DCUMunch

"More than just food."

CA326 Functional Specification:

DCU Munch

Joseph Adedayo 20303853 & Genesis Uwumangbe 20459666

Table Of Contents

1. Introduction

1.1 Purpose

This document outlines the technical description of our web-based application DCU Munch. It will explain further why the application is needed and what functions it has. It will be the document of reference for our CA326 Project.

1.2 Scope

DCU Munch is a food website application that is based on Dublin City University (DCU) Catering (DCU Cafeteria, Business Café and DCU Nubar). We saw that there was a need in the college for this application as students found it hard to find where to eat and not only find where but find nutritious foods to eat catered to their needs.

The goal is to provide students with a fun and enjoyable method of finding **affordable** and **healthy** food based on their dietary requirements and improving student health and well-being.

It will be available on all devices with browsers. The application will be vibrant, modern and exciting for students from all backgrounds. Students can log in and add all their dietary requirements and any useful information like sex, height, weight, activity level etc. The application will then show different meals catered to their choices.

DCU Munch will have a unique rating algorithm which will sort the food from DCU'S catering menus on the user's feed based on the user's information, if the user has liked the food or similar foods to it and if other users like or dislike certain foods etc. The price of each meal will also be shown, so the user can budget their spending.

Users will also be able to order and pay for food using the application. Students will then be given a confirmation number after successful payment and a QR code to scan at the food outlets to pick up the food when it is ready. This will help the cafeteria avoid long queues during rush hours.

There will also be sections to locate the different food places, contact and other useful information that students have been struggling to find in the past.

Finally, there will be an admin side of the application where DCU Catering can create its own profile, the aim of this side of the application is to make the process as easy as possible for admin users. Admins can upload their name, phone number and all relevant information. They can also upload all the meals in their menu, a drop-down menu will be

used where the catering can easily input their meals and we can translate those meals into their nutritional values like calories, sugars, and carbs to make the process fast and easy for the catering. Menus can be changed at any time if there is a slight change or even a menu revamp.

Affordable and healthy recipes for students who are living away from home and for those that have to cook for themselves will also be provided, it will be based on the students' food requirements. These recipes will have video tutorials and students can rate the recipes so DCU Catering knows which recipes work for students. The recipes section will use the same rating algorithm to decide what to be displayed to the user.

1.3 Glossary

DCU - Dublin City University
Users - we are normally referring to students
Admin Users - we are normally referring to DCU Catering
SQL - Standard Query Language

1.4 Business Context

DCU Catering is the business the application will be deployed on. DCU Catering feeds thousands of students, lecturers and staff every day. The main purpose of the application for DCU is to advertise their foods, find it easier for students to find their outlets for food and make the process of buying food easier, faster and less cost-effective.

1.5 References

[1] Google Sign-In integration system - [online] https://developers.google.com/identity/sign-in/web/sign-in [Accessed Nov 25 2022]

[2] MyFitnnespal API [online] - https://myfitnesspalapi.com/ [Accessed Nov 25 2022]

1.6 Overview

Section 2 is a general description of the product and its functionality. User characteristics and how we expect our users to react to the product which is shown by different use cases. The constraints of the system and operational scenarios will be discussed in this section also.

Section 3 will include all requirements of the system. All these requirements will be ranked from most important to least important, any technical issues will be discussed in this section.

Section 4 is the system architecture this will display the system overview and the organization of the different components of the system.

Section 5 is the high-level design where all design components will be expressed from the software design, UI, logic design and their associated diagrams.

Section 6 is the preliminary schedule which will be the plan for the project from the start date to the end of the project, including tasks that need to be completed. All important deadlines and information will be recorded here.

Section 7 is the change request management, how will we assess and examine suggested changes for the application.

Section 8 are the risks and assumptions where we examine things that could hinder the development of DCU Munch and some ways to resolve them.

Section 9 is the operational attributes which are the important qualities we would like the application to have and how to obtain those qualities.

The last section is the appendix.

2. General Description

2.1 Product / System Functions

User sign-in - When the user has entered the application on their local browser, they will be prompted to enter with their DCU Google email from there we will use the Google Sign-In integration system to get their name, profile image and a user id token will be created for each user. Once that is completed the user can then add their age, weight, sex, activity level, and dietary requirements to be attached to their profile. This information will be kept in our SQL database.

If the user is already located on the database they will not have to enter all this information.

User profile - the user will have a profile where they can change their profile image, update personal information and see their recent transactions.

User feed - the user will then be asked a series of questions to initialise the rating feed system and to give the system some data to work with so it will give an accurate relevant feed to the user. The feed will recommend foods at the top for breakfast, lunch and dinner based on the foods already consumed (E.g User had a heavy lunch, fewer calories for dinner)

Recipes - The next part of the feed will recommend recipes for the user, where the user can watch videos and read how to complete the recipes. Users can also rate the recipes the recipe will be on the same rating algorithm to keep the feed as relevant as possible to the user. The recipe can be added to the total calories consumed via the munch button at the end of each recipe. These recipes are created by the admins.

Order system - the user will be able to order food from different outlets in the college, DCU Nubar, Cafeteria and Business Café. Users can pay through a payment system after successful payment the user will be given an order confirmation with a QR code, time and where to pick up the food. The food consumed can be added to the total calories consumed via the munch button so the rating algorithm will know what food to recommend next. The recent transaction will also be added to the user's transactions page.

Payments - here the user can use their credit card/debit card to pay for each order. Once the meal has been paid for confirmation will also be sent to the food outlet chosen to prepare the food.

Admin profile - the admins will be able to log in to the system and add contact information and their menus. Admins can also add fun recipes for students to the database. The menu system will be done by an API where DCU Catering can input the food and the system will gather all the dietary information of that food to make the process easier and faster. Admins should be able to see all their transaction history and how much they have generated. They will also have a feature to see what recipes and foods the users like the most.

2.2 User Characteristics and Objectives

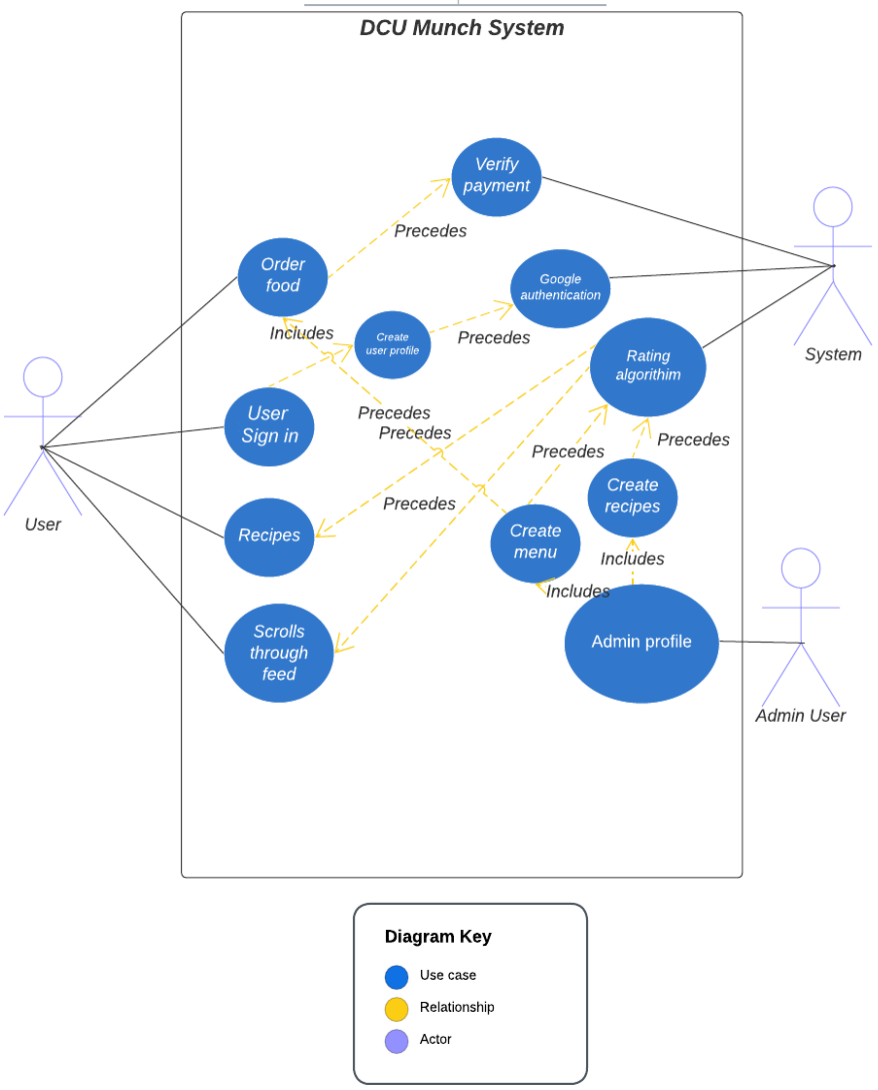
There are two users of the system admin users which are DCU Catering and normal users which are students. Admin users should be familiar with how to use a normal web browser application, one of the requirements is to make the system easy and fast for all those working in DCU Catering in line with their business objectives. The application should never feel like extra work for the staff. To do this we will have easy drop-down menus and simple sections to add or change the menus and recipes whenever needed. Once the admin users have entered all their menus, and recipes there should not be much work for them to do the only time the admin users should touch the application after the initial setup is to view orders, possibly add new recipes and change some elements in the menu.

Student users should have a device that can access the internet through a web browser. They should have a small knowledge of how to use a normal web browser application. The application should be simple, fast, fun and inclusive for all users in DCU as most users will be young adults. To achieve this we will design the application with nice vibrant colours, and the logic behind each function should be well-structured and tested. Finally, the speed of the application is a number one priority.

2.3 Operational Scenarios

Detailed here is a general use case diagram for all the features and separate use cases for different features and operational scenarios.

Use case diagram



USE CASE 1	User Register/Sign-in	
Goal in Context	Creating an individual unique profile	
Scope & Level	System, Core.	
Preconditions	User is creating a profile for the first time	
Success End Condition	User can sign in and create their user profile	
Failed End Condition	User cannot make user profile	
Actors	User, System	
Trigger	User enters their DCU Email	
DESCRIPTION	**Step**	**Action**
	1	User opens the DCU Munch Application
	2	Uses their DCU Email to sign
	3	Google sign in authentication checks their details
	4	Google information is used for the user profile
	5	User can now edit user profile image, name
	6	User can enter details about age, sex, weight, activiity level etc.
	7	User can look at the feed, order food, try out recipes etc.
EXTENSIONS	**Step**	**Branching Action**
	5a	User must be signed in at all times for the application to work
	2a	User cannot sign in if they have a problem with their email.

USE CASE 2	User feed	
Goal in Context	Swipe through different foods and recipes on the feed and rate them.	
Scope & Level	System, Core.	
Preconditions	User must be signed in	
Success End Condition	User can rate the recipes to tailor the algorithm to their needs.	
Failed End Condition	User cannot rate and the algorithm creates irrelevant feed	
Actors	User, System	
Trigger	User scrolls through feed	
DESCRIPTION	**Step**	**Action**
	1	User opens the DCU Munch Application
	2	User signs in
	3	Scrolls through feed
	4	User can rate meal or a recipe up to 5 stars.
EXTENSIONS	**Step**	**Branching Action**
	3a	User must be signed in at all times for the application to work

USE CASE 3	Ordering Food	
Goal in Context	Ordering, paying and receiving an order confirmation	
Scope & Level	System, Core.	
Preconditions	User must be signed in	
Success End Condition	User can order food and will receive an order confirmation to pick up the food	
Failed End Condition	User does not get an order confirmation	
Actors	User, System	
Trigger	User presses the order button	
DESCRIPTION	**Step**	**Action**
	1	User opens the DCU Munch Application
	2	Uses their DCU Email to sign in
	3	Users sees food that they would like to order
	4	User presses the order button
	5	User is prompted to pay
	6	After payment they are given the confirmation
	7	User can pick up the food at the various food outlets In DCU.
EXTENSIONS	**Step**	**Branching Action**
	3a	User must be signed in at all times for the application to work
	2a	User cannot sign in if they have a problem with their email.
	6A	User will not get confirmation if the payment is unsuccessful.

USE CASE 4	Admin User Registration	
Goal in Context	Creating an individual unique admin profile	
Scope & Level	System, Core.	
Preconditions	Admin user is creating a profile for the first time	
Success End Condition	Admin user can sign in and create their user profile	
Failed End Condition	Admin user cannot make profile	
Actors	User, System	
Trigger	Admin user enters their username and password	
DESCRIPTION	**Step**	**Action**
	1	Admin user opens the DCU Munch Application
	2	Admin user inputs their admin username and password
	3	Admin user has successfully signed in.
	4	Admin user can now edit user profile image, name, create menus, recipes and add important information.
	5	Admin user can look at their sales from orders, edit menu and recipes etc
EXTENSIONS	**Step**	**Branching Action**
	4a	Admin user must be signed in at all times for the application to work
	4b	Wrong username or password
	5a	Admin user must have a menu or recipe already saved to edit.

USE CASE 5	Creating Menu	
Goal in Context	Creating a menu that will be displayed on the feed for users	
Scope & Level	System, Core.	
Preconditions	Admin user has created a profile	
Success End Condition	Admin user can add all menu items	
Failed End Condition	Admin user cannot make menu	
Actors	User, System	
Trigger	Admin user enters create menu section	
DESCRIPTION	**Step**	**Action**
	1	Admin user opens the DCU Munch Application
	2	Admin user has successfully signed in
	3	Admin user enters create menu section
	4	Admin user can use the drop-down menu to enter common foods and add the prices.
	5	Admin user can save and publish menu
EXTENSIONS	**Step**	**Branching Action**
	3a	Admin user must be signed in at all times for the application to work
	5a	User must enter at least one menu item to save it and publish.

USE CASE 6	Creating Recipes	
Goal in Context	Creating a recipe displayed on the feed for users	
Scope & Level	System, Core.	
Preconditions	Admin user has created a profile	
Success End Condition	Admin user can add a recipe	
Failed End Condition	Admin user cannot add a recipe	
Actors	User, System	
Trigger	Admin user enters create recipe section	
DESCRIPTION	**Step**	**Action**
	1	Admin user opens the DCU Munch Application
	2	Admin user has successfully signed in
	3	Admin user enters create recipe section
	4	Admin user can enter recipe title, description, steps and videos.
	5	Admin user can save and publish recipe.
EXTENSIONS	**Step**	**Branching Action**
	3a	Admin user must be signed in at all times for the application to work

2.4 Constraints

Here is the list of constraints that could affect the development of the application.

Speed - We are unsure if the Django framework can handle all these functionalities quickly and efficiently as we would like because python can be slow at times.

User admin requirements - It will be difficult to provide a system that is 100% easy and fast for the user considering the time we have.

External technologies - The API and google sign integration system, we do not know yet if we will have access to use these technologies and how they will affect the system in terms of speed.

Stretched resources - this is a project with a large scope which will lengthen as time goes on and will take a lot of designing, there are only two persons in this project which might make it hard to finish the application according to the deadline.

3. Functional Requirements

3.1 User Profile

Description

We are going to make the system aware of all the DCU student's information. In order for there to be correct logins only for DCU students. We will store them inside a database. The user will have their own profile which they can upload their dietary requirements.

Criticality

This login in the element will be very critical to our system for identification purposes. Without this function users, will not be able to make orders from the application. Once users log in they can enter their dietary requirements and view the menus and use the website application.

Technical issues

- Making sure each login is stored in the database properly

Dependencies with other requirements

Users will not be granted any access to the application if they aren't signed in.

3.2 User dietary requirements

Description
Once the user is signed into their account they will be able to enter their dietary requirements such as age, height, weight and any other necessary information.
This will then filter out the meals that would best suit the user and easy-cook meal plans that the user can make at home

Criticality
Its important that we collect the user's dietary requirements in order to provide them with food that will best suit them. If the dietary requirements given are incorrect it will provide the user with the wrong items to purchase.

Technical issues

- The main issue is to make sure that all the dietary requirements put into our database link to the right meals

Dependencies with other requirements
For the above requirements to take place the user must be signed in, they also must be greeted with an interface that allows them to select their dietary requirements.

3.3 Payments

Description
When the user has chosen what they want to order they will be greeted with an interface to enter payment details. Once the user has entered their payment details they will be given a confirmation number after successful payment and a QR code to scan at the cafeteria to pick up the food when it is ready.

Criticality
It is critical that users pay for their food before placing an order. So that the restaurant can prepare the food on time. If the users could place an order without payment this would slow the production time as there could be some orders placed that people might not want anymore.

Technical issues

- Verifying that the credit card used is valid

Dependencies with other requirements
Users must be signed in to place an order for any item on the website. This is because our application needs to collect data from the user for recommendations and orders in the future.

3.4 User Feed

Description
Users will have a feed section in the application, which will display all their recommended meals, rating feed system will be generated after the user answers a series of questions which will give our system an idea of what the user likes to eat. It will also recommend how many calories they should eat for the day.

Criticality
It is critical that the users enter the right answers in order for our system to provide the best meals that will suit their needs.

Technical Issues
- To make sure that the calories recommended are correct

Dependencies with other requirements
Users must sign into their account and rate their meals, if they don't rate their meals the system will not know what the user likes to eat.

3.5 User Recipes

Description
Recipes will be recommended for the user, they can also watch videos and read how to prepare the recipe. Users can then rate the recipes, the recipe will also be added to the total calories that they have consumed that day.

Criticality
It is critical that we provide the right recipe for each user. As it will affect their diet if it is not correct.

Technical Issues
- The recommendation system will be hard to set up, we will use the same algorithm as the rating system.

Dependencies with other requirements
This interface will be directly interacting with our database. It will continuously find out the requirements and ratings that the users have placed.

3.6 Orders

Description
Users will be able to order from 4 different food places on campus. Users will be able to pay through our payment system. After the payment is successful the user will be given a QR code with an order confirmation, with a time and where they can pick up the food. Users' transactions will be kept on their transactions page

Criticality
It is critical that the cafeteria gets the food prepared for the time they allocated to the user

Technical issues

- Our objective is to make a QR code system that will work when scanned by the user. This may be challenging as we will need a lot of memory to store these codes.

Dependencies with other requirements
Users must sign in and enter their dietary requirements before being able to make an order.

3.7 Admin profile

Description
Admins will be able to log in in a separate section and add their contact information and add their menus. They can also add recipes for students to our database. Admins will be able to see all their transactions and how much they have made. Recipes and food that users like will also be shown.

Criticality
The admin profile is a critical feature to our application; without it the users won't be able to see the menus to order from our application.

Technical issues

- Setting up the API to allow DCU Catering to input the food into our system could be an issue due to us not having much experience using the API.

Dependencies with other requirements

For the above requirement to take place the admin must be provided with an interface that allows them to login and work their way through the steps to add items to their menu as well as uploading menus to the application

3.8 Admin creating menus

Description

Admins will be able to add new meals/menus to their section of the application. Once they are added we can translate the meals into users nutritional values e.g the amount of carbs, sugars etc. the menus can be changed at anytime

Criticality

It is critical that the catering team update the menus when needed in order for the users to be up to date.

Technical issues

- Providing the correct nutritional information might be difficult as it might not be completely accurate. (E.g adding extra bit of oil to original recipe)

Dependencies with other requirements

Admin will not be able to add menus/meals without being signed in.

3.9 Admin creating recipes

Description

Admins will be able to add healthy and easy recipes for students who are living away from home. Only recipes based on the students' food needs will be shown on their page, these recipes will be provided with video tutorials. Students can also rate the recipes so the cafeteria knows which recipe works for students.

Criticality

Its important that we provide an easy environment for the admins to upload their recipes. This will make the uploading process more efficient.

Technical issues

- The main issue here would be to provide only recipes based on the students' food requirements.

Dependencies with other requirements

Admin must upload the menus in order for the students to use them.

3.9.5 Admin transactions

Description

For this functional requirement the admins will be able to log into their account and view all the transactions they made on a daily, monthly or yearly basis. They can see how much money they make and what meals are popular compared to the unpopular ones.

Criticality

it is important that we provide accurate data for the DCU catering

Technical issues

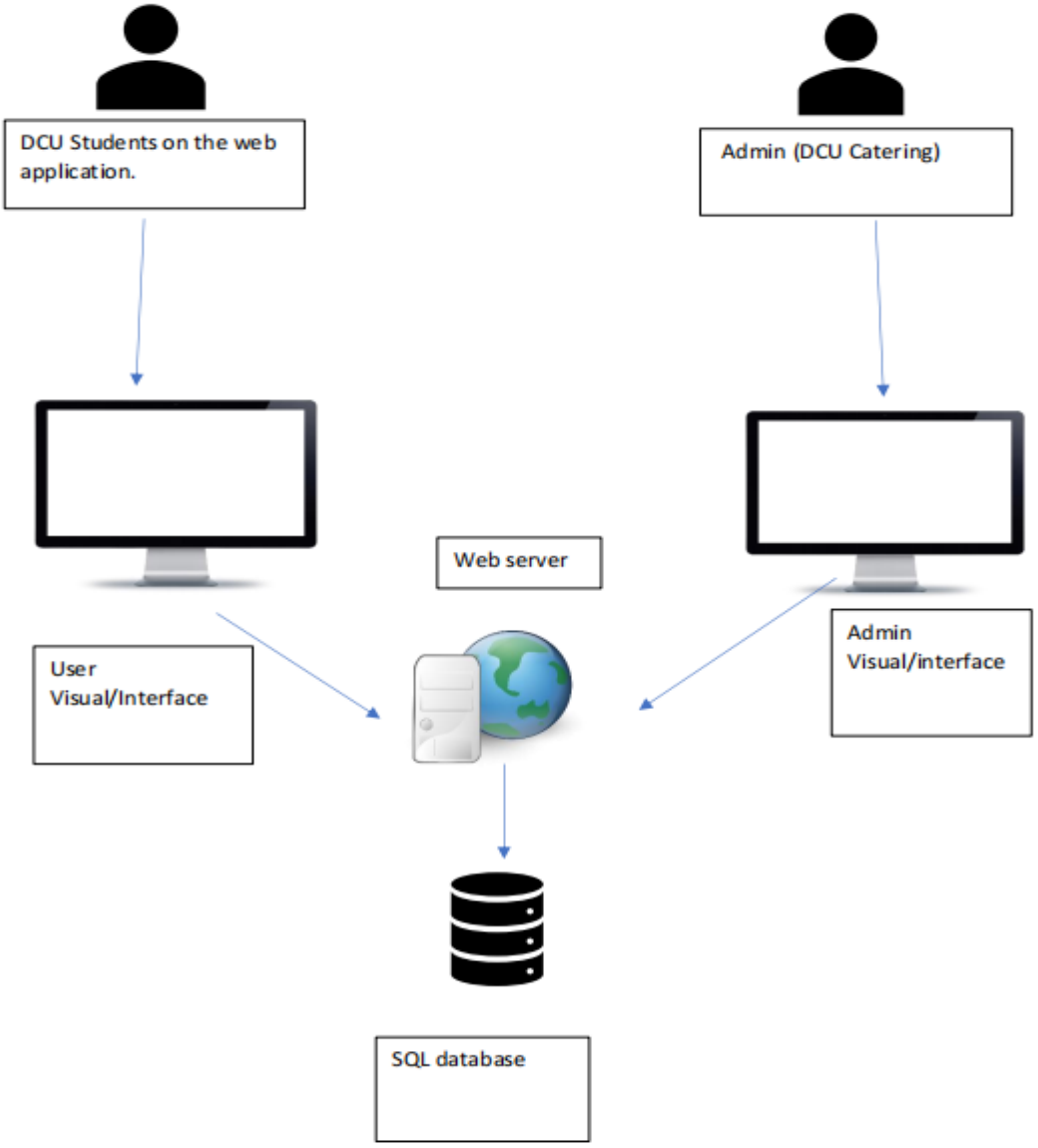
- Providing the amount that the catering company makes from each transaction and adding them all up may be challenging

Dependencies with other requirements

Students must be able to order from the application in order for there to be transactions on the admin's account.

4. System architecture diagram

In our system architecture diagram we show the structure of our application. DCU students and the admin will be connected to an interface where they will be connected to the web server which is linked to our database.



5. High-Level Design

In this section we are going to talk about our high-level design, the HLD provides a comprehensive overview of the software development process along with the system architecture, applications, database management, and complete flowchart of the system and navigation.

Context diagram

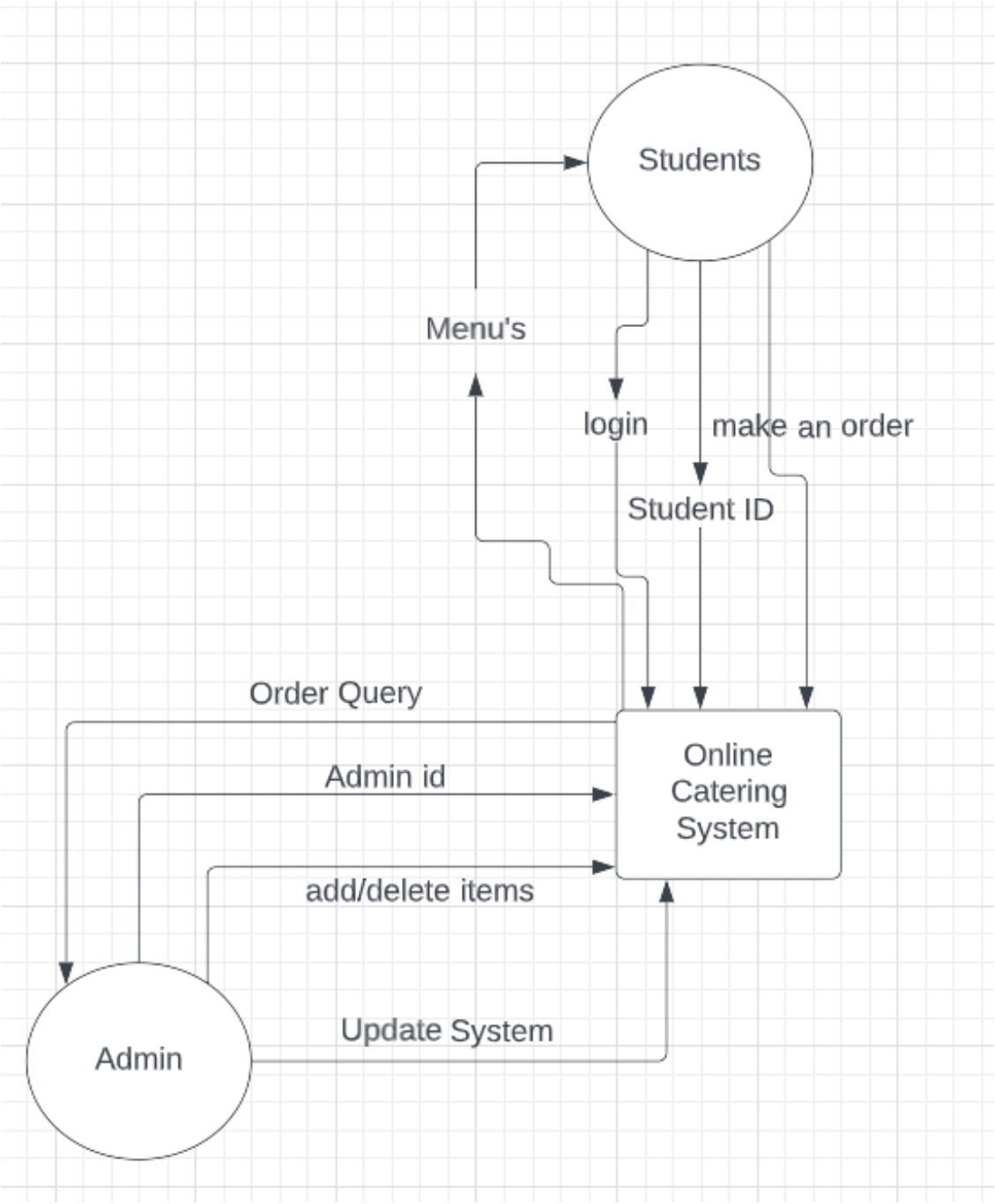
This diagram shows how the system interacts with its external entities to the system. It helps us wrap our heads around the scope of our system. How the users interact with the system.

Data flow diagram

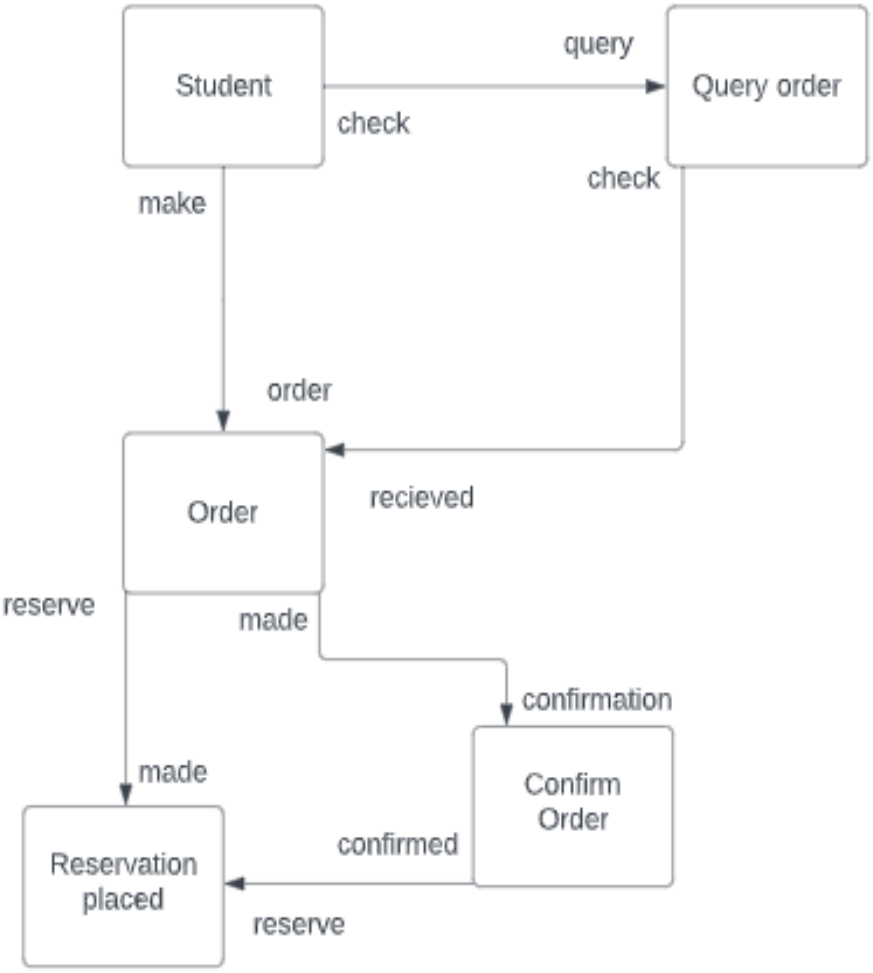
Here, we aim to present the system's data transit patterns and demonstrate its usability. We provide the system's internal workings, its external entities, the data flow between these workings and its data stores.

Logic Data System

In this diagram we will show how the users will interact with the interface to create an order.

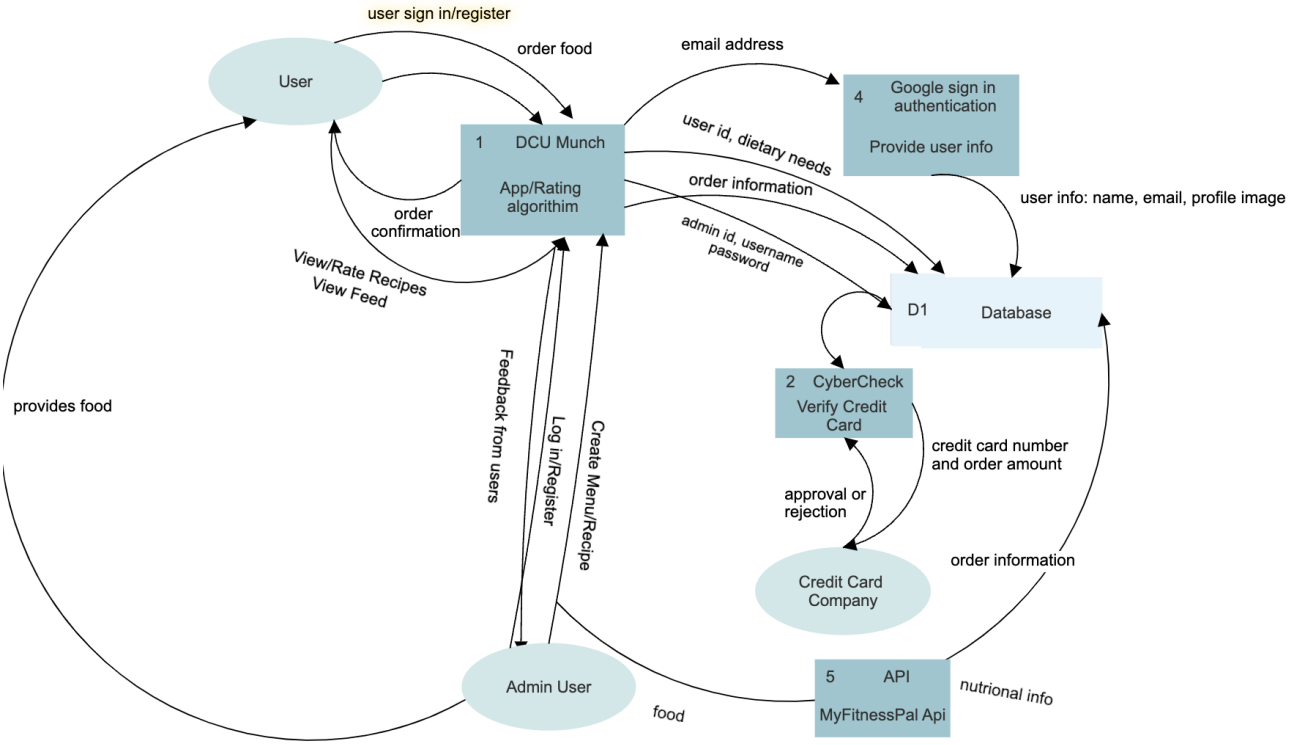


Context Diagram



Logic data system diagram

Data Flow Diagram

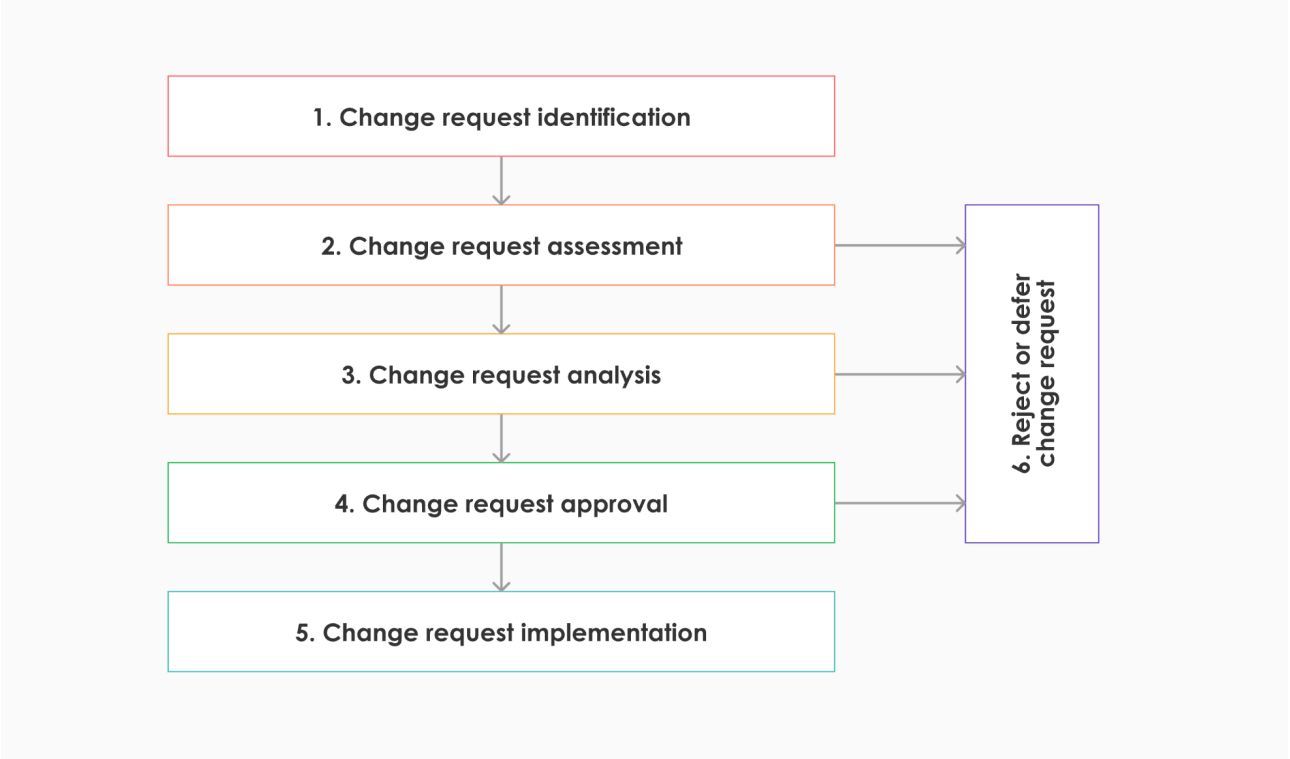


6. Preliminary Schedule

This is the first draft of the plan and schedule for the application. Below is a GANTT chart which shows the major tasks and timeline for the project. The main technology needed is Django (Python framework), Node (JavaScript), SQL for the database and a web server. [Fig 1]

7. Change Request Management

Below is a diagram of how design changes will be handled. Design changes will be generated usually by us as we debug and test our code. Each design change will have a request assessment and analysis where we will decide if the design change will be beneficial to the system or not. Following that, we can approve it and attempt to implement it into the application. If the design change does not work we will reject or defer the change.



8. Risks and Assumptions

8.1 Assumptions

- Accuracy - we do not know the full extent of how accurate the API is. Some information might be redundant or out of date. For example how quickly can the API update information on a new recipe for a food product?

- DCU Email - we are assuming that every student in DCU has a working college email. This might be risky if google has problems and their servers fail.

- Recipe changes - another assumption that might affect accuracy also is the fact chefs can change recipes slightly so calorie information might be slightly inaccurate.

8.2 Risks

- Large orders - a lot of the DCU food outlets cannot manage a large number of orders at once. So we need a way to make the process easier for them if there are a large number of orders.

- Scope creep - there are a lot of features in the application, as we test and debug there might be more features that need to be added therefore increasing the scope and the amount of work that needs to be completed.

- Large amount of users - there are thousands of potential users from alumni, staff and students. Even though the application is catered towards students, if anyone has a working DCU email address they will be able to use the application. We are not sure if the web server of DCU can manage all these users.

9. Operational Attributes

9.1 Speed

One of the main attributes DCU Munch will have is speed we want to make the application as fast as possible for both users and DCU Catering. Users need to have confidence the app is tracking and recommending the right foods and recipes in a fast, efficient and effective manner. It is quintessential that the order system is quick so it does not affect DCU's Catering good customer service.

9.2 Maintainability

The application should be able to be maintained for a long time even when new menus come out or new food places are established. The code will be of good structure and a good standard to avoid early errors at the beginning of the system.

9.3 Multi-Device usage

The application will be catered to mobiles, tablets and computers with modern browsers. To accomplish this responsiveness of the web design will be a key priority and handling data from the different device types will also be an issue that will be addressed.

10. Appendix

Fig 1