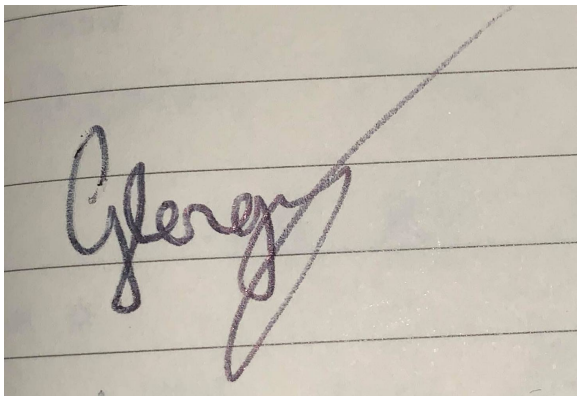
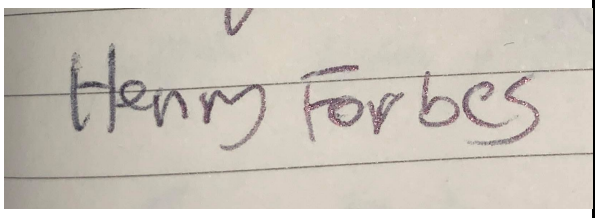
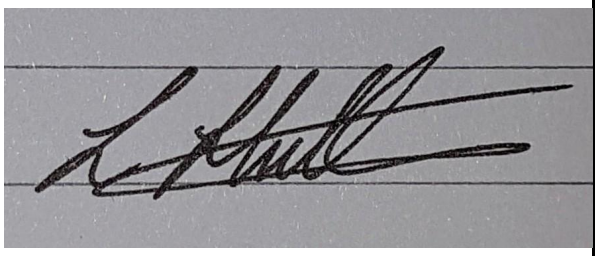
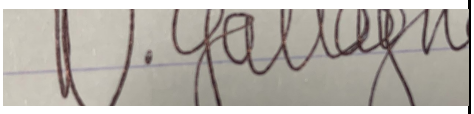
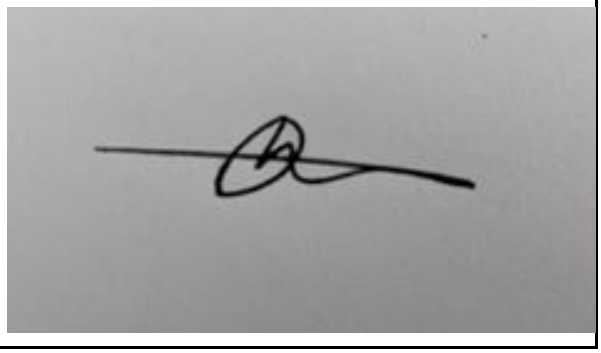
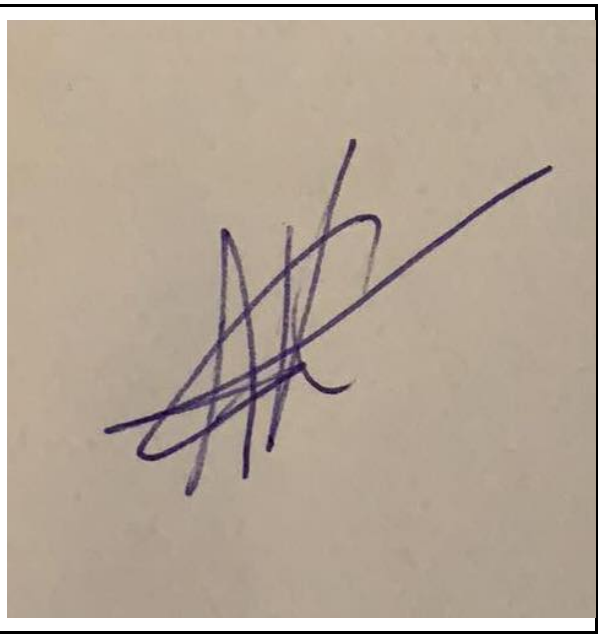
Name	Signature
Glen Griffiths	
Henry Forbes	
Laura Hulley	
Niamh Gallagher	
Olivia Adams	

Anton Kavaleuski	

Project Description

Target Market

We have created an app named Skük, we have created this app for students, with the average age for students being under 30 (UniversitiesUK, 2015). It is aimed at students due to the fact they have limited time due to their studies and other activities i.e. part-time jobs/sports. Through our research we have found students don't often enjoy spending time on shopping or planning meals. Through our app we are providing a simple and affection solution to meal planning, where students can go and find suggested recipes based on items in their pantry. (Anic, 2017)

Students tend to have limited space in their cupboards/fridge, so this app means that students are buying what's necessary to go towards their recipes, so they are making efficient use of what is in their cupboard. Students also shop less frequently, living near smaller shops, so with the app they would hopefully be more willing to go shopping as they can go for their certain items. - This information is gathered from interviewing fellow students, but more research is needed.

Mission Statement

To provide a recipe app which makes shopping and meal planning more efficient and enjoyable.

Mission Objectives

- To maintain (enter, update and delete) data on recipes
- To maintain (enter, update and delete) data on users
- To maintain (enter, update and delete) data on user's current ingredients
- To maintain (enter, update and delete) data on user's shopping list
- To maintain (enter, update and delete) data on all ingredients
- To perform searches on recipes
- To perform searches on ingredients
- To utilise a barcode scanner
- To utilise a login function
- To report on current usage statistics
- To report on recipes (most used, most reviewed)

Statement of Deliverables

User views and Requirements

- Description of anticipated documentation -
    - In our documentation we will include an installation guide, which will simply be instructions in how to install the app. This will carry on with a User Manual, which will document how to use the app properly. This will include clear descriptions of doing certain things within the app e.g. scanning items barcodes and adding it to your pantry. We will also include a separate section in the manual for administrators, which will just have specific instructions for what the admins can do.

- Description of anticipated software -
    - Skük, is simply a recipe app. To be more complex our app has various features, such as a database of stored recipes, a pantry which is a user specific database storing the items that the user has, and they can add/remove their food items to. To add items to the pantry the user can either do this manually or there will be a barcode scanner and they can scan their items and it will recognise that item and add it to that to the pantry. This scanning can be done whilst shopping and based on what you are adding the app can recommend adding certain items to complete a whole meal.

- Description of any anticipated experiments -
    - We will experiment the app by having select recipes in the app. We can put which ingredients we have in our pantry and if ingredients match the recipe, we should get a match. We will also test various different items to see if the scanner works and that it adds to the pantry properly.

- Description of methods for evaluation of the work

  We'll use criteria-based assessment in order to evaluate the software we have produced. The list of the criteria we will assess is also used from the Software Sustainability Institute (Mike Jackson 2011). By making sure all the criteria are accounted for, we can make sure that our software is easy to use and also easy to maintain in the rapidly changing tech industry. It will be essential to try and make sure these criteria are met as their implemented in order to avoid any backtracking.

  Some criteria will have to be tested by different groups. For the usability criterion, we'll have to measure it with our target audience in order to make sure that they are able to use it having never faced it before. Buildability is the exception where that will be assessed by us as it is not the user documentations.
  This is how we will assess each sub-criterion for usability
    - Understandability: – We will give the focus group the app and ask them whether they found the program easy to use.
    - Documentation: Give the user the user documentation and ask them on a scale of 1-10 with questions such as "How easy it was to

understand? Was it comprehensive? Was there anything unnecessary? Etc."

- Installability – Have the focus group try and install their app onto their phone and see how easy it was for them by asking them to give a scale of 1 – 10
- Learnability – Have the focus group try and use the app and see whether they are able to quickly learn the functionality of the app. Ask them whether they were easily able to navigate after a short while to know where everything is.

Criterion	Sub-criterion	Notes – to what extent is/does the software…
Usability	Understandability	Easily understood?
	Documentation	Comprehensive, appropriate, well-structured user documentation?
	Buildability	Straightforward to build on a supported system?
	Installability	Straightforward to install on a supported system?
	Learnability	Easy to learn how to use its functions?

(Mike Jackson, 2011)

The user will be uninterested in the sustainability and maintainability of the system. It would be better if we evaluated these criteria as a user wouldn't be interested or may not have the knowledge available to judge these criteria.

- Identity – Background research, making sure that we have a product that is differentiated enough from other services and communicates that well.
- Testability – By being able to complete unit testing and system testing quickly, it would prove that the device is easily tested.
- Analysability – If we are all able, as a group, to understand the source code for each other, then this will verify the analysability of the code that we have created.
- Changeability – To assess changeability, we need to be able to be able to easily change code without creating any errors.
- Interoperability – The intended system is meant to work using a database, phone app, barcode scanner and machine learning. If none of those work together, then we can evaluate that the system is not interoperable which would basically mean the system is not successful.

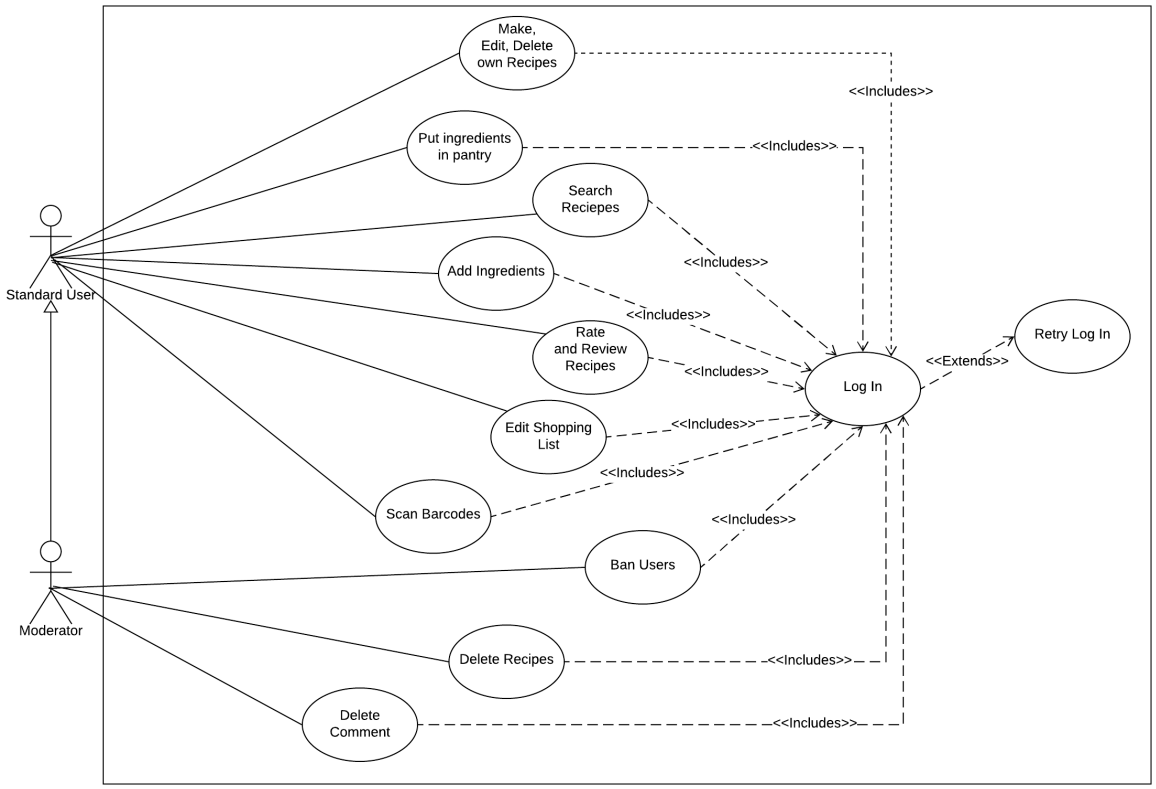
Sustainability and maintainability	Identity	Project/software identity is clear and unique?
	Copyright	Easy to see who owns the project/software?
	Licencing	Adoption of appropriate licence?

(Mike Jackson, 2011)

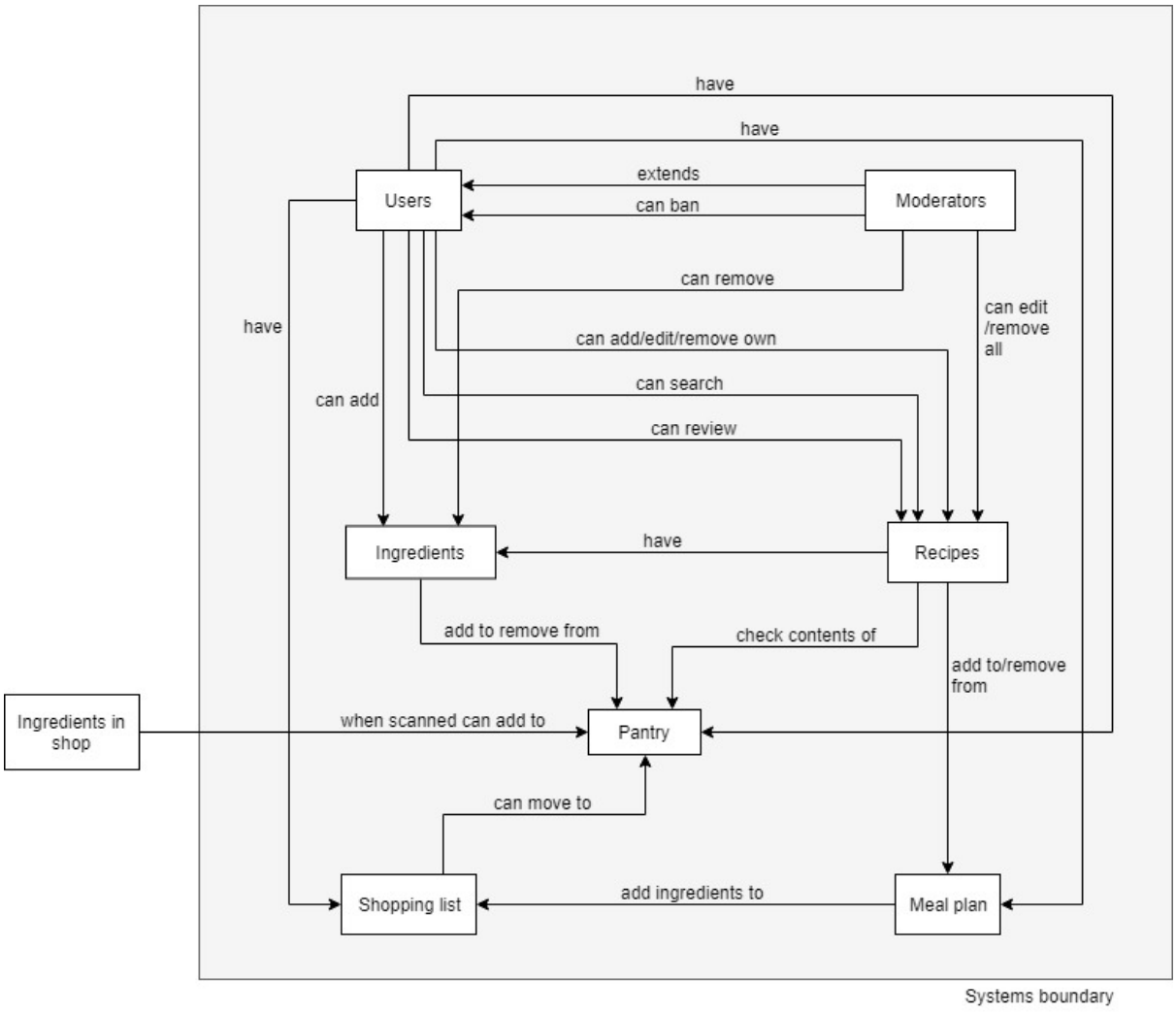
	Governance	Easy to understand how the project is run and the development of the software managed?
	Community	Evidence of current/future community?
	Accessibility	Evidence of current/future ability to download?
	Testability	Easy to test correctness of source code?
	Portability	Usable on multiple platforms?
	Supportability	Evidence of current/future developer support?
	Analysability	Easy to understand at the source level?
	Changeability	Easy to modify and contribute changes to developers?
	Evolvability	Evidence of current/future development?
	Interoperability	Interoperable with other required/related software?

(Mike Jackson, 2011)

Use Case Diagram



System Boundary Diagram



(Connolly & Begg, 2004) used as reference

Conduct of the Project and Plan

Background Research

As a means of framing the problem and developing solutions we conducted thorough research, so we could fully understand the needs of the user, products currently available to the user, and how we could implement our written solution computationally.

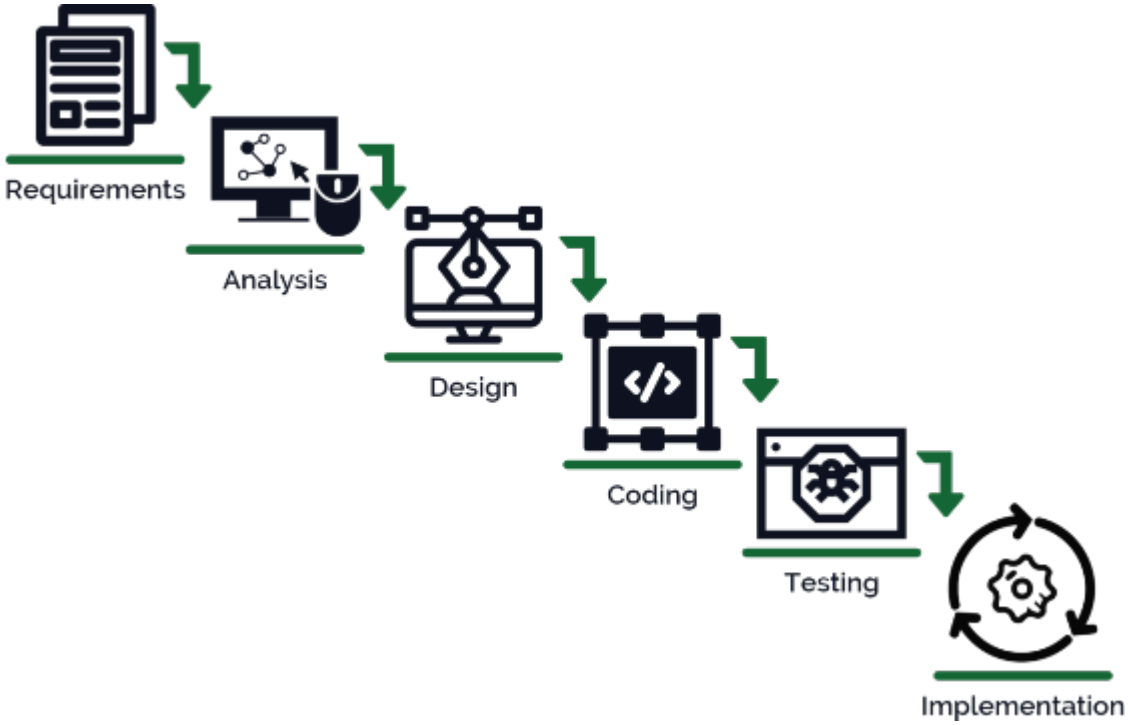
Starting off with the idea of having an app that suggests recipes for you given what ingredients you currently have available, we began to look into the likely users of the app. Being students ourselves, we recognised that meal planning is a difficulty the majority of us contend with and that we would benefit from an app like this. Therefore, we decided to conduct some research into what students would find useful in an app of this sort. We spoke to numerous students about what they find most difficult about meal planning. Conducting this research made it clear to us that students want a quick and easy way to plan their meals for the week, and subsequently what they need to buy. This research is how we reached the solution of combining the two within our app.

Following this revelation, we explored apps currently available to users. We found a number of popular apps where you could explore recipes - for example Tasty (tasty, n.d) - and only one app, named Mealime (Mealime, 2017), that created a shopping list from a meal you chose to cook. In order to create a unique, but still highly useful, experience for our users we decided to incorporate a scanner into the app, that you can use while shopping, as well as the ability to post your own recipes and review others'.

Our decision to incorporate a scanner into the app resulted in our research into the implementation of a scanner. After looking at a number of examples on GitHub, and exploring Swift, we came to the conclusion that we would use the Swift framework called AVFoundation to implement our scanner.

Through our research, we concluded that our app must offer an easy way to plan your meals for the week, your shopping list for the week, as well as, the unique and enjoyable aspects that Skük will offer. When moving forward we will research more into the implementation of the scanner.

Design Stage



To ensure we deliver our project on time we will be using the Waterfall method. We are limited to our timings on the project so by using Waterfall we have set stages which we can stick to.

Implementation Stage

The software used to create that app will be Swift, so therefore this app will only be available for Apple phone devices. Swift has essential libraries that we can use to make our app work, including one which allows for barcode scanning; if we cannot implement the barcode scanner ourselves, we will find an open-source API which is available to use through Swift.

The hardware used will be Apple iPhone devices. Three of the team members own iPhones, therefore the testing can be done using their personal devices.

Risk Assessment

Potential Risks	Description	Risk Level	Possible Solutions
The requirements and the system specification proposed are too ambitious.	Due to the lack of experience of the team with these kinds of projects the estimates of what can be produced in the time given can be incorrect.	Medium	Potentially cut down on some of the functionality proposed in order to deliver the product on time.
Issues storing the recipe methods and ingredients with in the database	Several issues were brought up with regards on how to store the method of the recipe and the ingredients for it within the database,	High	Store the methods separately, outside the database.
Difficulty in creating a working algorithm for recipe recommendation	An working algorithm that would recommend recipes based on the ingredients available may be difficult to create.	Low	Attempt to find a similar open source algorithm that would be available to be used for our product.
Difficulty in linking the database to the user interface	Although creation of the user interface and database should be straightforward, connecting the two together and achieving full functionality may be problematic	Low	Simplify some of the functionality proposed to the level at which it can be achieved
The requirements and specification have missed crucial features	Certain features that will be necessary for the system to operate have not be thought of or covered in the requirements and hence will need to be implemented “on the go”	High	Allocate additional time to allow to resolve these issues as they come up
Issues over the languages chosen	The team's technical skills and language	Medium	Organise the team in such a way that

for the project	knowledge differs greatly. It may prove difficult to agree on which language should be used to create the project and not all team members maybe confident in using said language.		each person's skills are best suited to the task.
Issues with the barcode scanner	We are planning to use an open source barcode scanner. It has not been tested and may not be effective for our project	Low	Find an alternative open source barcode scanner
Issues with classifying ingredients	Several issues were brought up regarding the translation of the ingredients' barcode to the entity stored in the database. Extracting the correct product from the barcode can prove challenging	Medium	Additional manual method of ingredient addition will be implemented

Low - Unlikely that this will occur during the life of the project
Medium - There is a 50-50 chance that this will occur during the life of the project
High - Likely that this will occur during the life of the project

Major Challenges

As this is a completely new project for the team a number of new skills will need to be required for this task. Creation of a well working database is one of the main challenges of the project. Several of our team members possess extensive theoretical knowledge of database creation and management and can assist the team in putting this knowledge into practice.

Another one of the major challenges that we will face is getting the scanner programmed right, making sure it is doing what it is supposed to. A similar problem we may face with the scanner is that it works for all of the barcodes that are being scanned and that it will be able to recognise which products they are to be checked off the Shopping List.

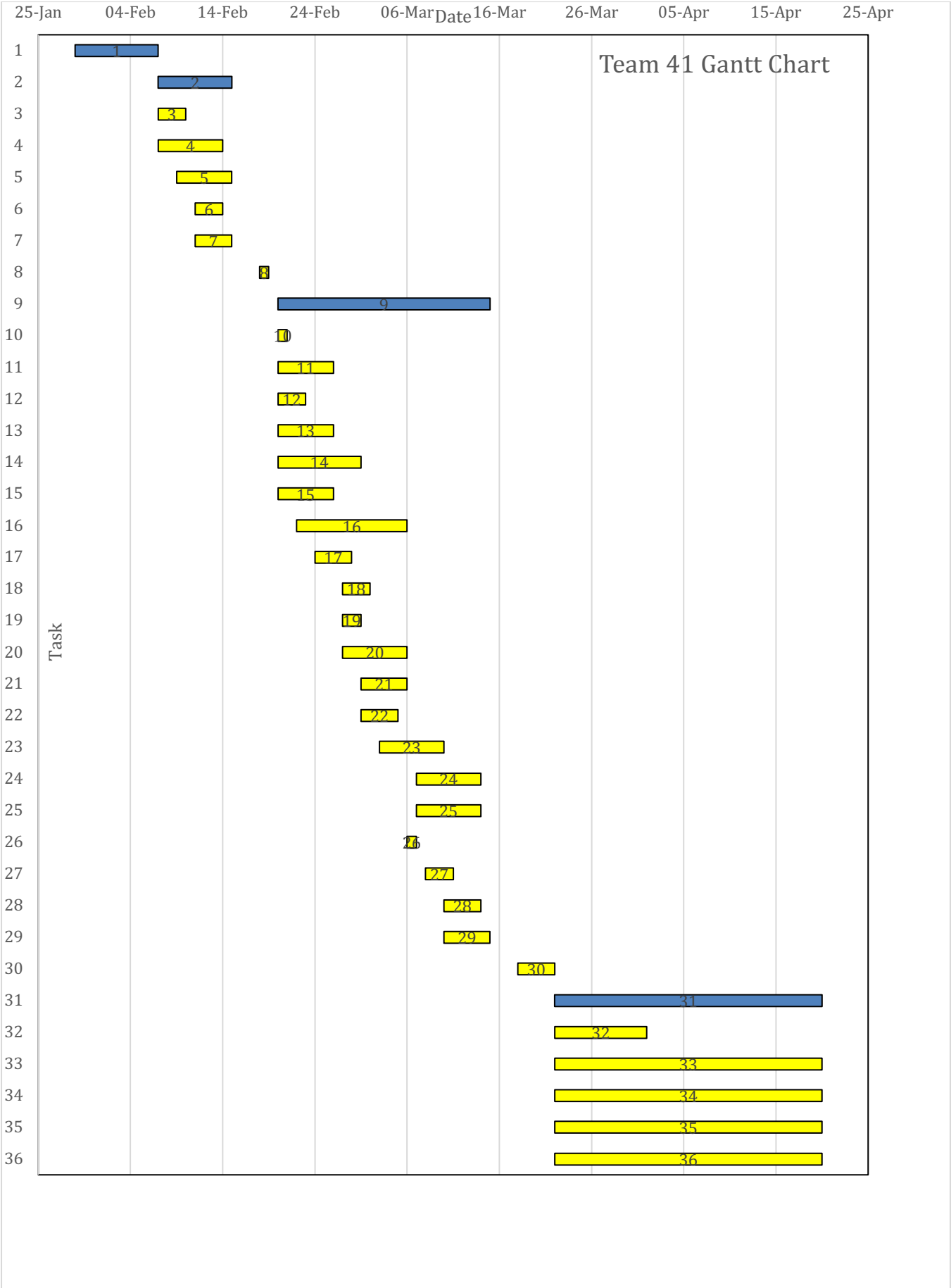
We will also face challenges with connecting recipes and ingredients to create the shopping lists and check each user's pantry for the items they may already have.

Skills Required

- App development is a skill that will be required for this project. Luckily a large portion of the team is taking an app development module which will provide them with the skill necessary to fulfil this task.
- We will need good presentation skills to make sure that all of our documents are presented correctly and can be easily followed, which we have used for previous projects, so we can transfer our previous skills over.
- Database design and SQL in also quite a major skill that we require, luckily one of our team members really enjoys SQL which helps us a lot as he has a lot of knowledge in that area.
- The final skill we feel is essential is some knowledge of machine learning. We had some information about machine learning in our AI module in first year, but also most of our team members took Advanced AI last semester which went into more depth of machine learning, which will be vital help to our project.

Gantt Chart

	Tasks				
Task No.	Start Date	End Date	Description	Taskmaster	Duration(Days)
1	29-Jan	07-Feb	Decide on a software idea	Everyone	9
2	07-Feb	15-Feb	Requirements document	Everyone	8
3	07-Feb	10-Feb	Project Description	Glen, Laura	3
4	07-Feb	14-Feb	Statement of deliverables	Henry, Glen, Niamh	7
5	09-Feb	15-Feb	Conduct of research and plan	Livi	6
6	11-Feb	14-Feb	Risk Assessment	Anton	3
7	11-Feb	15-Feb	Gantt Chart	Henry	4
8	18-Feb	19-Feb	Prepare presentation for requirements	Everyone	1
9	20-Feb	15-Mar	Design document	Everyone	23
10	20-Feb	21-Feb	Summary of proposal	Glen	1
11	20-Feb	26-Feb	A description of anticipated components of system	Anton	6
12	20-Feb	23-Feb	UML Class	Livi	3
13	20-Feb	26-Feb	Storyboard for UI design	Niamh, Laura	6
14	20-Feb	01-Mar	Research for algorithms	Laura	9
15	20-Feb	26-Feb	Entity relation model	Henry	6
16	22-Feb	06-Mar	Unit Tests	Glen	12
17	24-Feb	28-Feb	Interaction Chart	Livi	4
18	27-Feb	02-Mar	Process Map	Anton	3
19	27-Feb	01-Mar	Logical Table Design	Henry	2
20	27-Feb	06-Mar	Evaluation	Niamh	7
21	01-Mar	06-Mar	Boundary	Livi, Glen	5
22	01-Mar	05-Mar	Sample Tables	Henry	4
23	03-Mar	10-Mar	Methods of Barcode Scanner	Anton	7
24	07-Mar	14-Mar	System Tests	Glen, Anton	7
25	07-Mar	14-Mar	Prepare document for presentation	Niamh	7
26	06-Mar	07-Mar	Data Dictionary	Henry	1
27	08-Mar	11-Mar	SQL Pseudoqueries	Henry	3
28	10-Mar	14-Mar	Gantt Chart	Henry	4
29	10-Mar	15-Mar	Review of plan	Everyone	5
30	18-Mar	22-Mar	Prepare presentation for design	Everyone	4
31	22-Mar	20-Apr	Implementation	Everyone	29
32	22-Mar	01-Apr	Learn software	Everyone	10
33	22-Mar	20-Apr	GUI Implementation	Everybody but Livi and Henry	29
34	22-Mar	20-Apr	Database implementation	Henry, Anton	29
35	22-Mar	20-Apr	Scanner implementation	Livi, Glen, Niamh	29
36	22-Mar	20-Apr	Recipe suggesting implementation	Henry, Anton, Laura	29


Team 41 Gantt Chart
25-Jan
04-Feb
14-Feb
24-Feb
06-Mar
16-Mar
26-Mar
05-Apr
15-Apr
25-Apr
Date
Task
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36

Transaction Requirements

Transaction	**Users**
Make, edit and delete personal recipes	Standard user and moderator
Add ingredients to database	Standard user and moderator
Add/ remove ingredients to/ from shopping cart	Standard user and moderator
Add/ remove ingredients to/ from pantry	Standard user and moderator
Search Recipes	Standard user and moderator
Scan barcodes	Standard user and moderator
Rate and review recipes	Standard user and moderator
Log in	Standard user and moderator
Delete recipes	Moderator
Delete comments	Moderator
Ban users	Moderator

System Requirements

General

- The app should be clear and easy to navigate

Login

- Must have a Login function
    - Must ask for email and password
- Must have an option for forgotten password
- Must be an option to “register an account”
- The “forgotten password” button and the “register account” button should be easily visible from the login screen

Register

- Must have an option for new users to register
- Must check whether email is already in use
- Must request the name of the user
    - The name must only contain letters
- Must check whether password fits within constraints:
    - Password must be at least 8 characters long
    - Password must contain at least one uppercase and lowercase letter, number and symbol
    - Password must be case sensitive

- Must check to see if the confirm password matches the original password
- Should be easy for the user to see whether they have entered a username which is already in use

Front page

- From here, users must be able to access:
    - Pantry
    - Shopping list
    - Meal plan
    - Recipe bank
    - Account settings
- This must be the first page to load after the login screen
- Users must be able to navigate back to this page from any screen (can be via another screen)

Account Settings

- Must have an option to change email
- Must have an option to update password
    - Must enter previous password
    - Enter new password and confirm password, must follow above password constraints.
- Must have an option to log out of the account

Pantry

- Must be able to add new items
    - Through scanner
    - Or through manual input
- Must have access to scanner through this screen
- Must be able to edit items in the pantry

System Requirements **(Implemented)**

General

- The app should be clear and easy to navigate

Login

- Must have a Login function
    - Must ask for email and password
- Must be an option to “register an account”
- “register account” button should be easily visible from the login screen

Register

- Must have an option for new users to register
- Must check whether email is already in use
- Must request the name of the user
    - The name must only contain letters
- Must check whether password fits within constraints:
    - Password must be at least 8 characters long

    - Password must contain at least one uppercase and lowercase letter, number
    - Password must be case sensitive
- Must check to see if the confirm password matches the original password
- Should be easy for the user to see whether they have entered a username which is already in use

Front page

- From here, users must be able to access:
    - Pantry
    - Shopping list
    - Meal plan
    - Recipe bank
    - Account settings
- This must be the first page to load after the login screen *(Front page has changed to Meal Plan)*
- Users must be able to navigate back to this page from any screen (can be via another screen)

Pantry

- Must be able to add new items
    - Or through manual input
- Must have access to scanner through this screen
- Must be able to edit items in the pantry

Bibliography

Anic, P., 2017. What do students do in their free time and why?. *Polish Psychological Bulletin,* 48(4), pp. 504 - 515.

Connolly, T. M. & Begg, C. E., 2004. *Database solutions: a step-by-step guide to building databases.* 2nd ed. Harlow: Pearson Education Ltd. / Addison-Wesley.

mealime, 2017. *mealime.* [Online]
Available at: https://www.mealime.com/
[Accessed February 2019].

Mike Jackson, S. C. a. R. B., 2011. *Software Evaluation: Criteria-based Assessment,* s.l.: Software Sustainability Institute.

tasty, n.d. *tasty.* [Online]
Available at: www.tasty.co.uk
[Accessed February 2019].

UniversitiesUK, 2015. Patterns and trends in UK higher education. In: London: UUK, p. 19.