sun Aug 22 17:41:11 2021 +0800

Merge pull request #56 from sinhaugoh/feature/unit-test-for-favourite-retailer-list-repository

implement unit tests for favourite retailer repository

commit e33e62614d2eaf6aede0e06901f240928117e218
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Sun Aug 22 17:22:21 2021 +0800

implement unit test for favourite retailer notifier

commit 6db766f5349c2636307533c986f13f26d5a941f9
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Sun Aug 22 15:58:11 2021 +0800

implement unit tests for favourite retailer repository

commit 05bba0de3b35410e2b0b38be00daee62bd42f6c1
Merge: 77037e4 d563ed1
Author: sinhaugoh <shgoh025@mymail.sim.edu.sg>
Date: Sun Aug 22 11:54:07 2021 +0800

Merge pull request #54 from sinhaugoh/feature/unit-test-for-product-list-notifier

implemented unit test for product list notifier

commit 77037e475e1471bedfe551c7158b2f44cf338358
Merge: 0923fce 0ae121e
Author: sinhaugoh <shgoh025@mymail.sim.edu.sg>
Date: Sun Aug 22 11:53:48 2021 +0800

Merge pull request #53 from sinhaugoh/feature/unit-test-for-product-repository

implement unit test for product repository

commit 0923fce55aa61ed97bcbb1bc6ba3a2fbf25a7a55
Merge: 9828bc4 86b07b6
Author: lawrencehsj <58553029+lawrencehsj@users.noreply.github.com>
Date: Sun Aug 22 09:44:12 2021 +0800

Merge pull request #42 from sinhaugoh/feature/consumer-homepage

Feature/consumer homepage

commit 86b07b674fae28e4cad106b25634c19307c8d4ea
Merge: 58d5d25 9828bc4
Author: Lawrence Ho <lawrencehsj@gmail.com>
Date: Sun Aug 22 09:43:08 2021 +0800

Merge branch 'main' into feature/consumer-homepage

commit 58d5d255043debdc614845bf15a43c43a2e03380
Author: Lawrence Ho <lawrencehsj@gmail.com>
Date: Sun Aug 22 09:42:42 2021 +0800

remove comments

commit 5a1d3bd40d0d570474595ab7cce9a2ef7f580736
Author: Lawrence Ho <lawrencehsj@gmail.com>
Date: Sun Aug 22 09:39:16 2021 +0800

update searchbarfilter

commit 8168fbb7c2812b1602203fff8408bf37718f4400
Merge: 1e9e21f 7a94ea9
Author: Lawrence Ho <lawrencehsj@gmail.com>
Date: Sun Aug 22 09:37:54 2021 +0800

Merge branch 'main' into feature/consumer-homepage

commit 1e9e21f2f57152c1b0b8e44cd09b43edb6491aab
Author: Lawrence Ho <lawrencehsj@gmail.com>
Date: Sun Aug 22 09:35:17 2021 +0800

prevent overflow

commit 9828bc47bc248c788f8f56a10f7135b80eeb8504
Merge: 7a94ea9 59a0938
Author: lawrencehsj <58553029+lawrencehsj@users.noreply.github.com>
Date: Sun Aug 22 09:27:29 2021 +0800

Merge pull request #38 from sinhaugoh/feature/drawer-widget

Feature/drawer widget

commit 59a09380d3cce400bdd4c12d70f4a2a27e04d0c6
Merge: 6571980 7a94ea9
Author: Lawrence Ho <lawrencehsj@gmail.com>
Date: Sun Aug 22 09:26:50 2021 +0800

Merge branch 'main' into feature/drawer-widget

commit 6571980f78217629c4206741d1153bbb345f6f3a

Author: Lawrence Ho <lawrencehsj@gmail.com>
Date: Sun Aug 22 09:25:01 2021 +0800

remove profile pic

commit f79cee92859af7c5ed1279487ec3f714597b2862
Author: Lawrence Ho <lawrencehsj@gmail.com>
Date: Sun Aug 22 08:29:10 2021 +0800

added const

commit 17316b98f59193d930b057ee9bbb9824e6b5adce
Author: Lawrence Ho <lawrencehsj@gmail.com>
Date: Sun Aug 22 08:26:14 2021 +0800

Update custom_card.dart

commit 48e509af54f0dc93a178206305401a3b32040594
Author: Lawrence Ho <lawrencehsj@gmail.com>
Date: Sun Aug 22 08:15:03 2021 +0800

add product

commit d563ed18a450598c0881a49ef63d3317c1bcfd71
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Sun Aug 22 00:28:56 2021 +0800

implemented unit test for product list notifier

commit 0ae121e49e1ef5d904dde324bca14ba3fcce2cd8
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Sat Aug 21 22:10:56 2021 +0800

implement unit test for product repository

commit 8edf006ac2073fa59afe8f4a5bdbeba8cd3a0df1
Author: Kwang Kian Hui <2k1h297@gmail.com>
Date: Sat Aug 21 19:06:31 2021 +0800

implement partial unit test for add product form notifier

commit 57a0bb881d3953971004190023e1d8797cd46d36
Author: Kwang Kian Hui <2k1h297@gmail.com>
Date: Sat Aug 21 19:05:01 2021 +0800

replace not_found to not-found (error code)

commit b7b712957b49d36384d95935c8c1fa229635850c

Merge: 80dded7 7a94ea9

Author: Kwang Kian Hui <2k1h297@gmail.com>

Date: Sat Aug 21 11:24:14 2021 +0800

Merge branch 'main' into feature/add-on-unit-testing-to-retailer-list

commit 7a94ea9b44a1f5557edf00e09cd7c7986cebaee1

Merge: 5afc0e8 3879381

Author: sinhaugoh <shgoh025@mymail.sim.edu.sg>

Date: Fri Aug 20 22:52:35 2021 +0800

Merge pull request #51 from sinhaugoh/feature/unit-testing-for-all-value-validator

implement unit test for all value validator

commit 5afc0e89e35028af9321470b2081443c344a2c78

Merge: 33078dc bef526b

Author: sinhaugoh <shgoh025@mymail.sim.edu.sg>

Date: Fri Aug 20 22:52:14 2021 +0800

Merge pull request #52 from sinhaugoh/feature/unit-testing-for-retailer-repository

implement unit tests for retailer repository

commit bef526b84181ceec4ee1544de0cf84d33f0d9a6b

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Fri Aug 20 22:23:45 2021 +0800

implement unit tests for retailer repository

commit 3879381f24cf9068244e4fce4120648edfdb0637

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Fri Aug 20 21:00:00 2021 +0800

implement unit test for all value validator

commit 80dded786d8d57a522c5cf8adf0b367b5ed04e5f

Author: Kwang Kian Hui <2k1h297@gmail.com>

Date: Thu Aug 19 15:49:47 2021 +0800

update retailer_list test and added add_product_form testing

commit d747231956c2f80b2fbf422aa1d1d2cabdf69bdc

Author: Kwang Kian Hui <2k1h297@gmail.com>

Date: Wed Aug 18 16:02:12 2021 +0800

resolve errors after merging from main

commit 33078dcf6074edcaee5118db22b5172a4926d5ee

Merge: 2d67f70 a4600be

Author: sinhaugoh <shgoh025@mymail.sim.edu.sg>

Date: Wed Aug 18 11:10:51 2021 +0800

Merge pull request #50 from sinhaugoh/improvement/trim-text-fields

enable auto trimming for some text field

commit a4600be80e7be06dcd5af2a17c04c7590b6ed498

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Tue Aug 17 21:29:45 2021 +0800

enable auto trimming for some text field

commit e5543b8ec352560feeb2b9df05b29ef6ef7e4d02

Merge: ea2a2eb 2d67f70

Author: Kwang Kian Hui <2k1h297@gmail.com>

Date: Tue Aug 17 11:17:38 2021 +0800

Merge branch 'main' into feature/add-on-unit-testing-to-retailer-list

commit 2d67f7033a19314d5ea71e09c78cc3c1cfa6e793

Merge: 448819f 153adfe

Author: sinhaugoh <shgoh025@mymail.sim.edu.sg>

Date: Mon Aug 16 23:45:43 2021 +0800

Merge pull request #49 from sinhaugoh/feature/favourite-or-unfavourite-a-retailer-shop

Feature/favourite or unfavourite a retailer shop

commit 153adfeb7c0ae0c01244dfa945aa554ac0f3db42

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Mon Aug 16 23:38:28 2021 +0800

implement UI for favourite page

commit 5744bfea609a392d94e13b6c34b92b7d43f6a9

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Mon Aug 16 23:33:21 2021 +0800

implement toggle favourite button in consumer home page

commit 80a2cbad14d68ffe32bf6d928563e2703e28088d

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Mon Aug 16 23:32:47 2021 +0800

implement toggle favourite retailer logic in application layer

commit a1c7cac2d224d9360cc8f8ba6dd4e172d6e31014

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Mon Aug 16 23:32:19 2021 +0800

implement toggle favourite retailer logic in infrastructure

commit 5a57e073543bd161711b3c4d267ab111a8f5728f

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Mon Aug 16 18:05:06 2021 +0800

modify code so that the performance of the app improve

commit 05505ad96bff936d79860e1fbacd23ec3d4e0646

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Mon Aug 16 17:41:09 2021 +0800

restructure consumer folder

commit c02836e0c9b0349bf1c2db29be54a8f3a8907adc

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Mon Aug 16 17:23:00 2021 +0800

initialise favourite retailer in consumer home page instead of favourite page itself

commit 448819f780a93259f08db003b19d07c198c972af

Merge: 2e01f5f fd06702

Author: sinhaugoh <shgoh025@mymail.sim.edu.sg>

Date: Mon Aug 16 17:09:53 2021 +0800

Merge pull request #48 from sinhaugoh/bug/fix-favourite-retailer-page-bug

fix favourite retailer page after adding id field to retailer class

commit fd067028a661629b20d9c3ef73ff166941474146

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Mon Aug 16 17:09:28 2021 +0800

fix favourite retailer page after adding id field to retailer class

commit 2e01f5f03808ff75dd53a3c10e5e4a058791a203

Merge: ca74ddc 58ca333

Author: Kwang Kian Hui <2k1h297@gmail.com>

Date: Mon Aug 16 15:54:50 2021 +0800

Merge pull request #44 from sinhaugoh/feature/consumer/shop-detail

Feature/consumer/shop detail

commit 58ca3333b5d9c667699185cb517697907ea17c85

Author: Kwang Kian Hui <2k1h297@gmail.com>

Date: Mon Aug 16 15:52:57 2021 +0800

remove snack bar for no connection

commit 4df03193704fbc737326f7e75845bd2fa3c93deb

Author: Kwang Kian Hui <2k1h297@gmail.com>

Date: Mon Aug 16 15:10:38 2021 +0800

remove unused imports

commit 684c2fde3800ba77040eefe1db557f3fc9fb7135

Merge: 02f8f32 ca74ddc

Author: Kwang Kian Hui <2k1h297@gmail.com>

Date: Mon Aug 16 15:09:09 2021 +0800

Merge branch 'main' into feature/consumer/shop-detail

commit 02f8f32fee9491c67ee4098ab7a80a863aa5ec69

Author: Kwang Kian Hui <2k1h297@gmail.com>

Date: Mon Aug 16 15:08:51 2021 +0800

fixed bug where refresh of product list is not working

wrong usage of provider, changed to correct one

commit ca74ddc0b32e19960761eae9f1a7ab004e447eb9
Merge: 88d4d24 717b623
Author: sinhaugoh <shgoh025@mymail.sim.edu.sg>
Date: Mon Aug 16 14:50:16 2021 +0800

Merge pull request #47 from sinhaugoh/bug/retailer-image-not-getting-deleted-properly

Bug/retailer image not getting deleted properly

commit 717b6236d6dad360732d25f038df0fd58d3983af
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Mon Aug 16 13:26:52 2021 +0800

fix retailer image error on edit profile page when the retailer has no image initially

commit 7d0e246d55fe935c5eecda8fdd4564185c92b456
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Mon Aug 16 13:26:07 2021 +0800

fix retailer image not deleted properly

commit 6871ac68af70505fb55c5ab24cb7ac60162301c3
Author: Kwang Kian Hui <2k1h297@gmail.com>
Date: Sun Aug 15 17:11:51 2021 +0800

Changed usage of retailer uen to id for retrieving product list

commit 4f8782af755aaaf322cf1b0d23df3893b82c658c
Merge: 3e70b35 88d4d24
Author: Kwang Kian Hui <2k1h297@gmail.com>
Date: Sun Aug 15 16:56:40 2021 +0800

Merge branch 'main' into feature/consumer/shop-detail

commit 88d4d24a8998f621e348ce8a9cba8b104d6b54b2
Merge: 9a18a09 67036a4
Author: Kwang Kian Hui <2k1h297@gmail.com>
Date: Sun Aug 15 15:11:03 2021 +0800

Merge pull request #46 from sinhaugoh/improvement/add-id-field-to-retailer-and-it's-dto

implemented id field for retailer and fix all errors for it

commit 67036a46fadfc5d6cb6a71dceebd9b8ead1dc05

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Sun Aug 15 14:48:54 2021 +0800

implemented id field for retailer and fix all errors for it

commit 3e70b352aebabc9baac9be99de709aed22bb7b86
Author: Kwang Kian Hui <2k1h297@gmail.com>
Date: Sun Aug 15 13:57:35 2021 +0800

Implement internet connectivity checker for productlist

commit e36de6453bd75a0489f4d3afeef21c5f8b62b0bc
Author: Kwang Kian Hui <2k1h297@gmail.com>
Date: Sun Aug 15 13:18:44 2021 +0800

Implement refresh & noConnection logic

commit 9a18a09c424ff8e9ffd118f06992b810f0fc9872
Merge: 886b795 055a300
Author: sinhaugoh <shgoh025@mymail.sim.edu.sg>
Date: Sun Aug 15 12:50:19 2021 +0800

Merge pull request #45 from sinhaugoh/improvement/toggle-password-visibility

implement password toggle visibility for sign in and sign up for con...

commit ea5294a621085feae1011c701f5d09706ce01848
Author: Kwang Kian Hui <2k1h297@gmail.com>
Date: Sun Aug 15 12:22:46 2021 +0800

Merge main to current branch and resolve conflicts

commit 46364237a148563ad93d00735ff7dcd9c9b4314d
Merge: a3afb2d 886b795
Author: Kwang Kian Hui <2k1h297@gmail.com>
Date: Sun Aug 15 12:08:37 2021 +0800

Merge branch 'main' into feature/consumer/shop-detail

commit 055a300ca3069c49d14a93970381949ddbe15c6b
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Sun Aug 15 00:07:13 2021 +0800

implement password toggle visibility for sign in and sign up for consumer and retailer

commit 886b795d204cf19ed0486fa3be93ce909f2f6048
Merge: b04c97c 33d9afc
Author: sinhaugoh <shgoh025@mymail.sim.edu.sg>
Date: Sat Aug 14 20:06:08 2021 +0800

Merge pull request #43 from sinhaugoh/feature/display-favourite-retailers

Feature/display favourite retailers

commit a3afb2d97ae697484d66a586d3a0f4e068dc7704
Author: Kwang Kian Hui <2k1h297@gmail.com>
Date: Sat Aug 14 19:27:15 2021 +0800

Implement basic UI for consumer - retailer detail page

commit 33d9afc5468af47b0c2fa05357426ffdde9c7fa7
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Sat Aug 14 18:29:08 2021 +0800

implement routing and simple UI for favourite retailer list

commit a589769d075a7959e2d7d687f98eed213d3c374
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Sat Aug 14 18:28:41 2021 +0800

implement application logic for the favourite retailer logic

commit 1f523d2ec40d6ecdf934fe3234b36a580ed71fab
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Sat Aug 14 18:28:05 2021 +0800

implement retrieval of favourite retailer list in infrastructure

commit d16680abb7fc5d7b87bb498dcd9495dbdce918ba
Author: Kwang Kian Hui <2k1h297@gmail.com>
Date: Sat Aug 14 15:31:21 2021 +0800

Add consumer - retailer detail page

commit 2e94c1f59027689cef6c7330c383634dc0cd3514
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Sat Aug 14 14:29:14 2021 +0800

change the syntax to the stream subscription to be nullable

commit 2f3c59f2d524b5c77cf4ffaf132a09747b086b4e

Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Thu Aug 12 22:33:39 2021 +0800

commented out productlistview items in custom card

commit be201a7ec24fa4174a8ba5b2bf6eb8c34efff3a5

Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Thu Aug 12 22:25:49 2021 +0800

added productlistview to customcard

commit e09194e195691de169dcc1411aed0ca0db8e2fd6

Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Thu Aug 12 19:10:49 2021 +0800

prevent searchbar from resizing

commit af20c0e6b9a2bec1010ba366a38b0c344c2d58e0

Author: Kwang Kian Hui <2k1h297@gmail.com>

Date: Thu Aug 12 15:19:07 2021 +0800

Implement get product list logic

commit 516f4c3b469f05eb103671b18e99fd4fce37e287

Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Thu Aug 12 13:46:53 2021 +0800

deconflict searchbar

commit d6faf81e2f98a6ad9f1825cd87824e9bd81f6527

Merge: eaed724 b04c97c

Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Thu Aug 12 13:46:37 2021 +0800

Merge branch 'main' into junk/experiment-consumerhomepage-6-mergetest

commit fb27234ea79a02f77d82119c10cfb8a0022189f3

Author: Kwang Kian Hui <2k1h297@gmail.com>

Date: Thu Aug 12 12:22:04 2021 +0800

Implement consumer product list page

commit ffb0d7fb15d0c8eab9a2f0aa7acf38f4ffa50c21
Merge: 2de7c04 b04c97c
Author: Kwang Kian Hui <2k1h297@gmail.com>
Date: Thu Aug 12 10:43:50 2021 +0800

Merge branch 'main' into feature/consumer/shop-detail

commit 2de7c049c74b2fe48c5c33289586da03dbf86ed4
Author: Kwang Kian Hui <2k1h297@gmail.com>
Date: Thu Aug 12 10:42:10 2021 +0800

Add ConsumerProductList route

commit eaed724e1c61a1ab78b3b38ab845d163925212dc
Author: Lawrence Ho <lawrencehsj@gmail.com>
Date: Thu Aug 12 03:51:38 2021 +0800

updated class variables

commit bedeece2a77e6293a16995724b302582657a2f09
Author: Lawrence Ho <lawrencehsj@gmail.com>
Date: Thu Aug 12 03:45:05 2021 +0800

bump

commit 1a9d284be208482772a5d7f32ed012a32759b4ff
Author: Lawrence Ho <lawrencehsj@gmail.com>
Date: Wed Aug 11 23:35:02 2021 +0800

test fonts

commit b04c97caee974540b20941417fc887a1eb9d25d5
Merge: 5b6f4d4 2968608
Author: sinhaugoh <shgoh025@mymail.sim.edu.sg>
Date: Wed Aug 11 23:22:24 2021 +0800

Merge pull request #41 from sinhaugoh/feature/search-function

Feature/search function

commit ac9274609483fee11611c9b81b2a66c334390f77
Author: Lawrence Ho <lawrencehsj@gmail.com>
Date: Wed Aug 11 23:05:58 2021 +0800

solved network image issues

commit 29686081e23e7c3c175967890db01c793180fa4e

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Wed Aug 11 21:40:19 2021 +0800

implement search function logic in presentation layer

commit 4dc0e23ddf5b500386c45d1afd122e405d9d60ef

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Wed Aug 11 21:40:03 2021 +0800

implement search function logic

commit 6c42c9f4138a5163bd48802ba7343a5939fb77d9

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Wed Aug 11 21:38:48 2021 +0800

remove filter in database

commit 02066cd2b2c4b067c73a736455a00ff132f825eb

Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Wed Aug 11 18:42:40 2021 +0800

added image-not-found to assets

commit 99fc4392dadb5880fbe911bd2386bb727528f82b

Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Wed Aug 11 18:31:27 2021 +0800

seperate filter icon from search bar

commit 5b6f4d4c04622043fa738c180eb863c283a4bf8e

Merge: 10a0a21 12f05e0

Author: sinhaugoh <shgoh025@mymail.sim.edu.sg>

Date: Wed Aug 11 17:05:32 2021 +0800

Merge pull request #40 from sinhaugoh/feature/toggle-all-product-feature

Feature/toggle all product feature

commit 12f05e08fe5bcfc5c6abb2bc185e6b0aeb51c8ce

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Wed Aug 11 17:00:15 2021 +0800

implement toggle on/off logic in presentation layer

commit 181ca62efabe45473edf1060442d46a6f489d5ef

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Wed Aug 11 17:00:02 2021 +0800

implement toggle on/off logic for product in application layer

commit 9e1049bd6b37c92ceb02715d7634ffce6bf287f1

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Wed Aug 11 16:59:37 2021 +0800

implement updateProductList logic in infrastructure layer

commit cafb5ec7a5ee10ebaf7b97630e19937d077e4a46

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Wed Aug 11 16:59:10 2021 +0800

remove dataModified field for ProductDTO

commit 98e021e7e00355f3e784223164c004116b273148

Author: Kwang Kian Hui <2k1h297@gmail.com>

Date: Wed Aug 11 13:25:56 2021 +0800

merge main to current and resolve conflicts

commit f8d514fbe989e0380396b999aecfedd888306df9

Merge: 4a8db42 10a0a21

Author: Kwang Kian Hui <2k1h297@gmail.com>

Date: Wed Aug 11 13:11:07 2021 +0800

Merge branch 'main' into feature/consumer/shop-detail

commit 10a0a21a75742ece4f8d2dcc478e95e04f20b226

Merge: c733654 4b840f2

Author: sinhaugoh <shgoh025@mymail.sim.edu.sg>

Date: Wed Aug 11 13:05:41 2021 +0800

Merge pull request #39 from sinhaugoh/feature/update-and-delete-product

Feature/update and delete product

commit 4b840f28bc33fb43cadcb4a4bea0a8454f661c73
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Wed Aug 11 13:02:37 2021 +0800

remove unused import

commit 5d4150c977fa40c0d666aac5b7f9e2cdccb45c2c
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Wed Aug 11 13:02:11 2021 +0800

implement delete product function

commit eaba9bb4f04faf6551aad87e4e64ca48f53fb1cb
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Wed Aug 11 13:01:55 2021 +0800

remove comment

commit b76a22656331771905ef7061c34a905c93d6454a
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Wed Aug 11 13:01:38 2021 +0800

fix the drawer in retailer profile page

commit 4a8db42fed86bee9d03db32f206a46312402e839
Merge: fda8a5c c733654
Author: Kwang Kian Hui <2k1h297@gmail.com>
Date: Wed Aug 11 12:52:58 2021 +0800

Merge branch 'main' into feature/consumer/shop-detail

commit 766000aa9372c6ffc0d0a997eefe145e791e73e9
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Wed Aug 11 02:09:36 2021 +0800

display loading indicator when loding image in edit product form

commit 4a6c2cbed91c784128ad57b2c52f9838c84d7661
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Wed Aug 11 02:03:39 2021 +0800

improve add product logic (mainly prices and state)

commit 835d4f9eae5880a832fc001dc9b7b061f93e341e

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Wed Aug 11 02:02:43 2021 +0800

implement edit product in presentation layer

commit 8a59154b3a7ce63eb06356819ef19868f2c39014
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Wed Aug 11 02:01:57 2021 +0800

improve validator for usual and discounted price

commit 8f97e66e3f4f05024740d4b0ddc04ac6c8fbb4c2
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Wed Aug 11 02:01:23 2021 +0800

implement edit product notifier in application layer

commit 3affae9b39438566ceca61d14f6b6af62b0fecd5
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Wed Aug 11 01:59:44 2021 +0800

implement update and delete product logic in infrastructure layer

commit 1b4823c6e07ba23148601612f967ead4b20c0ed1
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Wed Aug 11 01:59:02 2021 +0800

remove unnecessary code

commit 91b9a9299d9a687997b33dcb859fb94b61381e9d
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Wed Aug 11 01:57:47 2021 +0800

implement initial factory for product entity

commit c7049184d3ff6b931b8b38c9db8a6903f236379e
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Wed Aug 11 01:57:18 2021 +0800

remove unused package

commit fda8a5caec65a12e602d3804e4d4b2f2f1d5e4a0
Author: Kwang Kian Hui <2k1h297@gmail.com>
Date: Wed Aug 11 00:48:12 2021 +0800

implement getProductList logic

switched from retailer id to uen of retailer

commit ebc251a2e4e9d87b1ee588e3ee75084dedbf31bb

Author: Kwang Kian Hui <2k1h297@gmail.com>

Date: Wed Aug 11 00:22:12 2021 +0800

implement Future getProductList() for shop detail

commit bc5caee768664c9f378f845a2b31a1b8f8bd7571

Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Tue Aug 10 23:22:03 2021 +0800

added custom card class for modular code

commit 38cb35e85d503696ea5820a9baa935256a6b82c2

Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Tue Aug 10 22:08:44 2021 +0800

remove unnecessary variables

commit 6d60892f2da41512ef3bb1bd6a27b459778b8321

Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Tue Aug 10 19:39:36 2021 +0800

added decoration options

commit c733654cb6ef9264c8347aac855cd8ff9555b80d

Merge: 58447ba e7e6d01

Author: Kwang Kian Hui <2k1h297@gmail.com>

Date: Tue Aug 10 19:30:59 2021 +0800

Merge pull request #37 from sinhaugoh/bug/add-product-feature-bug-fixes

Implement bug fixes for add product feature

commit e7e6d0155418b9080c66087aeda0fb2524290b9e

Author: Kwang Kian Hui <2k1h297@gmail.com>

Date: Tue Aug 10 19:25:13 2021 +0800

Combine hasFailureUploadingImage and hasFirebaseFailure state

commit 21ce0bb1d235759e1c0ffe9a2ef6ec0977bf3ecb
Author: Kwang Kian Hui <2k1h297@gmail.com>
Date: Tue Aug 10 18:49:14 2021 +0800

Implement bug fix

fixed bug where stuck on saving state when "add product" button was pressed while on airplane mode.

commit 8e37d32068ee3c14b605f806e6c84d600195fce7
Author: Lawrence Ho <lawrencehsj@gmail.com>
Date: Tue Aug 10 18:26:38 2021 +0800

update drawer widget UI

commit b4c0932a27e57b3db49d443592208a4f16e60618
Author: Lawrence Ho <lawrencehsj@gmail.com>
Date: Tue Aug 10 18:02:07 2021 +0800

added white logo text, update appbar UI

commit ccb6e47d14bdd871be47922a471a9eee681eceaa
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Tue Aug 10 15:59:52 2021 +0800

disable user from navigating to edit profile form when no connection

commit c6a82aa2e7d035f90e9a5a0e67fda0802a757dc8
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Tue Aug 10 15:59:25 2021 +0800

fix the infinite saving overlay when no connection

commit 8deed0476a08e36aecdb2c3d780864ffaabe1732
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Tue Aug 10 15:58:52 2021 +0800

implement connection checking in infrastructure layer of image picking logic

commit 0f2f52c44b66008a92c74d02d4b7e21fc9202e61
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Tue Aug 10 15:58:03 2021 +0800

add in no connection failure for image picking repo

commit b0158ceb94c4898e4ef3376ae6c705d831e85ee7

Author: Kwang Kian Hui <2k1h297@gmail.com>

Date: Tue Aug 10 15:12:30 2021 +0800

Implement bug fix

Moved controllers to retailer_home_page so that value does not get cleared while form is saving. Once the form saves successfully, value is cleared.

commit 9f8d878a4d0975b920472a1c3f6d9bef3ffd17d8

Author: Kwang Kian Hui <2k1h297@gmail.com>

Date: Tue Aug 10 14:51:39 2021 +0800

Implement bug fix

Changed default availability to false.

Fixed bug where usual and discounted price value displays 0.00 while saving.

commit c22b9af9b7b0020afc5c84bdce5108c90a177580

Merge: 5156d78 58447ba

Author: Kwang Kian Hui <2k1h297@gmail.com>

Date: Tue Aug 10 13:58:45 2021 +0800

Merge branch 'main' into bug/add-product-feature-bug-fixes

commit bae40d17321b6a3196d7b7e8ec316c69abddf426

Author: Kwang Kian Hui <2k1h297@gmail.com>

Date: Tue Aug 10 13:58:34 2021 +0800

Add retailer detail page and route

commit e2105c4703a011773806e1a29e1251bbfe6c917c

Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Tue Aug 10 13:28:52 2021 +0800

added logotext in appbar

commit 61fb78d0b2f8028a7d46dba5270cc292c520631a

Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Tue Aug 10 13:17:09 2021 +0800

added placeholder image

commit fc7d82acbb04bfc0b7954e5cf194d4b37377f846
Author: Lawrence Ho <lawrencehsj@gmail.com>
Date: Tue Aug 10 12:41:49 2021 +0800

using decorationimage

commit 87f850fbfe1689562bed69a199e00ef713e0cc4
Author: Kwang Kian Hui <2k1h297@gmail.com>
Date: Tue Aug 10 11:29:20 2021 +0800

Optimise consumer-retailer-list

use of provider scope to allow each RetailerItem() to be a const

commit 3155593fae313ca6cf428620dc2b157d303149ac
Author: Lawrence Ho <lawrencehsj@gmail.com>
Date: Tue Aug 10 01:32:29 2021 +0800

added original search and filter

commit 5156d78372f05d2f415f17c9f7c2555559632489
Author: Kwang Kian Hui <2k1h297@gmail.com>
Date: Tue Aug 10 00:56:45 2021 +0800

Update bug fixes

stuck on saving state when no internet found initially due to wrong placement of codes moved to correct part of it

commit da2dc94cc49557378ebdda81964eb8519587e987
Author: Kwang Kian Hui <2k1h297@gmail.com>
Date: Tue Aug 10 00:04:25 2021 +0800

Implement bug fixes for add product feature

commit 43339e9c4a15eac5a0716ea31ca47c479b73b3eb
Author: Lawrence Ho <lawrencehsj@gmail.com>
Date: Tue Aug 10 00:00:46 2021 +0800

added image from url

commit 58447baab259e3c6a312072521a8153c67a17e67
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Mon Aug 9 23:50:52 2021 +0800

remove unused imports

commit f2198da4e04285568caaa72b7cb57d6f78cf21c8

Merge: 3bc7e2d 35d0f49

Author: sinhaugoh <shgoh025@mymail.sim.edu.sg>

Date: Mon Aug 9 23:49:13 2021 +0800

Merge pull request #36 from sinhaugoh/feature/toggle-product-visibility

Feature/toggle product visibility

commit 35d0f496a5eb64115391ce952e711a7f16d87c6a

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Mon Aug 9 23:43:13 2021 +0800

Update retailer_home_page.dart

commit e0933bb2720bcda3572af00ede678cec27ffe4d5

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Mon Aug 9 23:43:08 2021 +0800

implement product toggle button functionality into presentation layer

commit e693f3b34051c29b4e728fd59bff53929bb3e0f1

Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Mon Aug 9 23:43:06 2021 +0800

shift search bar and filter to appbar

commit edbc9510b7f2c8066aff8bf3fae9433d923b345e

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Mon Aug 9 23:42:35 2021 +0800

implement update product logic in application layer

commit 2f6beefa39ca375e5b8c15c3ecece2e0b81729f0

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Mon Aug 9 23:41:49 2021 +0800

implement update product logic in infrastructure layer

commit 3899c8eaa555f0e5e6ca9d1f2bb3155af294ee0f

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Mon Aug 9 23:40:27 2021 +0800

rename product list repository to product repository

commit 97d3b7fddbf36f64bd2f134caf312a54cd55d82a

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Mon Aug 9 23:39:30 2021 +0800

restructure the files

commit 1c63822ab894d8c71df2d80e51d7fca536fd8bc1

Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Mon Aug 9 23:31:47 2021 +0800

update appbar UI

commit 439995d52dc7539e34f953eac72fa22f6eecd35f

Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Mon Aug 9 23:05:44 2021 +0800

update card UI

commit 3bc7e2d121655341e243c72f15d082581a6dbe20

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Mon Aug 9 22:16:17 2021 +0800

restructure files

commit f99094901449efed17fabf9fa8a59714620ebc5e

Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Mon Aug 9 21:26:00 2021 +0800

adjust alignments

commit 806f7f7bdc53e455491cdcb1867deadd1a610358

Merge: 6ebc253 52e0dd4

Author: sinhaugoh <shgoh025@mymail.sim.edu.sg>

Date: Mon Aug 9 21:17:05 2021 +0800

Merge pull request #35 from sinhaugoh/feature/display-retailer-product-list

Feature/display retailer product list

commit 52e0dd4835fc4973fb3c7b00c1f68cb2e095d634

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Mon Aug 9 21:15:50 2021 +0800

improve performance of product list view

commit 84c7cf82377838425aa6162df1eef6d09c9dba4d
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Mon Aug 9 20:41:30 2021 +0800

move generateProductId function to remote service layer

commit 388dd0e64a436ff1c463d2be930ad9d50fd93333
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Mon Aug 9 20:39:53 2021 +0800

implement display of product stream in retailer home page

commit cb4e21418188f265acda4a3541645515495b4a63
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Mon Aug 9 20:38:27 2021 +0800

implement get product stream in application layer

commit 0d4e767935461f9520f0f19d27e5bb4cfb15e9bd
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Mon Aug 9 20:38:06 2021 +0800

implement get product stream logic in infrastructure layer

commit f551a6bf75927ecb472a5f3ee6f266e46b4201be
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Mon Aug 9 20:37:26 2021 +0800

import rxdata to provide more control on stream

commit 6ebc25344307b13671dc003a21cc79567dfbf221
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Mon Aug 9 19:47:14 2021 +0800

remove comment

commit 0b1d4618b88f7d0a4dcb2a37229d179198daf98d
Merge: 6bbd685 3ebb50c
Author: Kwang Kian Hui <2k1h297@gmail.com>

Date: Mon Aug 9 19:37:37 2021 +0800

Merge pull request #29 from sinhaugoh/feature/retailer/add-product

Feature/retailer/add product

commit 3ebb50c597839dcc9502f1ce609232738522554a

Author: Kwang Kian Hui <2k1h297@gmail.com>

Date: Mon Aug 9 16:20:33 2021 +0800

Fixed validation check

commit 4b9a80951fa2ce499ac25ee98377a43e337ee592

Author: Kwang Kian Hui <2k1h297@gmail.com>

Date: Mon Aug 9 16:08:09 2021 +0800

Changed validation check for usual and discounted price

commit 50cd282379d3c3a7efc5d5e31fd0dc81d4777903

Author: Kwang Kian Hui <2k1h297@gmail.com>

Date: Mon Aug 9 15:45:53 2021 +0800

removed unused codes and other changes

add ignore statements and remove unused imports and code line

commit 0860f38b6fba3a00120f54e744f252fed786b971

Merge: baa97f4 6bbd685

Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Mon Aug 9 15:44:49 2021 +0800

Merge branch 'main' into junk/experiment-consumerhomepage-2

commit 2d7fd9dc45493f927c31141ec015a395d32dad73

Merge: 6f690e4 6bbd685

Author: Kwang Kian Hui <2k1h297@gmail.com>

Date: Mon Aug 9 15:42:42 2021 +0800

Merge branch 'main' into feature/retailer/add-product

commit 6f690e413ff9854464811a54ec94ec002f9bb9b1

Author: Kwang Kian Hui <2k1h297@gmail.com>

Date: Mon Aug 9 15:42:14 2021 +0800

Changes made following requested changes in review

commit 6bbd685f3a52788e4a77c4855300dd4b62ed443c

Merge: 313c9cc 08d0139

Author: lawrencehsj <58553029+lawrencehsj@users.noreply.github.com>

Date: Mon Aug 9 15:41:46 2021 +0800

Merge pull request #34 from sinhaugoh/feature/welcome&splashpage

remove problems

commit 08d01395456c76c4fa7c7c713fc54f43dcb46670

Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Mon Aug 9 15:41:04 2021 +0800

remove problems

commit baa97f4e6fb9cb0bf1be8c372c3e8c379cf0ca74

Merge: 00d2577 313c9cc

Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Mon Aug 9 15:38:38 2021 +0800

Merge branch 'main' into junk/experiment-consumerhomepage-2

commit 313c9ccf38b500ff7bcf23f6f70f0f975477cc4d

Merge: eefc6ea f97fa26

Author: lawrencehsj <58553029+lawrencehsj@users.noreply.github.com>

Date: Mon Aug 9 15:37:56 2021 +0800

Merge pull request #33 from sinhaugoh/feature/welcome&splashpage

update splash_page and welcomeroute

commit f97fa26c16148f7a72d2a1ef80c2aee492edb33b

Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Mon Aug 9 15:35:22 2021 +0800

update splash_page and welcomeroute

commit eefc6ea8203369869c0ec07df73f95abd913a1d8

Merge: c92ba77 c80481e

Author: lawrencehsj <58553029+lawrencehsj@users.noreply.github.com>

Date: Mon Aug 9 15:26:32 2021 +0800

Merge pull request #32 from sinhaugoh/feature/signuppage-consumer

Feature/signuppage consumer

commit c80481eae0bede6b42116d31f9cbc9e8d7124534

Merge: c8f0dc8 c92ba77

Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Mon Aug 9 15:23:03 2021 +0800

Merge branch 'main' into junk/experiment-signuppage-consumer

commit c92ba775e33a7dad537bc303a562f00514c4dc20

Merge: 70192bd 52977d5

Author: lawrencehsj <58553029+lawrencehsj@users.noreply.github.com>

Date: Mon Aug 9 15:20:08 2021 +0800

Merge pull request #31 from sinhaugoh/feature/login&signup-page

Feature/login&signup page

commit 00d2577df4d0e399718d4769b31d773bc1562adf

Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Mon Aug 9 15:10:14 2021 +0800

update to card

commit 87177b88fe65266f7b4f4ad679b8d8cbe1d6565f

Merge: 299da1c 70192bd

Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Mon Aug 9 13:44:58 2021 +0800

Merge branch 'main' into junk/experiment-consumerhomepage

commit c8f0dc85ea994286000cb65afe0deac9c082522d

Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Mon Aug 9 13:25:29 2021 +0800

update consumer sign up page UI

commit 7ee45ebfa518f2d30a0337a502dfd2e635236db5

Merge: 28db57f 70192bd

Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Mon Aug 9 13:04:32 2021 +0800

Merge branch 'main' into junk/experiment-signuppage-consumer

commit 52977d593928909dd5446edf0dc47169c7b3551a

Merge: b9cf859 70192bd

Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Mon Aug 9 13:00:03 2021 +0800

Merge branch 'main' into feature/login&signup-page

commit b9cf85972407a27bf81c64e8eddeff146defe082

Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Mon Aug 9 12:56:51 2021 +0800

undo changes to app_widget

changed back to welcomeroute

commit e351a185844082376f4ff72662b8da587f0c3181

Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Mon Aug 9 12:54:54 2021 +0800

added const keywords

commit ab72edc3a1fb14677ddc097c7abb4a7a5ba497a5

Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Mon Aug 9 12:42:58 2021 +0800

readjusted routes for new UI

commit 10528e6b585b4308ca17ac2993382bfe798499e1

Author: Kwang Kian Hui <2k1h297@gmail.com>

Date: Sun Aug 8 13:16:56 2021 +0800

Changed provider file based on feedback

Moved product list provider to retailer/shared/provider.dart

commit 4af3291a2a9731e90bb262dc72c2a7a761c78eb3

Author: Kwang Kian Hui <2k1h297@gmail.com>

Date: Sun Aug 8 13:12:35 2021 +0800

Implement validator and other changes

Added ValueFailure.invalidPriceValue() for usual and discounted price.

Removed some firestore failures.
fixed bug where products with no image does not get added.

commit 08f045b9c140b0b9e629526df06ec005fbead531
Author: Kwang Kian Hui <2k1h297@gmail.com>
Date: Sun Aug 8 11:42:48 2021 +0800

resolve conflicts and file path changes

Merged main branch to this branch. Resolve path and other conflicts

commit 637b9888cebebced63753c0a1b6beebe98f0cedc
Merge: 72d9eb8 70192bd
Author: Kwang Kian Hui <2k1h297@gmail.com>
Date: Sun Aug 8 11:36:56 2021 +0800

Merge branch 'main' into feature/retailer/add-product

commit 72d9eb890971378ea07c165415eb2865b0ff4d8a
Author: Kwang Kian Hui <2k1h297@gmail.com>
Date: Sun Aug 8 11:34:12 2021 +0800

Restructure add product logic

Improved image picking and restructure codes. Removed duplicate add product route

commit 44f2d372cba7849d7d6a67785d5352e3562df227
Author: Lawrence Ho <lawrencehsj@gmail.com>
Date: Sun Aug 8 00:45:57 2021 +0800

Update sign_in_page.dart

commit 70192bd8388b7b7980ef21dc9728cd782ac4e8b2
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Sat Aug 7 23:22:26 2021 +0800

change the title of the retailer edit profile page

commit 58a51a891db9b8d13fdbd9a41bf6827d7d81ff43
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Sat Aug 7 23:16:22 2021 +0800

remove unused imports

commit 08e0072f51dd1807280f6a2a6103af20049826c5
Merge: 40d62e3 169416e
Author: sinhaugoh <shgoh025@mymail.sim.edu.sg>
Date: Sat Aug 7 23:11:41 2021 +0800

Merge pull request #30 from sinhaugoh/feature/edit-retailer-profile

 Feature/edit retailer profile

commit 169416ea74780270bb8cf0c1812b00f08f4804f9
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Sat Aug 7 22:59:29 2021 +0800

 implement testing for retailer edit profile form notifier

commit c2615112cce44f7849e3d11ccd8600ff0dc77136
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Sat Aug 7 21:23:37 2021 +0800

 include routing to retailer edit form page

commit 8660fc32969430b77293467ee6e46ddd39995a6c
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Sat Aug 7 21:23:21 2021 +0800

 implement retailer edit form presentation layer

commit 7127bc68de41ca1f784769261c3b405a9532cbac
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Sat Aug 7 21:23:00 2021 +0800

 implement retailer edit form logic

commit 446288f819efba3ad8c1ef2fd63c5b8a5e26d651
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Sat Aug 7 21:22:03 2021 +0800

 implement routing from profile to edit form page

commit 82aaaf0987fa40f21ec7b52bf06ceae3972084114
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Sat Aug 7 21:19:58 2021 +0800

 implement setRetailer function to set retailer locally

commit e675bcf20f5f667af91bf7a7689ae622c2c360ec
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Sat Aug 7 21:19:29 2021 +0800

include initialValue parameter for form text field

commit a8aee241efcda2b48c600fef125a92d95d0cce28
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Sat Aug 7 21:19:10 2021 +0800

implement getUserId function in auth repository

commit 48e55912655324771a63716d430edc8760b2ab1a
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Sat Aug 7 21:18:36 2021 +0800

generated test

commit 299da1cf0084b99d83072f1412e86339ed798635
Author: Lawrence Ho <lawrenceehsj@gmail.com>
Date: Sat Aug 7 15:04:08 2021 +0800

fixed alignment in containers

commit 85ced8109ca5136ad23d721bd5a203d78a779e9c
Author: Kwang Kian Hui <2k1h297@gmail.com>
Date: Sat Aug 7 13:06:09 2021 +0800

Implement bug fixes and removed unused imports

commit 74bb99bc3b7edad0ec6e7351fdb5cd2ccab078c
Merge: 5b07652 40d62e3
Author: Kwang Kian Hui <2k1h297@gmail.com>
Date: Sat Aug 7 12:56:21 2021 +0800

Merge branch 'main' into feature/retailer/add-product

commit 40d62e3bb32b3e42cb13016165c9a60663da4b34
Author: Lawrence Ho <lawrenceehsj@gmail.com>
Date: Fri Aug 6 20:24:48 2021 +0800

commented out textTheme

something wrong. will remedy it

commit 1103674dcc70f2eef95938049dfd87f338642b3

Author: Lawrence Ho <lawrenceehsj@gmail.com>

Date: Fri Aug 6 19:50:46 2021 +0800

segregate containers with colours for better understanding

commit 5b07652039e9fe6b5437048583b4b8bc7c45ef83

Author: Kwang Kian Hui <2k1h297@gmail.com>

Date: Fri Aug 6 18:55:33 2021 +0800

Implement display product logic

Retrieved product from firestore and display in retailer home page.

commit bb868f04f056f4b974314f3aaa069621379802ca

Author: Kwang Kian Hui <2k1h297@gmail.com>

Date: Fri Aug 6 17:48:47 2021 +0800

Implement add product logic

Add product logic implemented, but error with retrieving product list due to firestore timestamp type needing to convert to FieldValue, will be fixed in next commit

commit 6f1dbafe963f22047c7222b4881f6e71daa32253

Merge: 9bf03b8 15a629f

Author: Kwang Kian Hui <2k1h297@gmail.com>

Date: Fri Aug 6 14:58:08 2021 +0800

Merge branch 'main' into feature/retailer/add-product

commit 9bf03b8c31815b11148f639f82e40a98efdb2c52

Author: Kwang Kian Hui <2k1h297@gmail.com>

Date: Fri Aug 6 14:54:42 2021 +0800

Merge branch 'main' into feature/retailer/add-product

commit 15a629f9d41bd595fb7377e5683340fcc70c0c2a

Merge: 10a0f7c cba30b3

Author: sinhaugoh <shgoh025@mymail.sim.edu.sg>

Date: Fri Aug 6 14:15:32 2021 +0800

Merge pull request #27 from sinhaugoh/improvement/cache-network-image

Improvement/cache network image

commit cba30b35130fd4bdd29403ff935510b8ab3a483d

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Fri Aug 6 14:14:06 2021 +0800

implement cached network image in retailer profile page

commit e7e1584ff6544835477ae3a787ab40c32df18adf

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Fri Aug 6 14:13:41 2021 +0800

import cached network image package

commit 10a0f7c0f3b0b810bf982712ee18ced471b7a0c6

Merge: 7552842 47f3ce9

Author: lawrencehsj <58553029+lawrencehsj@users.noreply.github.com>

Date: Fri Aug 6 13:54:39 2021 +0800

Merge pull request #25 from sinhaugoh/feature/size-config

added style_theme.dart for global styles

commit 755284220f1ab826bac2f21c3d8e812afdfb17c1

Merge: 2acfe59 a6105bf

Author: sinhaugoh <shgoh025@mymail.sim.edu.sg>

Date: Fri Aug 6 13:46:36 2021 +0800

Merge pull request #26 from sinhaugoh/feature/retailer-profile-logic

Feature/retailer profile logic

commit a6105bf6b6b14271fc85b79667e54f97eda31147

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Fri Aug 6 13:40:09 2021 +0800

implement hamburger nav bar routing logic

commit 55a207aea9ffe0add935ce8c10d250589e88afe6

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Fri Aug 6 13:39:51 2021 +0800

include the route to the retailer profile page

commit 8dc3eb9c8144cc7771798553b3a4c318876ea249

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Fri Aug 6 13:39:25 2021 +0800

refactor the name of retailerNotifierProvider

commit b662893c6e6c496af43c078bd8c1af1c4a43bdd4

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Fri Aug 6 13:39:06 2021 +0800

create retailer profile page

commit 28db57ff931dbe92aa919bf769f30a16ae5d44fa

Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Fri Aug 6 13:01:11 2021 +0800

shift code to consumer_sign_up_form

commit 47f3ce962c19050c8f6901986c6564efe0f127d2

Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Fri Aug 6 12:34:33 2021 +0800

added style_theme.dart for global styles

commit a5dcea5d7403cf7817f9a0b50c7f613b448981f3

Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Fri Aug 6 12:33:01 2021 +0800

update UI of consumer sign up

commit 2acfe59e2b0a9512a3280988fb90d09156d080fb

Merge: 8a95196 c26fe62

Author: lawrencehsj <58553029+lawrencehsj@users.noreply.github.com>

Date: Fri Aug 6 12:31:48 2021 +0800

Merge pull request #24 from sinhaugoh/feature/size-config

Feature/size config

commit 66f3a15af2a37d97188d18f396c6099cb7998bd7

Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Fri Aug 6 11:50:17 2021 +0800

Update sign_in_form.dart

commit c26fe6288da0ca078e595e573a62e1eefef112e4
Author: Lawrence Ho <lawrencehsj@gmail.com>
Date: Fri Aug 6 11:25:07 2021 +0800

added size_config.dart for global sizing variables

commit da4cde7c54e2cdb8736fa89cd7750a5baef2ee77
Author: Kwang Kian Hui <2k1h297@gmail.com>
Date: Thu Aug 5 23:31:22 2021 +0800

Merge duplicate product and product dto from main

commit 1b057f57848db917fd964fa8482ea2c57b2eec5e
Merge: 3a354d7 8a95196
Author: Kwang Kian Hui <2k1h297@gmail.com>
Date: Thu Aug 5 23:27:31 2021 +0800

Merge branch 'main' into feature/retailer/add-product

commit 8a951960e3c23948ce5ff6dc34d1983da156ffed
Merge: 9b4323a d584ac6
Author: sinhaugoh <shgoh025@mymail.sim.edu.sg>
Date: Thu Aug 5 22:25:46 2021 +0800

Merge pull request #23 from sinhaugoh/feature/implement-product-and-productDTO

implement produt and productDTO

commit d584ac6c55c950d70c28cc5db3f1ff91b0408586
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Thu Aug 5 22:10:01 2021 +0800

implement produt and productDTO

commit 3a354d7901182771119f714e992b04c82b13e20
Author: Kwang Kian Hui <2k1h297@gmail.com>
Date: Thu Aug 5 19:09:08 2021 +0800

Resolve conflict after main branch merge

commit 875e6e160ab9ac0291b0948d7338563006de57fe
Merge: e13bcd9 9b4323a

Author: Kwang Kian Hui <2k1h297@gmail.com>
Date: Thu Aug 5 18:55:05 2021 +0800

Merge branch 'main' into feature/retailer/add-product

commit e13bcd9fc0079fef60aa582bc93e2cb1b1b47b4d
Author: Kwang Kian Hui <2k1h297@gmail.com>
Date: Thu Aug 5 18:47:04 2021 +0800

Implement retrieve product list logic and form page

- 1)Added retrieve product list logic and display on retailer home page (subject to further changes, mainly used for visualizing and testing here)
- 2)Added product form page, error messages and text fields with formatting
- 3)Changes to FormTextField
- 4)Added extended_masked_text package (for usual price and discount price)

Changes to FormTextField:

- 1)Added controller field
- 3)Added prefix text field

commit 9b4323a785caf17a339e56ca276e0aa2bdbc449c
Merge: e5b1442 9878adf
Author: sinhaugoh <shgoh025@mymail.sim.edu.sg>
Date: Thu Aug 5 18:06:45 2021 +0800

Merge pull request #22 from sinhaugoh/feature/retailer-shop-visibility

Feature/retailer shop visibility

commit 9878adf3c6c7ad2335054ad771dc922674a49cb8
Merge: 892d2b3 e5b1442
Author: sinhaugoh <shgoh025@mymail.sim.edu.sg>
Date: Thu Aug 5 18:00:23 2021 +0800

Merge branch 'main' into feature/retailer-shop-visibility

commit 892d2b381595a03557b798a5aefae94a92196efe
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Thu Aug 5 17:58:50 2021 +0800

implement toggle visibility functionality

commit 1ec6b953518709548f31f6b945193f4f5dcdd852

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Thu Aug 5 17:58:14 2021 +0800

remove warning in this page

commit aed7312cecec4c7b989e471e00b2e43e25f3f1c2
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Thu Aug 5 17:58:03 2021 +0800

delete unused files

commit a2b6d22d776a75e2ee45b99578ab467657b5ad2e
Author: Lawrence Ho <lawrencehsj@gmail.com>
Date: Thu Aug 5 17:00:16 2021 +0800

testing welcome page layoutbuilder

commit acf20315a0270318640c8c7fa8d76faa1733b90e
Merge: 20a1130 e5b1442
Author: Lawrence Ho <lawrencehsj@gmail.com>
Date: Thu Aug 5 11:59:45 2021 +0800

Merge branch 'main' into feature/size-config

commit ce0836067cec172382976a055ada720c4ee972b9
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Thu Aug 5 08:43:33 2021 +0800

modify get retailer logic to show no connection page when no connection in home page

commit 80e095bcbad9c81b8a7d92c64dfd3ea44e510509
Author: Lawrence Ho <lawrencehsj@gmail.com>
Date: Wed Aug 4 19:02:17 2021 +0800

update to text instead of button

commit 7341ef05576805a6a100e47816dd3c9cd767c878
Author: Lawrence Ho <lawrencehsj@gmail.com>
Date: Wed Aug 4 18:25:58 2021 +0800

added in signup options

commit 3da13825b2069b4b8e381f380fae83a7027ea444
Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Wed Aug 4 17:34:32 2021 +0800

updated visuals for login page

not fully functional and

not responsive yet

commit e5b1442d34cf806f30c84bb8be4933d92a7a738

Merge: 84428c9 0493ee0

Author: lawrencehsj <58553029+lawrencehsj@users.noreply.github.com>

Date: Wed Aug 4 15:40:45 2021 +0800

Merge pull request #19 from sinhaugoh/feature/splashpage-2

Feature/splashpage 2

commit 0493ee015a166a3beca0e57c1af5c2f18cf19cb

Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Wed Aug 4 15:39:04 2021 +0800

removed warnings

commit ad779cac7913930a92afef16c1efca9840e8fbe9

Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Wed Aug 4 13:36:14 2021 +0800

updated 2nd splash screen and other changes

- added in assets/images

- enlarged splash.png

- set 2nd splash screen to same background color (with default fade transition)

- created images.dart class for image variables

commit 523ef849c7aad4ac166cfa630c39c1cd98ea44ed

Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Wed Aug 4 12:53:09 2021 +0800

updated logo_icon.png with no border

commit ab3380571d8c68ff7f928cb89d7c1bb9faeeae10

Author: Kwang Kian Hui <2k1h297@gmail.com>

Date: Wed Aug 4 11:46:59 2021 +0800

added firestore failures

added out of resource and unavailable service exceptions

commit 511a9633ac97aa80d66c33878bf5e826cc89a5b1

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Tue Aug 3 23:56:18 2021 +0800

implement retailer home page with initialisation logic

commit f95029757b007a1c31fb27357d5a5bfcb8e09aa9

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Tue Aug 3 23:55:32 2021 +0800

create providers to link up the logic

commit d29406de6757e89684d6438dc0abe0220a3d139b

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Tue Aug 3 23:55:00 2021 +0800

implement retailer initialisation logic in application layer

commit 8d20a4fc94efaae717fcc4f328fc5d99d0f5abf

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Tue Aug 3 23:54:33 2021 +0800

implement get retailer function in infrastructure layer

commit e8e0f6315246132a2126f4d1e21b0e04c10bfaa7

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Tue Aug 3 23:53:55 2021 +0800

restructure directory

commit e83b21709bd44fc786355a4754ad51a3d4b962d2

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Tue Aug 3 23:53:22 2021 +0800

implement retailer failures

commit bcb9577f14df9c1d08c7ae938d66e60141747793

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Tue Aug 3 23:53:00 2021 +0800

create connectivityProvider

commit ef226724cbaf44a1fc113b4c25cf34047cf29e1c
Author: Kwang Kian Hui <2k1h297@gmail.com>
Date: Tue Aug 3 21:34:23 2021 +0800

changed import file path

commit 4d5f86956916b12da30cd4573fd2abc0f235bab6
Merge: d6f92a3 84428c9
Author: Kwang Kian Hui <2k1h297@gmail.com>
Date: Tue Aug 3 21:31:47 2021 +0800

Merge branch 'main' into feature/retailer/add-product

commit 12d9b5d8f27cd9efa247b696648df748ca22e1cd
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Tue Aug 3 21:13:26 2021 +0800

remove duplicate provider

commit 84428c998a2d72b55e0685789685ff66b4ee6bf0
Merge: 5cc1b2e bb86bd6
Author: sinhaugoh <shgoh025@mymail.sim.edu.sg>
Date: Tue Aug 3 18:06:25 2021 +0800

Merge pull request #18 from sinhaugoh/feature/retailer-sign-up-logic

Feature/retailer sign up logic

commit bb86bd61a977ad7243ac27db2e645c74062f097e
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Tue Aug 3 01:05:01 2021 +0800

implement test for retailer sign up logic

commit d6f92a3960fadfb53d28e7d7ed8d4002d68f3988
Author: Kwang Kian Hui <2k1h297@gmail.com>
Date: Mon Aug 2 23:15:11 2021 +0800

implement add product notifier and state

Added add_product_notifier and add_product_state.

Edited product repository and remote service to implement add product feature.

Added 'product' and 'retailer' folder in core/application and core/infrastructure to group files.

commit a44133517ae199283d582205ce42ffecaff9ed70
Merge: 2bab374 5cc1b2e
Author: Kwang Kian Hui <2k1h297@gmail.com>
Date: Mon Aug 2 19:04:01 2021 +0800

Merge branch 'main' into feature/retailer/product-list

commit 20a1130423c60b29d4ec4dd71a67082c72f3a960
Author: Lawrence Ho <lawrencehsj@gmail.com>
Date: Mon Aug 2 16:10:35 2021 +0800

commented out portrait/landscape interaction first

commit b9e3fb25a203fd4794b2c2d7b622a3333ae868b4
Author: Lawrence Ho <lawrencehsj@gmail.com>
Date: Mon Aug 2 16:07:32 2021 +0800

added in size_config.dart for screen size config

commit ea2a2ebe4c054b67587ff1c5b341f29a1aee6866
Author: Kwang Kian Hui <2k1h297@gmail.com>
Date: Mon Aug 2 14:57:04 2021 +0800

Changed unit testing for retailer list notifier

commit 1598b0a8a2fb3b94e645ed3f9e08d56e2c5837db
Merge: 96087db 5cc1b2e
Author: Kwang Kian Hui <2k1h297@gmail.com>
Date: Mon Aug 2 13:59:47 2021 +0800

Merge branch 'main' into feature/add-on-unit-testing-to-retailer-list

commit 2a1ce512f60571ef17446b2c6096af8085efa6ae
Author: Lawrence Ho <lawrencehsj@gmail.com>
Date: Mon Aug 2 13:06:33 2021 +0800

added in splash img icons and splash screen

commit 20b1e58d2edbb766ca907a7fe68a3075f2f6041e
Author: Lawrence Ho <lawrencehsj@gmail.com>
Date: Mon Aug 2 12:51:18 2021 +0800

remove unused heading.dart file

commit 6b136b4f236249f7b9aac6eeb28fc1660f66c1da
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Sun Aug 1 23:18:45 2021 +0800

enable multiline for operating hours field

commit 7b1d2ab6576000915fc40cbf47b98b11027a2e57
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Sun Aug 1 23:03:06 2021 +0800

implement routing logic for retailer sign up feature

commit 5184617795710d05bde60c640840775141891495
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Sun Aug 1 23:02:38 2021 +0800

implement simple retailer home page

commit 480d6b62b7e7de885c447946f45da0403737b5b6
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Sun Aug 1 23:02:15 2021 +0800

implement storing of image in auth repository

commit 01d4a051bdbdc6e22e6d5fed028f8d8e15584a37
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Sun Aug 1 23:01:52 2021 +0800

change the name of the image field into imageFile

commit d913b4af341e2b8d2b65e960922839277c3daa83
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Sun Aug 1 23:01:07 2021 +0800

implement sign up logic for retailer in notifier

commit 5cc1b2e5aff9696b71e0ef18de01d3121b77eb31
Merge: baa69d1 d74b191
Author: Kwang Kian Hui <2k1h297@gmail.com>
Date: Sun Aug 1 16:06:35 2021 +0800

Merge pull request #14 from sinhaugoh/origin/feature/implement-connectivity-checking-for-retailer-list

Origin/feature/implement connectivity checking for retailer list

commit d74b191fb5c253ebd0e35aa3fc310962b54709f8
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Sun Aug 1 15:59:33 2021 +0800

adjust no connection flow for consumer home page

commit 96087db16c24401272938008dad9e83323cc2319
Author: Kwang Kian Hui <2k1h297@gmail.com>
Date: Sun Aug 1 12:47:30 2021 +0800

retailer list notifier unit testing

rename of retailer_list_repository_test unit test cases.
added unit test cases for retailer_list_notifier_test.
removed unnecessary mock in retailer_dto_test

commit 48bfb0a573e5e07d36e83aecf518c331e4da75b6
Merge: 13bbf70 4100a8a
Author: Kwang Kian Hui <2k1h297@gmail.com>
Date: Sun Aug 1 11:13:58 2021 +0800

Merge branch 'origin/feature/implement-connectivity-checking-for-retailer-list' into feature/add-on-unit-testing-to-retailer-list

commit 13bbf705973d0cfa8663accdd8afc05e5d2e29a5
Author: Kwang Kian Hui <2k1h297@gmail.com>
Date: Sun Aug 1 00:51:27 2021 +0800

retailer list repo unit testing

additional case test added "should return empty RetailerDTO list if failed to retrieve Json data from firebase collection list".

commit 24a2dea6f55b1a70df169d082613b45aa083ea38
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Sat Jul 31 23:46:38 2021 +0800

Create retailer_sign_up_form_notifier.freezed.dart

commit 0730905386fda266f7ebb39d36c0a0a160c0a2b8
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Sat Jul 31 23:46:22 2021 +0800

modify the parameter of retailerSignUpFormNotifierProvider

commit 075131bd5661394867b4951cd6c111ccdd44ffc5

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Sat Jul 31 23:45:58 2021 +0800

implement method to disallow password field from having spaces

commit 4fbe2ead20dd07f587359e286511261bf791f9b1

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Sat Jul 31 23:45:13 2021 +0800

implement placeholder UI for retailer sign up page

commit d49eafe6ca2909ece7f20f782940d319746f9dd2

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Sat Jul 31 23:43:29 2021 +0800

add in more parameter for FormTextField

commit f0025c79cee3531afe4273ff4b84805f5ca26331

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Sat Jul 31 23:43:04 2021 +0800

implement validation for non-empty and postal code text field

commit d6abf5d05461f04b847439371bd63cf63dd9046c

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Sat Jul 31 23:42:17 2021 +0800

implement ValueFailure for incorrect length

commit 67d002313d97c61a05984955e9a115e1c578eb4d

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Sat Jul 31 20:56:17 2021 +0800

add padding to the FormTextField

commit c1419d404e253fbbcc4560ed423c9f67e960fb4f

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Sat Jul 31 20:55:18 2021 +0800

convert email and password textfield into a universal FormTextField

commit 6038030ffed36cbf7816cace16dc5463bd758b90

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Sat Jul 31 20:18:28 2021 +0800

fix naming of SignInFormState

commit 6a8cf71081caa577b73d14719a3ca9ec9943c9a7

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Sat Jul 31 20:15:31 2021 +0800

include uen field for Retailer and RetailerDTO class

commit 72c038f92426768a1e6f22d4031253503ee6e181

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Sat Jul 31 18:50:07 2021 +0800

implemented retailerSignUp in auth repository

commit db726b5045e61a27e737eda5f6fe1b87bbac0c50

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Sat Jul 31 18:49:46 2021 +0800

implemented logic for creating a retailer firebase account

commit 82854666b1e5c588fa35e1ae18b225256fb88c62

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Sat Jul 31 18:49:27 2021 +0800

implement logic for creating retailer document

commit bbe057d29ebc37dcca7e165e781730c43841e1c7

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Sat Jul 31 18:49:05 2021 +0800

modify Retailer and RetailerDTO to make the description and image non nullable

commit 6fa1f6c47dbbcfbfe23bc1a31b1bf0c4bd829fdf

Author: Kwang Kian Hui <2k1h297@gmail.com>

Date: Sat Jul 31 17:53:13 2021 +0800

retailer dto unit testing

commit d53022ce48b73c866ac64a1dfcd0287db07c9365
Author: Kwang Kian Hui <2k1h297@gmail.com>
Date: Sat Jul 31 17:29:07 2021 +0800

retailer list repo unit testing

unit testing for retailer list

commit 4100a8a018c0cc191761003ae51a1ff13fd20156
Merge: d8ccb51 baa69d1
Author: Kwang Kian Hui <2k1h297@gmail.com>
Date: Sat Jul 31 12:09:53 2021 +0800

Merge branch 'main' into origin/feature/implement-connectivity-checking-for-retailer-list

commit d8ccb515b00122172eb68f2e1a0692426b9fc3c4
Author: Kwang Kian Hui <2k1h297@gmail.com>
Date: Sat Jul 31 12:09:07 2021 +0800

feature add on - refresh page

Used refresh indicator(pull down refresh) to allow refresh of consumer home page

commit baa69d199ef4e142d8ba07e7fb3311f848f09292
Merge: c3a84e0 1ad1cb4
Author: sinhaugoh <shgoh025@mymail.sim.edu.sg>
Date: Sat Jul 31 01:13:09 2021 +0800

Merge pull request #15 from sinhaugoh/feature/add-on-unit-testing-for-sign-up-and-sign-in

Feature/add on unit testing for sign up and sign in

commit 1ad1cb427a8c634d53f0709630eb81a184248b0b
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Fri Jul 30 15:25:54 2021 +0800

implement testing for value validator

commit 8dab8cd600be62eda368e7a0306f58d9de6d26dd
Author: Kwang Kian Hui <2k1h297@gmail.com>
Date: Fri Jul 30 14:33:13 2021 +0800

changed noConnection from firestore failure to state

removed noConnection firestore failure and switched to state.
displays text when state is noConnection

commit bce39913fd6423fb01251edf22907a554f988962
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Fri Jul 30 13:54:08 2021 +0800

implement test for auth notifier

commit 182eee0bae1ad93f117cb810c367b39812c21412
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Fri Jul 30 13:53:48 2021 +0800

remove linting rule so that generated mock class wont show error

commit 32f994bef16e12bc2497c0112e6e855dfedcce23
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Fri Jul 30 13:46:04 2021 +0800

remove redundant state

commit 2bab374f11852e2da837d2b07b8c3cb5de584f1e
Author: Kwang Kian Hui <2k1h297@gmail.com>
Date: Fri Jul 30 00:55:36 2021 +0800

file movement - firestore_failures.dart

Moved firestore_failures.dart and the freezed file to core/infrastructure since these failures can be used for both retailer and consumer query failures. Updated file path for necessary files

commit d3fd4324230333b37b2914ecfa81ae3b5ffceb8e
Author: Kwang Kian Hui <2k1h297@gmail.com>
Date: Fri Jul 30 00:48:18 2021 +0800

implement product list feature

- 1) core/shared/provider - included providers for provider
- 2) product remote service - query nested firebase collection data
- 3) retailer home page - initState -> get userID(uid) and pass uid to getProductList

commit ce7714f0c00340a9e766d3019ee6fdd40fd2adef
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Fri Jul 30 00:32:54 2021 +0800

implement unit testing for auth repository

commit 282ea12b64566fb05e26c5e61ff87b6de1723abd

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Fri Jul 30 00:32:26 2021 +0800

import mockito package for unit testing

commit c216b44ee6d767490cf38bf6a229673217527eb6

Author: Kwang Kian Hui <2k1h297@gmail.com>

Date: Thu Jul 29 19:04:59 2021 +0800

widgets for implementing product list feature

retailer drawer widget.

basic structure of retailer home page, product list view and product list item.

commit 2ad9ceaf75106c98b6902a15a31a995551ff1334

Author: Kwang Kian Hui <2k1h297@gmail.com>

Date: Thu Jul 29 18:12:42 2021 +0800

structure for implementing product list feature

necessary files (without actual code) and structure for retrieving product list data from firestore.

commit ce4569c0eb809cee1db47af7a7ea86d986563919

Merge: 1803dc9 c3a84e0

Author: Kwang Kian Hui <2k1h297@gmail.com>

Date: Thu Jul 29 17:35:13 2021 +0800

Merge branch 'main' into feature/retailer/product-list

commit e0ca2bc67807178cc5a0431af34ebcc60e4cc8bf

Merge: b9c6029 c3a84e0

Author: Kwang Kian Hui <2k1h297@gmail.com>

Date: Thu Jul 29 14:49:36 2021 +0800

Merge branch 'main' into origin/feature/implement-connectivity-checking-for-retailer-list

commit b9c602971ef10515d58393a4a59e948a5128f094

Author: Kwang Kian Hui <2k1h297@gmail.com>

Date: Thu Jul 29 14:48:55 2021 +0800

clean up unnecessary comments

commit fe68f6982121496328c97e214f1b8b0759781ee9

Author: Kwang Kian Hui <2k1h297@gmail.com>

Date: Thu Jul 29 14:47:59 2021 +0800

implement connectivity checker feature

import connectivity package v^3.0.6.

snackbar to notify user of "no connection" when no connection detected

commit c3a84e0d97a8d3f9d6b961a8137dff5354891db

Merge: b7d876c b774888

Author: sinhaugoh <shgoh025@mymail.sim.edu.sg>

Date: Thu Jul 29 13:35:25 2021 +0800

Merge pull request #12 from sinhaugoh/feature/sign-out-logic

Feature/sign out logic

commit b774888c50021e7f004147c6603bcf3c45266315

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Thu Jul 29 13:24:17 2021 +0800

implement sign out logic in presentation layer

commit a42ddc7bd3b920fe7e70d2f9b3a1650588649ff9

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Thu Jul 29 13:23:53 2021 +0800

implement sign out logic in application layer

commit 94092e94d7ec4f851e699f5efb3d43f99b6a2122

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Thu Jul 29 13:22:58 2021 +0800

implement signout logic in infrastructure layer

commit 90ebf7ada8a79d1072738932b82e1a846a269089

Author: Kwang Kian Hui <2k1h297@gmail.com>

Date: Thu Jul 29 13:13:53 2021 +0800

build_runner for firestore_failures

added no connection failure

commit b7d876cf3c743596bc9e2b6cbb84f9a085316a

Merge: a72112d 85a72d6

Author: sinhaugoh <shgoh025@mymail.sim.edu.sg>

Date: Thu Jul 29 12:57:02 2021 +0800

Merge pull request #11 from sinhaugoh/bug/fix-addressline-no-implementation-error

fix addressline no implementation error

commit 85a72d62978f04c34d2881e50df49a2b6de8f72c

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Thu Jul 29 12:41:37 2021 +0800

fix addressline no implementation error

commit a72112de6b5ad203a7a3d9951740de89467514f2

Author: Lawrence Ho <lawrencehsj@gmail.com>

Date: Thu Jul 29 11:36:43 2021 +0800

added google fonts

commit 7679d3d38c48d0ba68129b3c0014e34c7c5a29bc

Merge: 44adb7f 046ffef

Author: Kwang Kian Hui <2k1h297@gmail.com>

Date: Thu Jul 29 01:38:29 2021 +0800

Merge pull request #8 from sinhaugoh/presentation/design-and-UI

Presentation/design and UI

commit 44adb7f4f314eb73c3d233b5622ee8bcd4a33670

Merge: f8b7615 3d9969f

Author: sinhaugoh <shgoh025@mymail.sim.edu.sg>

Date: Thu Jul 29 01:37:00 2021 +0800

Merge pull request #9 from sinhaugoh/feature/image-picking-and-storing-from-local-to-cloud

Feature/image picking and storing from local to cloud

commit 046ffef1b66168dacf5f37b2596462362f433052

Author: Kwang Kian Hui <2k1h297@gmail.com>

Date: Thu Jul 29 01:29:11 2021 +0800

file movements

moved themedata_swatch_color to core/presentation/const

commit 4747f6e24d1e8eeb21daf2f78728ee8ae282f48f

Merge: 6f3c7af f8b7615

Author: Kwang Kian Hui <2k1h297@gmail.com>

Date: Thu Jul 29 01:27:12 2021 +0800

Merge branch 'main' into presentation/design-and-UI

commit 3d9969f769f9c424a94cba08d51b767c90be1fb0

Merge: 55af4f6 f8b7615

Author: sinhaugoh <shgoh025@mymail.sim.edu.sg>

Date: Thu Jul 29 01:26:09 2021 +0800

Merge branch 'main' into feature/image-picking-and-storing-from-local-to-cloud

commit f8b76151b36ab19d674da702b2d0ccaff19f1c9c

Author: Kwang Kian Hui <2k1h297@gmail.com>

Date: Thu Jul 29 01:17:29 2021 +0800

remove unnecessary files

retailer_provider.dart was not necessary and is removed.

all providers for retailer list moved to consumer/shared instead of core/shared.

commit 3501b83085c6d42ddbae5448a921fd88b3a9a1a8

Merge: ca04ce1 60c6f4c

Author: Kwang Kian Hui <2k1h297@gmail.com>

Date: Thu Jul 29 01:02:19 2021 +0800

Merge pull request #7 from sinhaugoh/origin/provider/models-and-riverpod-prov

feature/consumer/retailer-list

commit 60c6f4c1559a7f2e59f931928fc228c8867cf583

Merge: 0bc2ec0 ca04ce1

Author: Kwang Kian Hui <2k1h297@gmail.com>

Date: Thu Jul 29 00:56:36 2021 +0800

Merge branch 'main' into feature/consumer/retailer-list

commit 0bc2ec08497046845fbf4a1d830353a32a2b66a5
Author: Kwang Kian Hui <2k1h297@gmail.com>
Date: Thu Jul 29 00:48:38 2021 +0800

Update retailer list notifier

unnecessary unknown state setting removed as per requested changes.

commit ca04ce1d01ae5eb14b53298b09b708470bf388e4
Merge: f609b6b 2116035
Author: sinhaugoh <shgoh025@mymail.sim.edu.sg>
Date: Thu Jul 29 00:48:25 2021 +0800

Merge pull request #10 from sinhaugoh/feature/implement-connectivity-checking-for-sign-in-and-sign-up

Feature/implement connectivity checking for sign in and sign up

commit 211603559222ee91c8bb24e17026daec5ec34f99
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Thu Jul 29 00:02:13 2021 +0800

implement sign up connectivity check

commit 41b456e99dc184327e69672e2b478aad21c21e8d
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Thu Jul 29 00:01:59 2021 +0800

implement sign in connectivity check

commit 1d293e333e193f151120d65a514282adfffae5a5
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Thu Jul 29 00:01:37 2021 +0800

implement provider for the new package

commit 4ab1aaeafaa6b3345051196339ad5eb2ce53c5b3
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Thu Jul 29 00:01:00 2021 +0800

check for connectivity before communicate with remote services

commit 05ac002ec496572acea7a8b9764f1ce0a711b6e1
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Wed Jul 28 23:59:24 2021 +0800

import internet connection checker package

commit 1803dc9a671d3e366d7d24aeff5a4ae611878fb3

Author: Kwang Kian Hui <2k1h297@gmail.com>

Date: Wed Jul 28 20:28:39 2021 +0800

retailer product list feature

class object and dto file only.

commit 2851a15664371114a12f7d4cb56ce3312de6cd65

Author: Kwang Kian Hui <2k1h297@gmail.com>

Date: Wed Jul 28 19:39:34 2021 +0800

removed all product related files

all product related files will be in the feature/retailer/product-list branch

commit 55af4f63d22703287753cd753f202a0d78669e43

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Wed Jul 28 18:29:53 2021 +0800

implement provider for image picking feeature

commit 8a046bcf05958e2bfd8fd10fd419f7123fff10a3

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Wed Jul 28 18:29:17 2021 +0800

implement upload logic of product image to cloud storage

commit d37d5e764e58ba5d63d2255849b72bc852e741e8

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Wed Jul 28 18:22:52 2021 +0800

implement logic for uploading retailer logo to cloud storage

commit 72ba331f7df7bde539c1065ce389b3bad57a9638

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Wed Jul 28 18:22:18 2021 +0800

import path package

commit 6f3c7afda6fc96eaba4d8143fbe089a8fde93da9
Author: Kwang Kian Hui <2k1h297@gmail.com>
Date: Wed Jul 28 17:16:59 2021 +0800

minor changes

Moved swatch color to themedata_swatch_color.dart file

commit 30dd8f81c18d260b0356e2dd85814ff9cc3a1c4d
Author: Kwang Kian Hui <2k1h297@gmail.com>
Date: Wed Jul 28 17:09:52 2021 +0800

changes after feedback 3

notifier, provider files

Changed consumer home page of ConsumerWidget to Consumer StatefulWidget.

get retailer list from firestore on initialisation of consumer home page.

commit b6ddd06c15245022cba1b1b66407a44bfd612105
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Wed Jul 28 15:41:45 2021 +0800

implement image picking logic

commit aa34f062ba2edbf914c92c001be32111a36a4f9f
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Wed Jul 28 14:43:57 2021 +0800

import cloud storage for storing image on cloud

commit 853fe48b4bb00bab96e58bec408d303b218dad09
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Wed Jul 28 14:40:23 2021 +0800

import image picker package to enable picking of image from local gallery

commit 550483d4158ac15c460392a880c9301e8b654e17
Author: Kwang Kian Hui <2k1h297@gmail.com>
Date: Wed Jul 28 14:11:33 2021 +0800

changes after feedback 2

Removed unnecessary lines of code in analysis_options.yaml.

Included retailer list state and notifier

Removal of no connection failure as error is not stated in the firebase doc

Changes made to error code to follow android error codes (previously was web error codes).

commit d0acc85671092ac4b981067354f98a826f2305f7

Author: Kwang Kian Hui <2k1h297@gmail.com>

Date: Wed Jul 28 13:11:47 2021 +0800

changes after feedback

Retailer list - remote service and repo. Possible failures of firestore.

Retailer data transfer objects(dto)

commit c05f4de6d07a2f03d177396e7ed373f7170092c9

Author: Kwang Kian Hui <2k1h297@gmail.com>

Date: Tue Jul 27 17:06:45 2021 +0800

firebase test data

future builder and list view to retrieve firebase retailer test data

commit 5bb5768dd0002fb324ce02c7233acef7ad97c3bb

Author: Kwang Kian Hui <2k1h297@gmail.com>

Date: Tue Jul 27 13:38:33 2021 +0800

listview display dummy data

listview builder successfully displays dummy data. Added 1 additional dummy data. Basic listview item structure.

commit 0a2d501f806a2877040137f5e99e410bcd843829

Merge: c13e7b7 f609b6b

Author: Kwang Kian Hui <2k1h297@gmail.com>

Date: Tue Jul 27 00:38:37 2021 +0800

Merge branch 'main' into presentation/design-and-UI

commit 927276c264edbe2b83c1dfbfc5ddda353a6fd9e5

Author: Kwang Kian Hui <2k1h297@gmail.com>

Date: Tue Jul 27 00:37:53 2021 +0800

minor changes

merge from main branch and some changes

commit a66f52e9b66388d8e9711c5935f61ead01157ace

Merge: cac2116 f609b6b

Author: Kwang Kian Hui <2k1h297@gmail.com>

Date: Mon Jul 26 23:13:28 2021 +0800

Merge branch 'main' into origin/provider/models-and-riverpod-prov

commit f609b6bea1aaa1712d323280d7a3036b499df771

Merge: 049fbce f3f9131

Author: sinhaugoh <shgoh025@mymail.sim.edu.sg>

Date: Mon Jul 26 23:10:39 2021 +0800

Merge pull request #6 from sinhaugoh/feature/consumer-sign-up-logic

Feature/consumer sign up logic

commit c13e7b7d4d467f921c2fd9ec9f7013bb65b7bb85

Author: Kwang Kian Hui <2k1h297@gmail.com>

Date: Mon Jul 26 22:38:41 2021 +0800

Primary Swatch

primary material color

commit f3f913162f139f0c97ace929ed43b2054c6759ee

Merge: 454698b 049fbce

Author: sinhaugoh <shgoh025@mymail.sim.edu.sg>

Date: Mon Jul 26 18:26:38 2021 +0800

Merge branch 'main' into feature/consumer-sign-up-logic

commit 454698bd646242a195fa533432d4b55642284b2d

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Mon Jul 26 18:22:49 2021 +0800

rename email and password textfield

commit 8c7f7afb34e9c499e911ac4336cebfbba03bbf94

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Mon Jul 26 18:17:22 2021 +0800

implement presentation layer and routing for consumer sign up

commit 79b99c76f2e8d1d3b42a4657880d1d690299e363

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Mon Jul 26 18:16:48 2021 +0800

implement providers needed for consumer sign up feature

commit 98bdf3351295c79a7e3e52088cf2505ddf8fef77

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Mon Jul 26 18:16:25 2021 +0800

implement application logic for sign up form

commit d3327cb79b802cd8e86cc84bb4af4f1c4f08de33

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Mon Jul 26 18:15:38 2021 +0800

remove todo comment

commit 19f262f2abae75d46125b1876af76111d2e552d4

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Mon Jul 26 18:15:20 2021 +0800

implement infrastructure logic for consumer sign up feature

commit 5b34b27c775d4496a83032b9289fd65ccfb09170

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Mon Jul 26 18:13:47 2021 +0800

remove unused field in sign in form notifier

commit cac21162dd74f8a266097efb66ebfae79ad3febe

Merge: 3043369 049fbce

Author: Kwang Kian Hui <2k1h297@gmail.com>

Date: Mon Jul 26 15:34:33 2021 +0800

Merge branch 'main' into origin/provider/models-and-riverpod-prov

commit 049fbce18a01099f8bffc9f1c150476fd5e82cd7

Merge: 6c0e4f0 81a55a8

Author: sinhaugoh <shgoh025@mymail.sim.edu.sg>

Date: Mon Jul 26 15:32:31 2021 +0800

Merge pull request #5 from sinhaugoh/bug/disable-navigation-back-to-splash-page

fix navigation to splash page issue

commit 85daf3399f52895a5881a8723c05b71e7fd9fa96
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Mon Jul 26 15:18:44 2021 +0800

make email and password text field reusable in exchange of slight drop of performance

commit 81a55a8a8a00d3602c0d351961dce8bef81c167b
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Mon Jul 26 15:02:24 2021 +0800

fix navigation to splash page issue

commit 304336936acbea4d797465e898056c08a6eef63f
Author: Kwang Kian Hui <2k1h297@gmail.com>
Date: Mon Jul 26 12:24:23 2021 +0800

Widgets and some changes

Added widgets structure (listview, etc) for the consumer home page.
Remove ProviderScope() wrap of the AppWidget (ProviderScope() to wrap specific widgets that requires it).
Added 4 dummy data in retailerlist.

commit f1b472e0fedd02fc8ab49b3c69e5304f6b93a8b
Author: Kwang Kian Hui <2k1h297@gmail.com>
Date: Sun Jul 25 01:47:15 2021 +0800

Models and Providers

class and provider files for Product and Retailer

commit 6c28da03090fba3dd26ffb8578d02eccf2d36847
Merge: 3441387 6c0e4f0
Author: Kwang Kian Hui <2k1h297@gmail.com>
Date: Sun Jul 25 01:02:17 2021 +0800

Merge branch 'main' into origin/provider/models-and-riverpod-prov

commit 6c0e4f08b205055163965c942cf87fc24109c19f
Merge: 01e6838 1be1714
Author: Kwang Kian Hui <2k1h297@gmail.com>

Date: Sun Jul 25 00:28:49 2021 +0800

Merge pull request #4 from sinhaugoh/feature/user-authentication

Feature/user authentication

commit 1be1714e30ea667c88b1cd0ac03a8f4852d0f25e

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Sat Jul 24 23:35:21 2021 +0800

implement sign in logic for both consumer and retailer

commit 57f7309647707b73676c15d0b69c6bc7d906e2c1

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Sat Jul 24 23:34:28 2021 +0800

create placeholder consumer home page widget

commit 40b533cb0cf23009ff12f2852118656d59ee470a

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Sat Jul 24 23:28:22 2021 +0800

implement password validation

commit 102e83d52b83d2f57376c0831897008fdce9c594

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Sat Jul 24 23:28:02 2021 +0800

implement const form error messages class

commit 755bd543d915eb38031c2893deb4b75f8cc6b5c1

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Sat Jul 24 23:27:32 2021 +0800

create saving in progress overlay widget

commit a9b6444904adcc50d460a3759562b51a6b5cc70e

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Sat Jul 24 23:27:04 2021 +0800

create empty retailer home page

commit ba5be3fb3c80a5141d8de479500193568dd2c02f

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Sat Jul 24 23:25:55 2021 +0800

adding temporary admin functionality to splash page

commit 66e0863abb06172959cd5d92dd1c4ba4601b8a60

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Sat Jul 24 23:25:09 2021 +0800

create app user entity

commit 3441387e9d55eb97c8e535266af646bfef113ccf

Author: Kwang Kian Hui <2k1h297@gmail.com>

Date: Sat Jul 24 18:01:39 2021 +0800

Product object and Provider basic structure

commit c8562028e57328fcd93aa5b6eb390df5c74e459c

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Sat Jul 24 17:50:20 2021 +0800

implement email validation

commit 16e4bf14d1d0c293720c64948cfdfa66f7e6a8f8

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Sat Jul 24 17:50:03 2021 +0800

implement value failures union class

commit 6f6b42e1d99c2541b9a5aaca79152519dabdbb62

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Sat Jul 24 17:49:33 2021 +0800

include sign in and consumer page route into navigation

commit c1905c312cc9b316f79542478b1cc3b2a3ed1eab

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Sat Jul 24 16:14:39 2021 +0800

reimplement getting auth status function from firebase auth

commit 412465ba1a7bd2e95c7d314b96552ffa8cfa9038

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Sat Jul 24 01:30:23 2021 +0800

create a initialisation provider that will always run after app launch

commit 581d7f9eeed096df3ea1070b96cd15757393efe2

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Sat Jul 24 01:29:10 2021 +0800

configure and implement auto_route package

commit 63fe3fce8dccb21f2faf69bd86e337415d5b023e

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Sat Jul 24 01:28:50 2021 +0800

wrap AppWidget with ProviderScope

commit d40c02dbd7f8e7133545733227b89ab96c1c41fd

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Sat Jul 24 01:28:06 2021 +0800

create provider for authNotifier

commit e7620f29d979581df00dd721512d0903963f9dd9

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Sat Jul 24 01:27:25 2021 +0800

implement checking of user auth state in application layer

commit 7f4e6c69cb1bea51d218cb34f61fd2a63e011c08

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Sat Jul 24 01:26:41 2021 +0800

implement checking of user auth state in infrastructure layer

commit e6060e002c2fac50df429485977444787da84f09

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Sat Jul 24 01:25:40 2021 +0800

create placeholder page (welcome page and splash page)

commit 01e6838c224d21ded23b2993d114f01534e1f269

Merge: 24fdd07 b31f997

Author: sinhaugoh <shgoh025@mymail.sim.edu.sg>

Date: Fri Jul 23 22:42:29 2021 +0800

Merge pull request #1 from sinhaugoh/feature/configure-firebase-services

Feature/configure firebase services

commit b31f9976f07a862422b3c677826d733bb0674299
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Fri Jul 23 17:22:28 2021 +0800

import firestore and authentication packages

commit 35b98012fa8bbc8b9b7227d615d4f2c6f9ecafdf5
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Fri Jul 23 17:16:57 2021 +0800

update minimum sdk version to 21

commit 566ca9e631ba529a634b643a9bed4d1fd98a2c64
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Fri Jul 23 17:16:42 2021 +0800

configure firebase service into this project

commit 24ffd079803ad67f6043fbe1ef38d1c63e5629cf
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Fri Jul 23 16:37:19 2021 +0800

restructure main.dart

commit 6949e632f60d72ceefe92e7f2ac0d97ec8a0420a
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Fri Jul 23 16:36:56 2021 +0800

import necessary packages

commit b828414b9495453b683a1a55026e582ff596463b
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Tue Jul 20 01:55:35 2021 +0800

add stricter linting rules using lint package

commit e3315097c14f0280bf2f691be6f01e608af796dc
Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>
Date: Tue Jul 20 01:25:58 2021 +0800

clean up comments in the project

commit 47321b8c265a7c728ad5545fe8800fa5e8e4d813

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Mon Jul 19 23:20:35 2021 +0800

Update .gitignore

commit 4277699b008f4e7d9041e308f81a991038d296d4

Author: Goh Sin Hau <shgoh025@mymail.sim.edu.sg>

Date: Mon Jul 19 15:46:00 2021 +0800

initial project

References

Babich, N. (2019, August 6). Progressive Disclosure: Simplifying the Complexity. Shopify. <https://www.shopify.com.sg/partners/blog/progressive-disclosure>

Best fonts for mobile app design. (2020, April 17). JustInMind. <https://www.justinmind.com/blog/best-font-mobile-app-design/>

Bhagavatula, S., Bauer, L. and Kapadia, A., 2020. *(How) Do people change their passwords after a breach?*. p.1.

Decker, K. (2020, July 7). The fundamentals of understanding color theory. 99designs. <https://99designs.com.sg/blog/tips/the-7-step-guide-to-understanding-color-theory/>

Experience, A. (2018, December 13). 10 Most Important Mobile UX Design Principles You Should Know. Saudi Arabia's 1st Web & Mobile UX, Usability & Ecommerce Research, Experience Design and Testing Consultancy. <https://uxbert.com/10-mobile-ux-design-principles/#.YScXcl4zY2x>

Firebase. n.d. Usage and limits | Firebase Documentation. [online] Available at: <<https://firebase.google.com/docs/firestore/quotas>>

Giampedraglia, P., 2020. What are push notifications and why are they important? - asap developers. [online] asap developers. Available at: <https://www.asapdevelopers.com/what-push-notifications/>

Jeff Sauro, P., 2021. 10 Things To Know About Completion Rates – MeasuringU. [online] Measuringu.com. Available at: <https://measuringu.com/completion-rates/>

Kolieboi, A. (2019, April 14). Thumb Thing Thursday: The role of hand position and mobile design. Asher Kolieboi. <http://www.asherkolieboi.com/blog/2019/4/14/thumb-thing-thursday-the-role-of-hand-position-and-mobile-desgin>

Nea.gov.sg. n.d. Food Waste Management. [online] Available at: <https://www.nea.gov.sg/our-services/waste-management/3r-programmes-and-resources/food-waste-management>

Nielsen Norman Group. 2021. Time Budgets for Usability Sessions. [online] Available at: <https://www.nngroup.com/articles/time-budgets-for-usability-sessions/>

Nielsen Norman Group. 2021. How Many Test Users in a Usability Study?. [online] Available at: <https://www.nngroup.com/articles/how-many-test-users/>

Nielsen Norman Group. 2020. 10 Usability Heuristics for User Interface Design. [online] Available at: <https://www.nngroup.com/articles/ten-usability-heuristics/>

Towardszerowaste.gov.sg. n.d. Food Waste. [online] Available at: <https://www.towardszerowaste.gov.sg/foodwaste/>

UNEP. n.d. Worldwide food waste. [online] Available at: <https://www.unep.org/thinkeatsave/get-informed/worldwide-food-waste>

What are User Interface (UI) Design Patterns? (2021). The Interaction Design Foundation. <https://www.interaction-design.org/literature/topics/ui-design-patterns>

WittySparks. 2021. *Importance of Animation in App Design*. [online] Available at: <https://wittysparks.com/animation-importance-in-mobile-app-design/>

Todd DeCapua n.d. [online] Available at: <<https://techbeacon.com/app-dev-testing/understanding-front-end-vs-back-end-performance-metrics-mobile-apps>> [Accessed 20 September 2021].

Reflection

Throughout the duration of this second part of the group assignment, my team and I worked closely to develop the app. I got to experience working with others remotely with the use of git where we work on our assigned tasks very easily as there is no need to wait for other members to complete their task before starting to work on mine.

We planned each sprint and met regularly to ensure we can always meet the deadline and have sufficient time for testing. As most of us are relatively new to developing mobile applications, we had to take some time to learn from more sources. We searched for various tutorial videos on developing mobile apps with clean architecture and had to read and understand the documentations written by flutter.

I am glad that my team was able to work well together and meet our deadlines despite having busy schedules throughout the duration of the assignment. I learnt to work with other members of the group through the use of git, resolving conflicts and experience what it is like to meet users to perform testing. The knowledge I gained on both mobile development and other aspects of agile techniques will definitely be beneficial to future works.