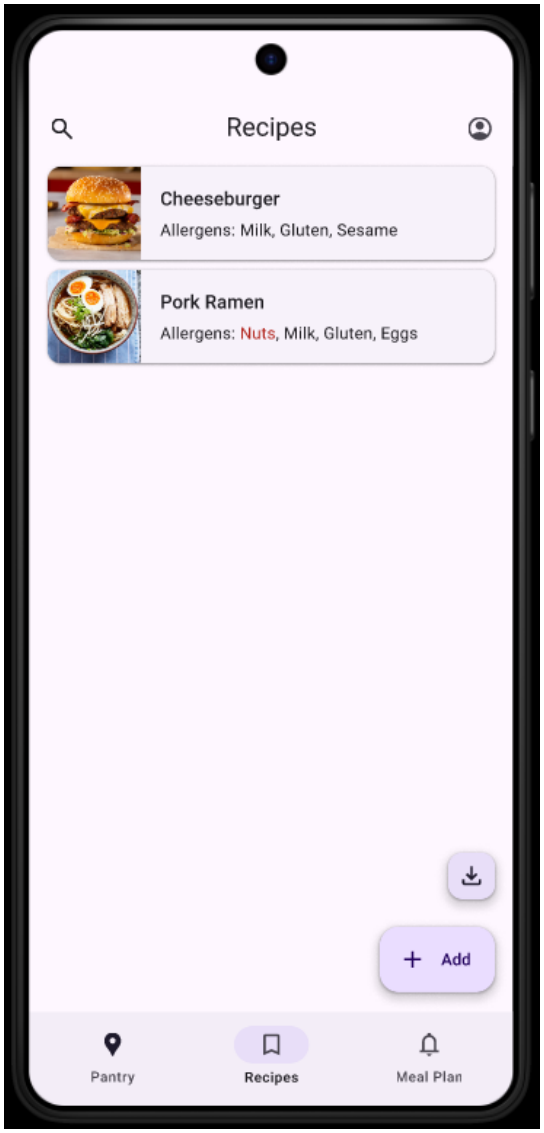


PANTRY PLAN

Food tracking app

ABSTRACT

Pantry Plan is an android application to help keep your pantry organised and well stocked.

Ellie Antonowicz, Layton Cripps, Seth Humphries, Billy Putland, Neo Salmon, Megan Welter

CI536 Integrated group project

Lab tutor: Karl Cox

Table of Contents

CI536 Group Submission Statement of Contribution

Student name	Percentage contribution
Ellie Antonowicz	100%
Layton Cripps	100%
Seth Humphries	100%
Billy Putland	100%
Neo Salmon	100%
Megan Welter	100%

Introduction

Goals

Our goals for this project were to create an application that can track what food you have in your pantry and give you meal recommendations. While brainstorming ideas we decided we wanted to do an android app and came up with a few different project ideas, for example an app that identifies something either using image recognition or a string of questions. By the end of the first lab, we had agreed on making 'pantry plan'.

'Pantry Plan' is an app that tracks what food you currently have and what meals you can make with it, it also allows you to set notifications to remind you when you need to buy new food or when food is about to go out of date. Users can provide a list of allergies and intolerances in their profile, as well as having access to dietary information to help users understand the basics of nutrition. Each meal you can make comes with a list of ingredients and a step by step guid of how to cook it, showing you how much of your current food it will use up.

Our goals for this project are to create an efficient app that can help reduce food waste and make meal planning easier. It should allow users to easily track their pantry and see what food they need to buy. We want our app to be user friendly by being intuitive and fast. The app should be scalable as for this assignment we are making a prototype, we want to make sure we can make this app better in the future with more features.

During this project we achieved making an app and meeting the majority of our planned goals, we also worked well together as a professional team, working together on different sections of the assignment based on our skill sets.

Here is a link to a video demonstration of our application:
https://github.com/jds691/CI536-Group-Project/blob/main/submission/recording.mp4

Our GitHub repository can be found using this link:
https://github.com/jds691/CI536-Group-Project

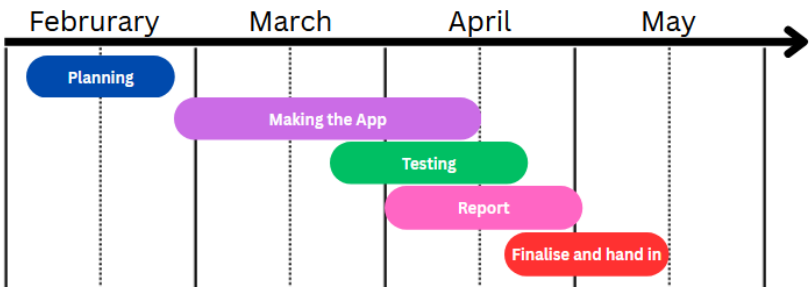
Methodology

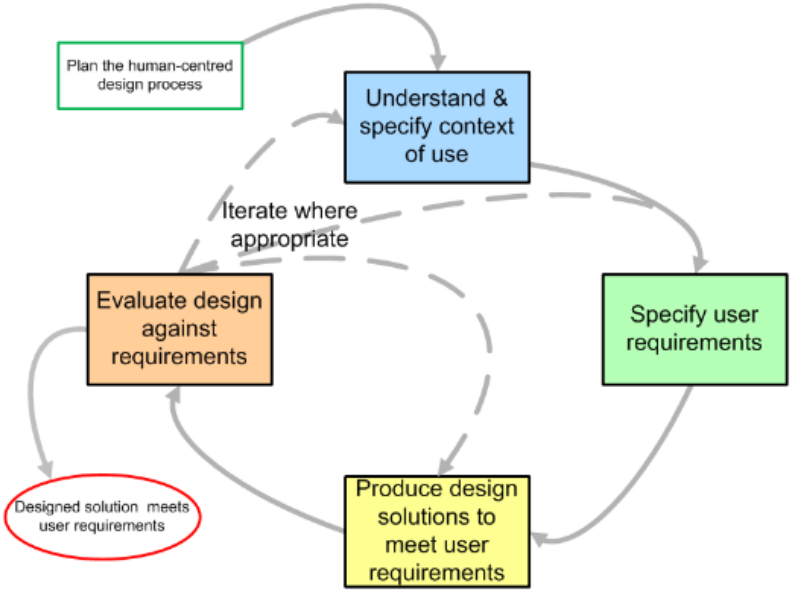
Management

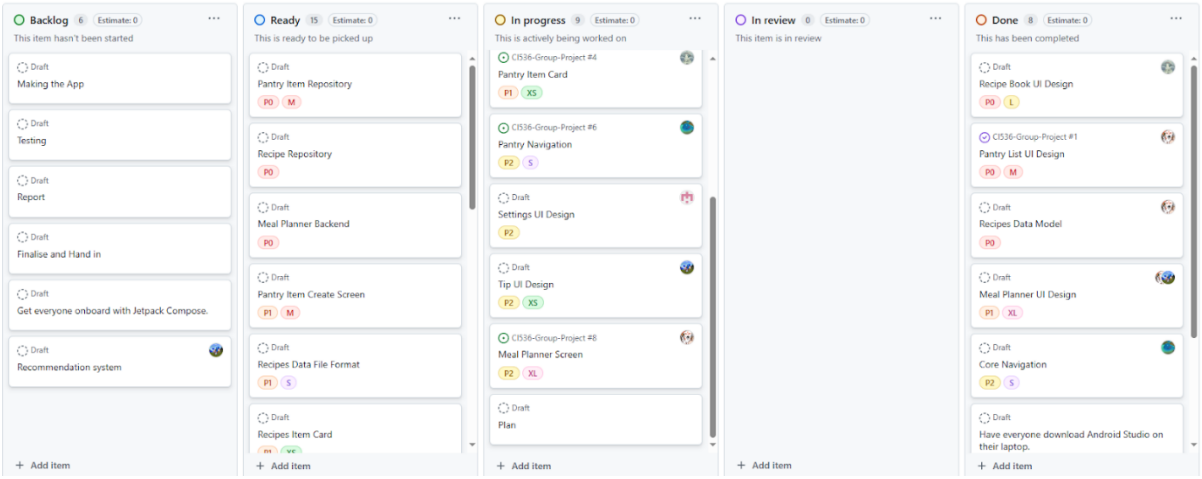
To manage our project, we decided to set up a discord group so that we could have easy communication, we assigned different sections of the project to each member dependent on their skills and set out a time frame of when we wanted each section done.

Each week we had meeting in the CI536 lab and an extra meeting on a Monday (see appendix for meeting minutes) and checked in with everyone to make sure we were on track, also help people if needed. At these meetings we could also give feedback.

We used waterfall project management, for the first few weeks of the assignment we spent time forming our group and deciding on what we wanted to make, then the requirements of the app we had decided on and what roles each member would play. We spent the majority of the time making the app, starting with designing then the implementation. During this time we also started testing and writing some parts of the report. Finally, we finalised everything before the hand in. We made sure to document our project throughout.







Each members roles and tasks are summarised below (Not all tasks):

Billy: Designed and programmed recipe screens, created UI cards to be used throughout the app.

Ellie: Documented the project and completed the weekly lab tutorials, as well as writing the report.

Layton: Designed Figma notifications screen, programmed macros UI card, ‘pantryItemDetailsScreen’ UI and functionality.

Megan: Database

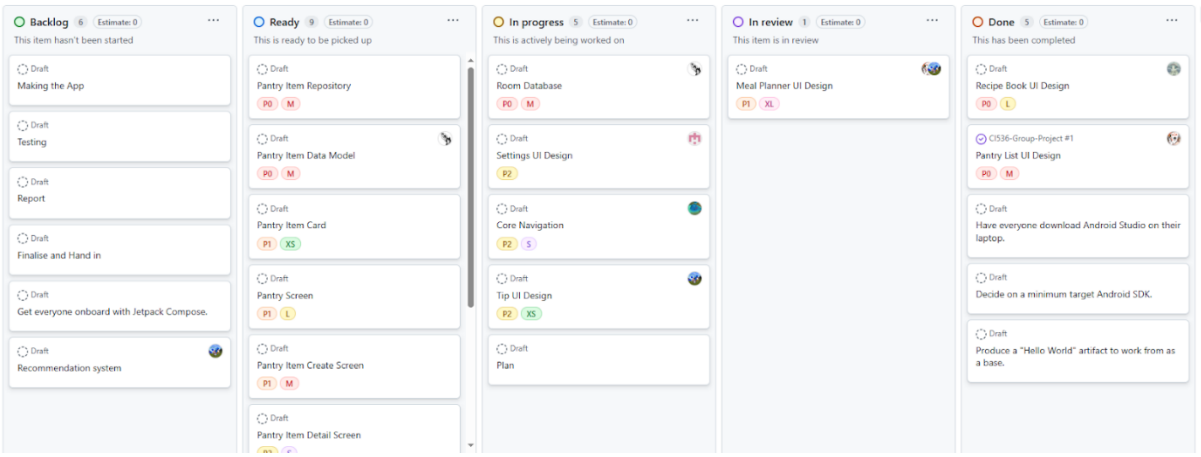
Neo: Designed meal planner, macros card, leaded UI design, managed project and pull requests. Implemented CI, unit tests, meal planner feature, data-access module, settings, data store module and navigation.

Seth: Designed the recipe ingredient card. Programmed the pantry item list screen, the navigation bar and the barcode scanner.

Development tools

For the development of our mobile application, we used software like: Figma, GitHub, and android studio. Figma is a collaborative design tool that we used to design and prototype our user interface, it allowed us to all help design the app at the same time, this was helpful as we were able to work off each other and easily change our design over the weeks we worked on the project. We used GitHub to both store our code and to manage the projects task board; the GitHub project tool allows us to assign each member to different tasks, enabling us to keep track of what is currently being done and what needs to be done. Android Studio is an integrated development environment for creating applications for the Android OS, it has an emulator allowing us to test and build the app on our desktops, which is important as not all of us have android devices.

We talk more about tools in the ‘implementation’ section of this report.



Testing

To reduce bias in our test results we got one of our group members who hadn't worked on the code to do testing. To test the app, she completed multiple tasks such as adding a new item to the pantry using both the manual input and the barcode scanner, then she tested the recipe section by adding and uploading new recipes and trying them out, she said that the instructions were easy to read and appreciated the ease of use of the app.

User Experience approach

Analysis of existing systems and applications

Before we started on this project, we did some market research to find what similar apps that are and what they do. Form this we could find what we did/didn't like and how to make our app different and stand out from the others. Existed apps included: Mealime[1], Tasty[2].

Similar apps	
Positives	Negatives
Mealime	
While onboarding, asks for diet, allergies, dislikes serving preference and asks if you want to set weekly reminders.	Doesn't mention dairy allergies, so this feature is useless for one of our group members. Prompts you to create an account.
Lists cookware and ingredients needed.	There is a lot of features locked behind a subscription, for example: Some recipes, nutritional information, ability to add own recipes, search features and meal plan history.
Food waste calculator.	Food is extremely bougie, which wouldn't be helpful for our target audience.
Once meal plan set, the app fully opens up and you can access all menus. The grocery tab has a badge with the number of items you need to buy.	Lists allergens after you have already selected the recipe.
Tasty	
During onboarding, asks about diet, allergies and 'what kind' of recipes.	
Simpler foods.	Not organised very well.
Lists time required to prepare and has social media like feature, for example, like to dislike ratio, comments and community tips.	Challenging to navigate due to lack of filters.
Doesn't paywall nutritional information.	

[1] (Mealime, n.d.)
[2] (Buzzfeed, n.d.)

Personas

We have to consider what implications our users characteristics have on our design, for us, our users (based on our persona) are mostly busy young adults who want to save money by reducing food waste, this has an impact on our design as we need to make it quick and easy to use, so direct users don't need to take time out of their day to use our app, it should be quick and easy without being a chore. We also want the design to be modern to fit with people's expectations and have meals that are realistic for students to have.

When thinking about our users we need to consider different demographics as the development of our app would be tailored for specific demographic due to their different needs and life experiences.

Age

Links to other attributes, age effects how users engage with content, our target audience is people aged 18-25.

Location

We are based in the UK so meals will be based off ingredients that are commonly found here. If we wanted to further our design, we could make it more specific for location so for areas (for example) that have lower food prices on average, then be recommended meal that use more ingredients. We could also expand it for other countries as everywhere has different foods that are found in shops and different food preferences.

Socioeconomic status

As we are aiming at predominantly university students, we want to make meals cheap, we could improve the app more in the future so the user can give a weekly budget for food so they can get more help with budgeting and planning meals.

Religion

We need to make sure dietary requirements are met, it would be good to ask what food they don't eat so that their meal options are not filled with meals that they cannot have, which would make it more time consuming to dig through recipes to find something that fits your diet. Another approach could be that they are given alternatives. Another thing we need to be aware of is fasting; people may want to turn off notifications during certain times.

Marital status

We could develop the application so that multiple accounts can accessing the pantry, different users could have personalised profiles like unique allergies. Having multiple accounts access the same database would also allow easier shopping as both partners would know how much food they have and what they need to buy. If they have a family

this could be further expanded so kids can also be given access. On the children's app they could have more educational content, so they learn from a young age the importance of eating healthy, how to make food, and possibly even food budgets. However, this would be too complicated to implement for a prototype so we will be focusing on one user only, which also fits our persona as most university students are not married and do not have kids. The majority of students share kitchens, however not many share food so this is another reason we are currently keeping it to one user per pantry.

Ownership

If the user has a pet (unlikely for university students) they could be reminded when they need to buy new pet food or have notifications for when it's time to feed pets.

If the user does not own a car (common for students) then the app would need to be aware that large shops are not really accessible as the user will have to carry everything home. Our persona lives in Brighton which has a lot of student accommodation on hills, making carrying more that 2-3 bags difficult so it would be better to do more frequent small shops, or take transport costs into account.

People will need to own a phone to use the app, however this isn't being considered as all university services require a phone.

Mobility

Expanding on from ownership, people without a car will have less mobility then those with and some people may want to save money and not use public transport. There is also a problem if people have disabilities so it would be harder for them to shop.

Using the app may be harder for some people due to vision or movement impairments, to combat this we will make sure our app complies to usability and accessibility laws and is easily readable with larger buttons in case of tremors.

Language

In our prototype we will only have English as an option, but other languages could be implemented if we wanted to make a finished product. We will not be using unnecessarily complicated words so even if the user has low literacy, then it should still be accessible. However, due to our target audience being students they will have a higher literacy skill.

Affect (Intrinsic and Extrinsic Motivations)

The user we based our persona on wants to reduce food waste; this is an intrinsic motivation as there are no external rewards. By using our app, users can save money which could be spent elsewhere (extrinsic motivation), the app could have a 'money saved' calculator so user could see how much they've saved and spend the saved money on a treat.

The user may want to start cooking as a hobby, this would be a good starting point as it will have lots of simple recipes to get started with without having to spend lots of money on food/equipment. Due to our target audience being students, they would have a positive attitude towards learning. However, this isn't needed for our application as although there is the ability to learn from it, it can solely be used for productivity.

Another aspect we need to consider is people's attitudes towards technology, many people (especially older generations) do not want to have technology in every aspect of their life and would not even consider using an app for tasks they have been doing their entire life without. This should not be a big problem for us as our target audience have a much more positive attitude of technology and, due to them being young, may still be struggling to manage their food without assistance.

Visual Literacy

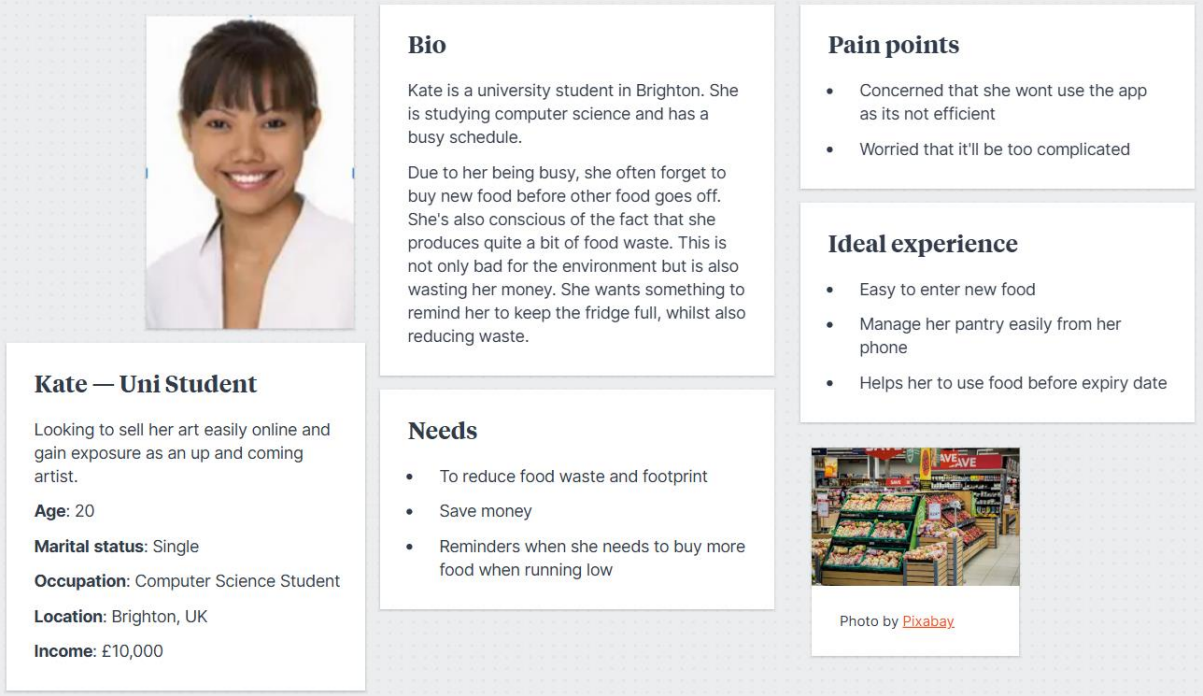
There will not be any complicated graphs in our application and any graphs used will not only be simple buy also have explanations. Graphs will be used for those who visualise better with graphs instead of words so different types of learners will be accommodated.

Disability

When designing applications we need to consider people with disabilities, it is important to make apps accessible, to a wide range of people, for example, people with dyslexia or visual impairments need to be able to use screen readers on the app. People with motor impairments need bigger buttons to click easily. Need to make sure colours and font are accessible by using contrasting colours and large, clear fonts.

Our app could also help people who we are not targeting, for example it could help carers for people with dementia as they can easily see what food they have and need to buy.

Other demographics like orientation, ethnicity and education should not have a big effect on how the user would interact with the app.



Task Environment

For the application to work efficiently and improve users experience, the app would need to be used regularly to keep the pantry list up to date. If the app is not used often then the food amounts will be wrong, and the user may not have the correct amounts of ingredients for their meals, making the app redundant.

Task Scenario

Task Scenario

4 cards

Project Title: Pantry Plan

Date: 10.05.25

Company: University of Brighton

Authors: Pantry Plan Group

Scenario 1: Choosing a meal

Practical Goal: Using the Pantry Plan app, choose a meal to have for dinner.

Context: In a kitchen

Users: University Students

At home, in the kitchen. It is late, so you want to make a quick and easy meal. Your phone is at 50% charge and the 'Pantry Plan' app is unopened.

1. Open the app
2. Click on the 'Meal Plan' tab located at the bottom of the screen
3. Click on your chosen meal
4. Read the ingredient list and instructions
5. Cook!

Requirements based of our task scenario	
Functional Requirements	• Allow users to add/delete food items • Track quantities and expiration dates of food • Allow users to select meals based on available items
Non-Functional Requirements	• Load each page within 4 seconds
Usability, Legal, Security and Ethical Requirements	• New users should be able to navigate without training • Meal suggestions should be easily accessible from main screen • Comply with GDPR • Allow users to delete account • Meals should avoid promoting unhealthy eating • Will not promote waste

Product description

Pantry list

No home screen so its quicker access and less buttons to press to get to the feature you need. This page shows what items you have in your pantry with thir expiry dates, as well as having options to add items manually or using a barcode scanner.

Recipe book

Contains all your recipes, you can add more manually or by uploading them from your files. Each recipe card has a photo, title and allergens list.

Meal planner

This screen shows the users nutrition for meals they have logged that day and shows recipes from those that are saved. It also contains Tips about how to have a balanced and healthy diet.

Requirements

Pantry List

- Searchable
- Notifications about expiry
- Order by state and expiry date
- Stats about wasted food
- Sharing a pantry list with other people (e.g. members of the same household)

Recipe Book

- Recipe creation
- Recipe sharing + importing
- Guided cooking steps
- Searchable (Filterable)
- Intolerances
- “Type” (Breakfast, Lunch, Dinner etc)
- Other relevant tags

Meal Plan

- Meal recommender system
- Tailor meals by diet requirements e.g. aiming for high protein diet
- Reminders about cooking food
- Pie chart of your diet (using the 5 food groups)
- Overview of diet in terms of nutrients and macros
- Info about importance of healthy diet

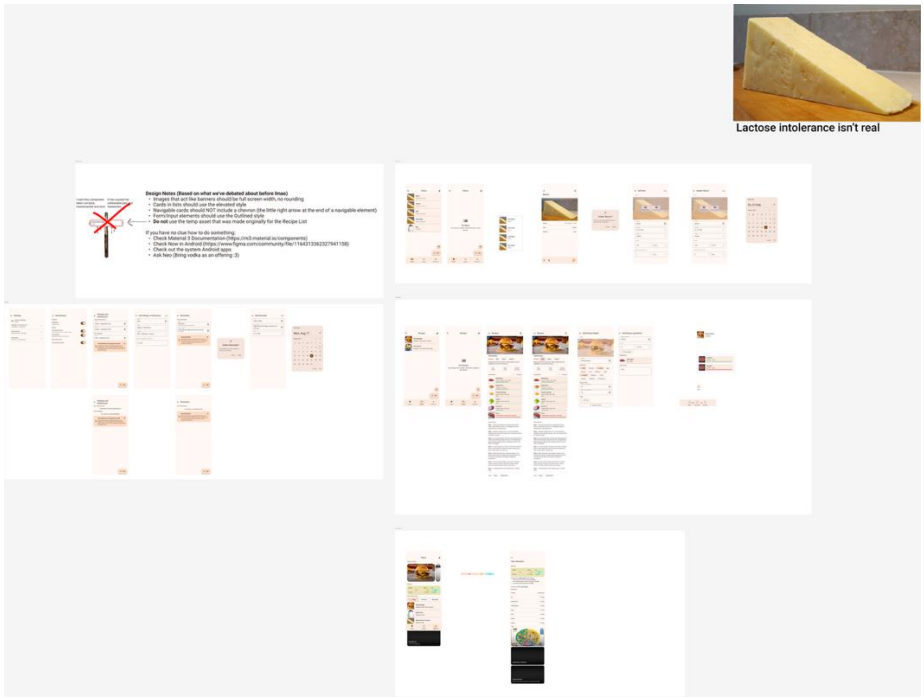
Settings (Android apps use a menu in the top bar)

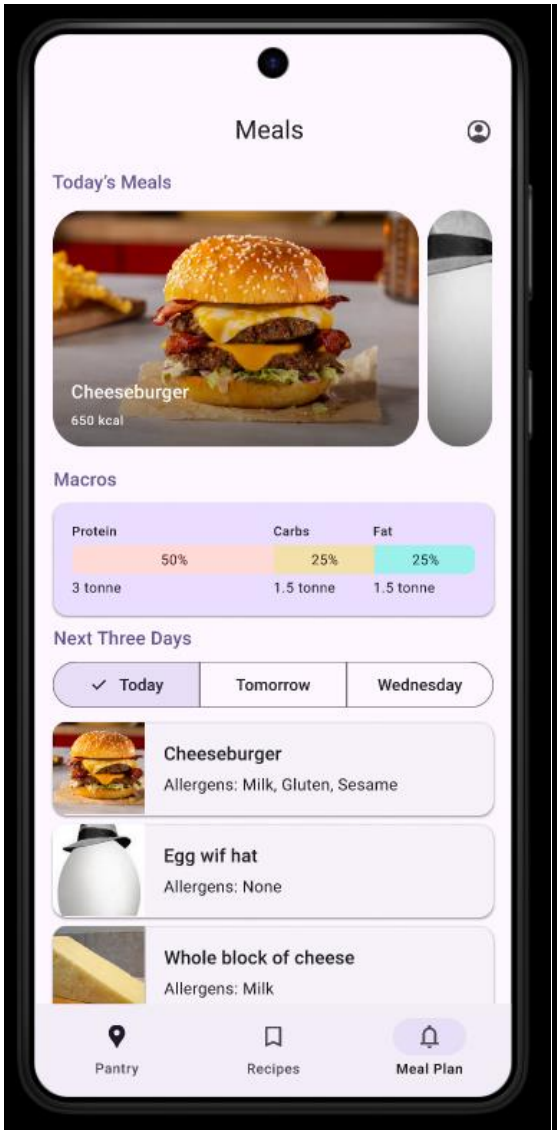
OS Integration

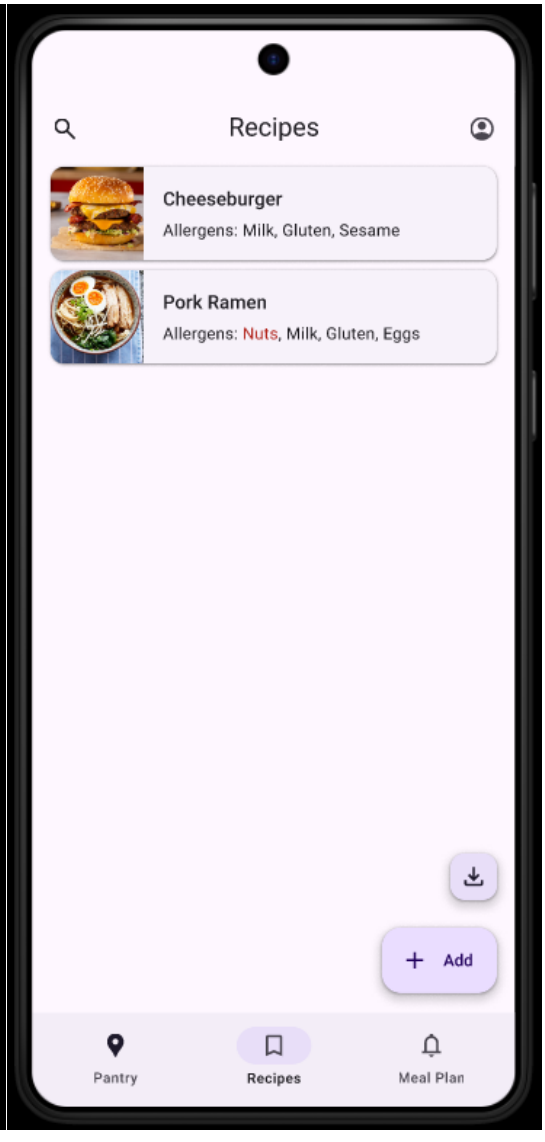
- Notifications
- Widgets
- Material You?
- Google Assistant Support?
- Health app support?

Designs

We spent a while in the design process and tried a few different designs and decided to use the designs shown below. We wanted it to be intuitive, so users don’t need to spend time learning how to use the app or struggle to find what they want, this is usually a big barrier for people using new applications as most people do not want to spend time learning hoe to use it. It is especially important for us as the goal of our app is to be efficient.


Lactose intolerance isn't real


Meals
Today's Meals
Cheeseburger
650 kcal
Macros
Protein
Carbs
Fat
50%
25%
25%
3 tonne
1.5 tonne
1.5 tonne
Next Three Days
Today
Tomorrow
Wednesday
Cheeseburger
Allergens: Milk, Gluten, Sesame
Egg wif hat
Allergens: None
Whole block of cheese
Allergens: Milk
Pantry
Recipes
Meal Plan


Recipes
Cheeseburger
Allergens: Milk, Gluten, Sesame
Pork Ramen
Allergens: Nuts, Milk, Gluten, Eggs
Add
Pantry
Recipes
Meal Plan

Implementation

To implement our app design, we used Android Studio making use of the Kotlin programming language which allowed us to implement the UI as well as build the backend that would cause the UI to update as values are changed/added across the app (i.e. new recipe created). While creating the UI we used the Figma board created as a base which made for a smooth conversion since both platforms use Google's official Material 3 UI library. This meant that UI elements could be accurately converted over especially with the use of developer mode which had code samples for each UI element on how to write it for layout in Jetpack Compose which handles the UI framework. Other APIs we used within Android Studio would be Coil for image loading, Hilt for dependency injections and Room for handling the database that stores aspects such as the pantry/recipe items that can then be loaded and edited through the app by the user.

To collaboratively work on the project all together simultaneously we made use of GitHub to create branches that each person could work on while other sections of the project were being developed. These could then be reviewed by the members managing the project through pull requests to approve changes and comment on needed changes before getting merged into the main branch. Doing it this way meant we avoided heavy merge conflicts while still making it so each member could program and test their own additions to the application without issue. By building up the project overtime with these small branch cycles along with following Android's best practice of :feature and :core modules we could efficiently include more features into the app as time progressed. These could then be tested through the built in Android Studio emulator to see if our changes persisted and everything in the app could be navigated in between.

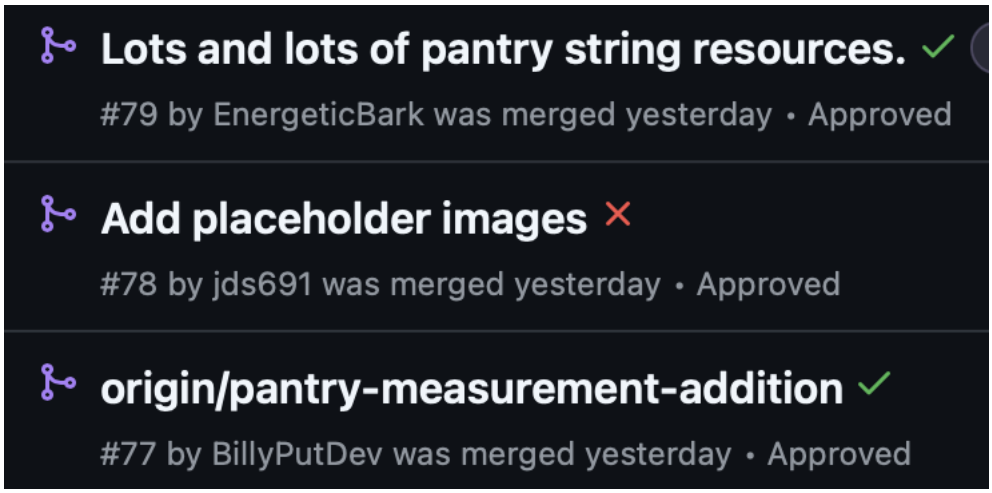
Testing

For testing our application, we made use of Android's instrumented tests and GitHub Actions. Instrumented tests allow us to perform unit tests on a real or emulated Android device and is very useful for UI testing or testing API that is specific to Android. It is an invaluable resource that allowed us to determine very quickly if key parts of the application ever stopped working due to another change we had made elsewhere. It also caught a few compilation errors as the tests would always fail if they could not be compiled.

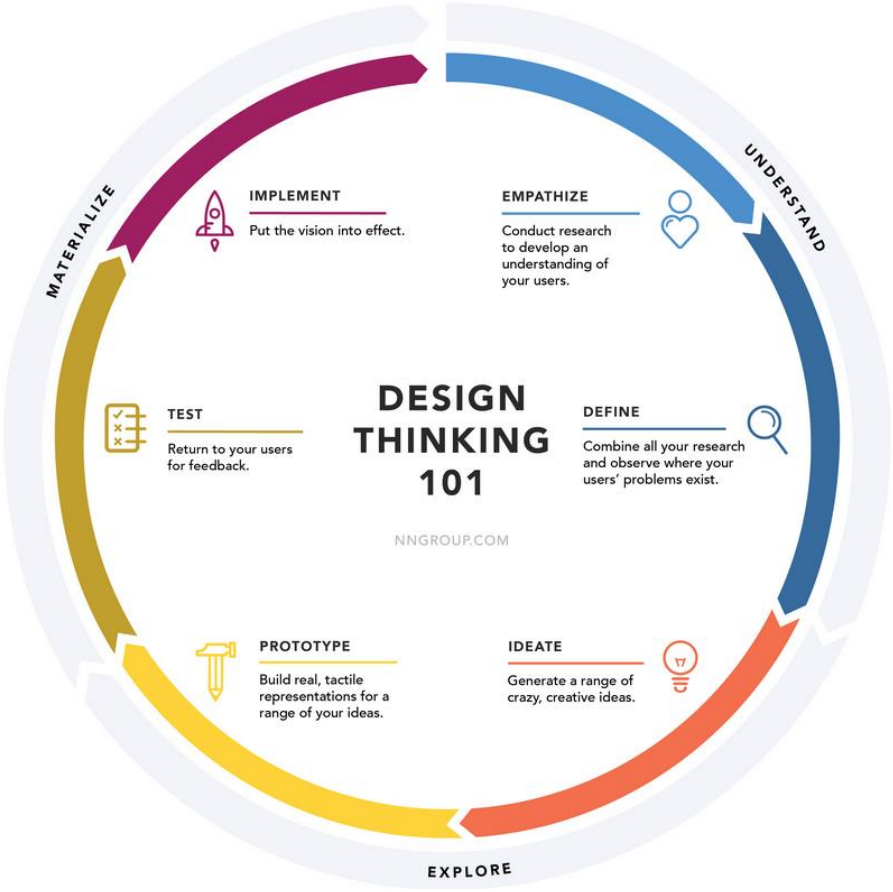
However, testing is a continuous process. They are not particularly useful if we must remember to manually run the tests every single time we make a change. That's where GitHub Actions comes in. GitHub Actions is a form of continuous integration (CI) which automates the testing process and other tasks that should be run on set intervals or when set events occur e.g. opening a pull request.

We defined an action that would automatically run the instrumented tests whenever a change was pushed to main or a pull request was being opened. This allowed anybody

to ensure that their changes always worked when attempting to merge to main. In the screenshot below:



It is easy to see directly in GitHub if a pull request can be successfully merged to its target branch or if someone has needed to skip the tests for one reason or another. It allowed us to remain automatically accountable for the changes we've made and reduce the need for changes to be manually checked for compatibility.



Legal and Ethical

Organizational ethos

We want our app to make good, healthy meals accessible, without paywalls. Easy to use recipes and support/info on good diets with reminders should help to improve people's wellbeing, especially those who struggle to do this, for university students many may forget to look after themselves properly as their priorities lie elsewhere, which is understandable, but we believe that having good food should not come at the cost of other stuff. We want this app to be user first and do not expect to make any money off it. If we did want to then we would use non-invasive ads so that the users experience is not impacted or slowed down.

Educational: easy to access information about how to eat healthy and why, cooking instructions and information about intolerances and allergies.

Security

Our app is all offline so there is no security features needed, we thought about adding a log in system, however we decided against this due to it taking longer to get onto the app each time. People wouldn't want to use it if every time they must log in as it takes too long.

Ethical

Encouraging eating healthy and sustainably, with ethical sourcing. Included information about intolerances, allergies and anaphylaxis (if someone is majorly allergic to food then do not buy so there's no cross contamination). Has information of the importance of having a healthy diet and also doing exercise (although good separately, both are needed).

The accessibility regulations came into force for public sector bodies in 2018. They state that you must make your mobile app more accessible by making it 'perceivable, operable, understandable and robust'[3]. We are not a public sector body so although we do not need to comply to these standards, we will try to make our application as accessible as possible and hold our app to a high standard.

[3] (The Public Sector Bodies (Websites and Mobile Applications) (No. 2) Accessibility Regulations 2018)

Critical review

Originally, we had planned to use a recommendation system that would recommend meals to the users, however we didn't have the time or data to do this. The recommendation system would have recommended meals based off different attributes like time until expiry and preferences, we could have expanded it by making the system learn about the eating habits of the user.

If multiple people in a household, then the system could recommend meals based off what everyone likes so you don't have to spend ages deciding, this would be especially helpful for people with children who are picky eaters because the system would be able to calculate the best meal that takes everyone's needs and preferences into account.

We worked well as a team, and we hit the majority of our goals; we are proud of the final result and are happy with how it turned out. We are especially pleased with the efficiency of the app as this was one of our primary goals and we feel it is more efficient than we had originally thought. This is due to the applications speed and ability to add items using the barcode scanner, The layout and design is also a major factor in the success of this product as it is easy to use and looks professional.

During this project we learnt a lot about project management and different development tools that we hadn't used before, for many of us, this was our first time using android studio which was challenging. In the future we will be much more confident when working in groups and making mobile apps, as this project has given us many important skills.

If we were to do this again, we would make sure we hit the original timeline goals. In the future, if we wanted to expand the app, then we would add the recommendation system as this would have a huge positive impact on the project.

References

Fallahkhair, S. and Cox, K. (2025). *Lectures*.

Mealime (n.d.). *App*. [online] Available at: https://www.mealime.com/ .

The Public Sector Bodies (Websites and Mobile Applications) (No. 2) Accessibility Regulations 2018.

Appendix

Appendix 1: GitHub code repository: https://github.com/jds691/CI536-Group-Project

Appendix 2: Video file (needs to be downloaded first): https://github.com/jds691/CI536-Group-Project/blob/main/submission/recording.mp4

Appendix 3: Record of team meetings

Date	Time	Who attended	What we did	Take aways	Notes
6.2.25 (lab)	2h	Seth, Ellie, Neo	Brainstormed ideas (in google docs) and decided to do a pantry plan)	Need to set up at time outside of labs when we meet.	
13.2.25 (lab)	2h	Seth, Ellie	Started report, GitHub, Trello, Meeting Records.	Using Kotlin or development	Put stuff on the github planning tool
17.2.25 (House)	4h	All	All on the same page, specifications so can divide work evenly		
20.2.25 (Lab)	2h	Seth, Ellie, Neo, Billy	Figma	Need to finish Figma so we can start code	
24.2.25 (House)	2.5h	Seth, Ellie, Neo, Billy	Figma		Report shared with everyone
27.2.25 (lab)	2h	Seth, Ellie, Billy, Neo	Figma		
6.3.25 (Lab)	2h	Seth, Ellie, Layton	Figma		No house meeting due to unavailability and sickness. Layton joined group
10.3.25 (House)	2h	Seth, Billy, Neo, Megan	Individually working on different parts of project		
13.3.25 (Lab)	2h	Ellie, Neo, Seth, Megan	Individually working on different parts of project, giving the rest of the group updates on where we all are.		
20.3.25	2h	Ellie, Seth	Individually working on different parts of project, giving the rest of the group updates on where we all are.		No house meeting this week
27.3.25	2h	Seth	Continued to work on the code		Last one before easter
24.4.25	2h	Ellie, Billy, Seth, Neo	Individually working on different parts of project, giving the rest of the group updates on where we all are.		First session back after easter
28.4.25 (House)	2h	Billy, Megan, Neo, Seth, Layton	Individually working on different parts of project, giving the rest of the group updates on where we all are.		
1.5.25	2h	Seth, Ellie, Billy, Neo, Layton	Individually working on different parts of project, giving the rest of the group updates on where we all are.		
5.5.25 (house)	2h	Neo, Ellie, Billy, Seth	Individually working on different parts of project, giving the rest of the group updates on where we all are.		
8.2.25 (lab)	2h	Ellie, Megan, Billy, Seth, Neo	Individually working on different parts of project, giving the rest of the group updates on where we all are.		
12.5.25	2h	All	Spoke about final additions we needed and what needed to be done to finalise the project. Worked on code and report.	Finish report Finish app (assigned final cards/tasks off GitHub board)	Final meeting at house before we have to demonstrate our app in the Thursday lab session
15.5.25	2h	Seth, Ellie, Layton	Finalised project and demonstrated our app on an android phone to our lecturer.	We are happy with how it turned out.	

Overall attendance (work was done outside of meetings):

Seth = 18/18 = 100%

Ellie = 16/18 = 89%

Neo = 13/18 = 72%

Billy = 11/18 =61%

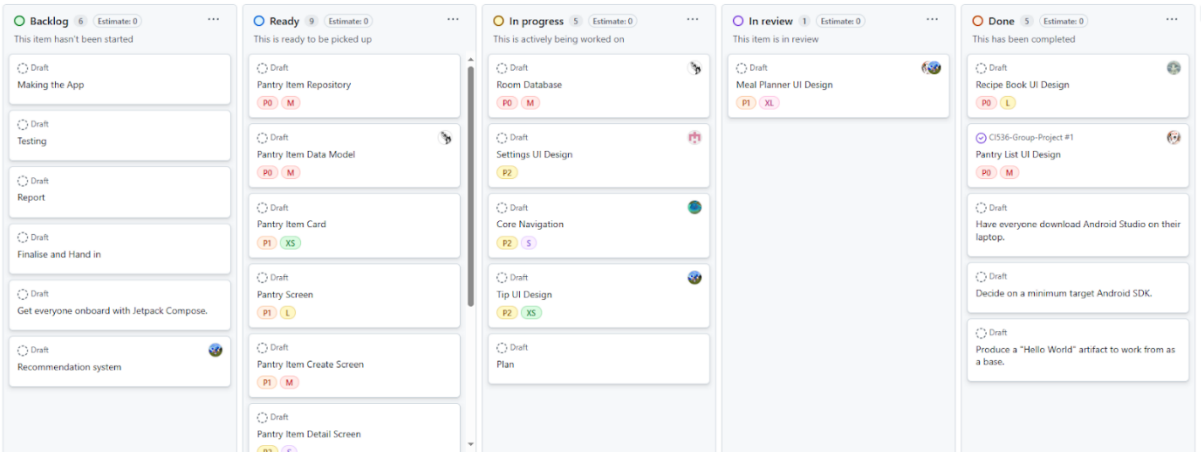
Megan = 6/18 = 33%

Layton = 5/12 = 42%

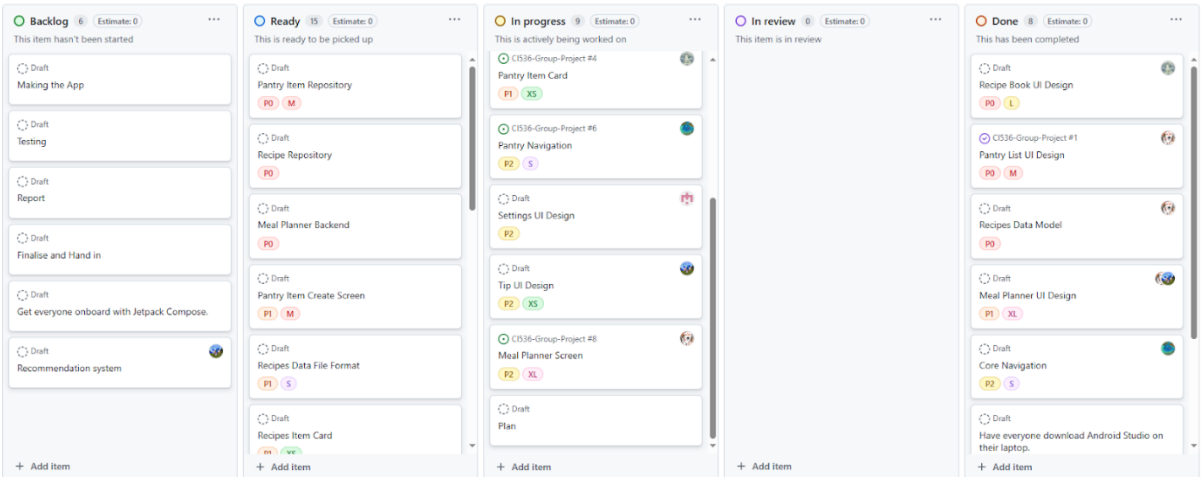
Average = 66%

Appendix 4: GitHub Project board screenshots

20.03.25



24.04.25



15.05.25