

# **SOFTWARE REQUIREMENTS SPECIFICATION**

**For Image Based Food Classifier (Calorie Click)  
Version 1.0**

**Prepared by:**

- 1. Aryan Garg (102103768)**
- 2. Samarth Paliwal (102103775)**

**Submitted to: Dr. Nitigya Sambyal**

## Contents

1.1	Purpose.....	3
1.2	Product Scope .....	3
1.3	Definitions, abbreviations, and acronyms .....	3
1.4	References .....	4
1.5	Overview .....	4
2	OVERALL DESCRIPTION .....	5
2.1	Product Perspective .....	5
2.2	Product Functions.....	5
2.3	User Characteristics .....	6
2.4	General Constraints, Assumptions and Dependencies .....	6
2.5	Apportioning of requirements.....	7
3	SPECIFIC FUNCTIONAL REQUIREMENTS .....	7
3.1	External Interface Requirements .....	7
3.2	Detailed Description of Functional Requirements .....	8
3.3	Performance Requirements .....	9
4	DOCUMENT APPROVERS .....	9

# 1 Introduction

## 1.1 Purpose

This SRS document aims to provide a detailed overview of our software product: Tuberculosis Detection web application. It grabs the purpose and various features of the app, its aim, and the constraints for the functioning of the app. This document is intended for end users and web app developers.

## 1.2 Product Scope

1. **Authenticate User:** It must be able to authenticate the user using a combination of username and password. This will be compared to securely stored data in the database.
2. **Maintain User Records:** As new users sign up for the web-app, it is important to retain their data to serve the next time they wish to login.
3. **Scan Image and predict class of food:** The program must be able to scan the entire image using a machine learning model specifically trained on food, and be able to find a food product in the image.
4. **Track User History:** The program must be able to track all the previous items scanned by the individual and show statistics for the previous data.

Initially, we aim to develop a web application with the primary function of analyzing food photos to estimate their calorie content. This application will be piloted in four Software Development and Testing (SDET) unit labs, targeting an audience of 10 individuals. If the pilot phase proves successful, we intend to expand the use of this calorie prediction web app on a greater scale.

## 1.3 Definitions, abbreviations, and acronyms

<i>S. No</i>	<i>Term</i>	<i>Description</i>
1	<b>Authenticate</b>	<b>Provides access control for systems by checking to see if a user's credentials match the credentials in a database</b>
2	<b>CNN</b>	<b>Convolutional Neural Network</b>
3	<b>API</b>	<b>Application Programming Interface</b>

## **1.4 References**

- [1] Google Images (For Dataset): <https://www.google.com/imghp?hl=en>
- [2] Kaggle: <https://www.kaggle.com/datasets/aryan401/indian-food-16>
- [3] TensorFlow Playground: <https://playground.tensorflow.org/>

## **1.5 Overview**

The remaining sections of this document provide a general description, including characteristics of the users of this project, the product's hardware, and the functional and data requirements of the product. General description of the project is discussed in section 2 of this document. Section 2 gives the functional requirements, data requirements and constraints and assumptions made while designing the multi-utility system. It also gives the user viewpoint of product use. Section 3 gives the specific requirements of the product. Section 3.0 also discusses the external interface requirements and gives detailed description of functional requirements

## 2 Overall Description

### 2.1 Product Perspective

The “Calorie Click” application is a user-friendly web application that helps to detect various meals that the user consumes, whenever they upload an image of it. The web-application provides the following categories of services:

- **Scan Image and Predict Class of Food:** The web application allows users to upload a picture of their Meal. The model processes these uploaded images and determines which of the pre-trained meals the user is having
- **Track users’ progress:** The web app allows users to track their progress and Nutritional History.
- **Maintain user profile records:** The web app maintains a database storing a patient’s profile containing their meal history as well as the calorie, and macro/micro-nutrient intake per meal.

### 2.2 Product Functions

The major features of Calorie Click are:

- : It allows user to register, complete their profile and later, log in. During log in, the user and password are authenticated.
- It scans images and gives result on the class of item being consumed.
- It tracks the progress of the user.
- Gives information about different types of nutrients that the body requires.
- It maintains a database storing a user’s profile containing their nutritional history.

## **2.3 User Characteristics**

The goal of Calorie Click is to enable healthy eating habits within its users. These include:

- Admins/Developers
- General Users

The admins have the following privileges:

- Add/Delete a record
- Update a record
- Terminate Spammy User Accounts

General Users will have the following privileges:

- Add images to their personal profile

## **2.4 General Constraints, Assumptions and Dependencies**

The following list presents the constraints, assumptions, dependencies, or guidelines that are imposed upon implementation of the CSC based Multi-Utility System including Access Control and Attendance Monitoring software:

- The website operates in any Operating Environment - Windows98 and upper Versions (EX: Win98, windows 2000 prof, XP Vista and win NT Server, Windows 2000 server, 2003 server, and Windows 2008 server), Unix and all Unix flavors like LINUX, Solaris, etc.
- Using browsers such as - Google Chrome, Microsoft Edge, Brave Browser, Mozilla Fire-fox, AOL, and Netscape Navigator etc.
- The user has a system which can support the latest version of React.js used.
- The server can support the latest version of Node.js and is powerful enough to run a large CNN.
- The user has an Internet connection with a minimum speed of 0.5 Mbps and recommended speed of 1.5 Mbps.
- The central database server and backup database servers should be updated regularly. This updating and replication of data from central database server to the backup database server can introduce additional latency in the working of the system

## 2.5 Apportioning of requirements

The Calorie Click is to be implemented in three phases:

- i. **Pilot Phase:** This application will be piloted in four SDET unit labs, we will initially be beta testing with a target audience of 10 individuals. If the pilot phase proves to be successful, we intend to expand the use of this calorie prediction web app on a greater scale.
- ii. **Extended Beta Phase:** Following successful nature of the initial pilot phase, we will further expand the roll out of the webapp to new testers and users. We may also add new food types and retrain our model.
- iii. **Full Rollout:** In the future we will add more food items, and eventually be able to differentiate many different food items. We will also be able to show more complex graphs based on user feedback.

## 3 Specific Functional Requirements

### 3.1 External Interface Requirements

The following list presents the external interface requirements:

- **User Interface:** The app should have a user-friendly interface to capture food photos. It should also display the predicted calorie count and any additional relevant information to users
- **Camera Integration:** The app should have access to the device's camera to capture food images. Integration with the camera API on both mobile and desktop z platforms is essential
- **Calorie Database:** Access to an external calorie database or API to retrieve nutritional information for various food items. This database should be regularly updated and accurate.

The functional requirements for the project are explained below in the form of tables with their respective purpose, inputs, processing, and output.

## **3.2 Detailed Description of Functional Requirements**

### **3.2.1 Functional Requirement for Sign-Up Screen**

Purpose	Users can enter their information and create a UserID and password, which is stored in the database for future logins
Inputs	Name, Date of Birth, User ID, Password, Confirm Password
Processing	All the entered information is stored in a database, and passwords will be hashed for security
Output	A new Account is created for the User

Figure 3.1: Functional Requirement for Sign-Up Screen

### **3.2.2 Functional Requirements for Login Screen**

Purpose	To authenticate users and validate their credentials
Inputs	UserID and password
Processing	The ID and password are sent to the database and compared with the stored value. When validated users are redirected to the home screen. On failure, the process is repeated.
Output	The users are logged into the account

Figure 3.2: Functional Requirements for Login Screen

### **3.2.3 Functional Requirements for Food Upload Page**

Purpose	To Upload the Image of the food item
Inputs	Image of the Food on the Device, can also take a photo on the spot with camera
Processing	The food image is fed into an ML model. The model evaluates the image based on trained data and determines which class the food items belongs to.
Output	The page determines which food class the image belongs to

Figure 3.3: Functional Requirements for Food Upload page



### 3.2.4 Functional Requirements for Statistics Page

Purpose	To check food logs
Inputs	None
Processing	Process previous data and generate meaningful data to the end user
Output	Tables and graphs

Figure 3.5: Functional Requirements for Statistics Page

### 3.3 Performance Requirements

The requirements associated with performance are:

- The application requires an Internet connection with a minimum speed of 0.5 Mbps and recommended speed of 1.5 Mbps
- The application supports up to 100 concurrent users.
- The application takes 0.5-3s to load, depending on the users' connection speeds.
- The application must be hosted on a reliable server like AWS servers.

## 4 Document Approvers

SRS for Calorie Click: Image Based food recognition and nutrient information web-app  
approved by: \_\_\_\_\_

**Designation:**

**Date:**

# Feasibility Report: Calorie Click

## Executive Summary

This feasibility report assesses the viability of developing a web application that enables users to log their meals by taking images of their food and provides nutritional information such as calorie count and other nutrients. The proposed web app aims to promote healthier eating habits by making meal tracking more convenient and informative. This report evaluates various aspects of the project, including technical, financial, and market feasibility.

## 1. Technical Feasibility

### 1.1. Technology Stack

Developing the web app requires web development, image recognition, and nutrition database integration expertise. The technical team must have experience with technologies such as:

- Frontend: HTML, CSS, JavaScript, and a JavaScript framework.
- Backend: Node.js for server-side logic.
- Image Recognition: Integration with machine learning libraries or APIs (Pytorch, HuggingFace).
- Nutrition Database: Access a reliable nutrition database or API (e.g., USDA Food Database).

### 1.2. Image Recognition

Developing a robust image recognition system capable of accurately identifying various types of food is a significant technical challenge. Consideration should be given to leveraging pre-trained models and continuously improving the model's accuracy over time.

### 1.3. User Authentication

Implementing secure user authentication to protect user data is crucial. This requires encryption, secure password storage, and user account management features.

### 1.4. Database Management

A database is needed to store user accounts, food images, and nutritional information. Choosing the appropriate database system and architecture is essential for scalability and data security.

### 1.5. Mobile Compatibility

To increase user accessibility, the web app should be mobile-responsive and ideally have a dedicated view for iOS and Android smartphones.

## 2. Financial Feasibility

### 2.1. Development Costs

Estimate the costs associated with software development, including personnel, software licenses, hosting, and infrastructure.

In this specific case we face little development cost, most of which will be hosting the main website as well as any ML model APIs will also need GPU processing power.

### 2.2. Operating Costs

Calculate ongoing expenses for server maintenance, database management, and potential third-party API costs.

Given the small nature of our initial beta testing, we will require little operating budget. Training of the ML model may be an expensive process.

### 2.3. Revenue Streams

Need to investigate potential revenue sources, such as subscription fees, in-app advertising, or partnerships with nutrition companies.

### 2.4. Monetization Strategy

Develop a clear monetization strategy that outlines how the project will generate revenue and when it is expected to become profitable. This will require reiterating our revenue streams until we reach a point where the revenue from users crosses our development and operating costs.

## 3. Market Feasibility

### 3.1. Market Research

Although there are apps that can enable users to scan an image for food, none of them are specific made for Indian foods. These apps are also expensive for the services they provide. We plan to undercut them by creating a lower cost software for the end user.

### 3.2. Target Audience

Our target audience is mostly teenagers and adults in their mid-20s and 30s who are key on their health and calorie count. As we improve the accuracy of the model that we use, more customers of this demography will be attracted to our product.

## 4. Legal and Ethical Considerations

### 4.1. Privacy and Data Security

Comply with data protection regulations and ensure user data privacy and security. Develop a clear privacy policy for the same.

### 4.2. Intellectual Property

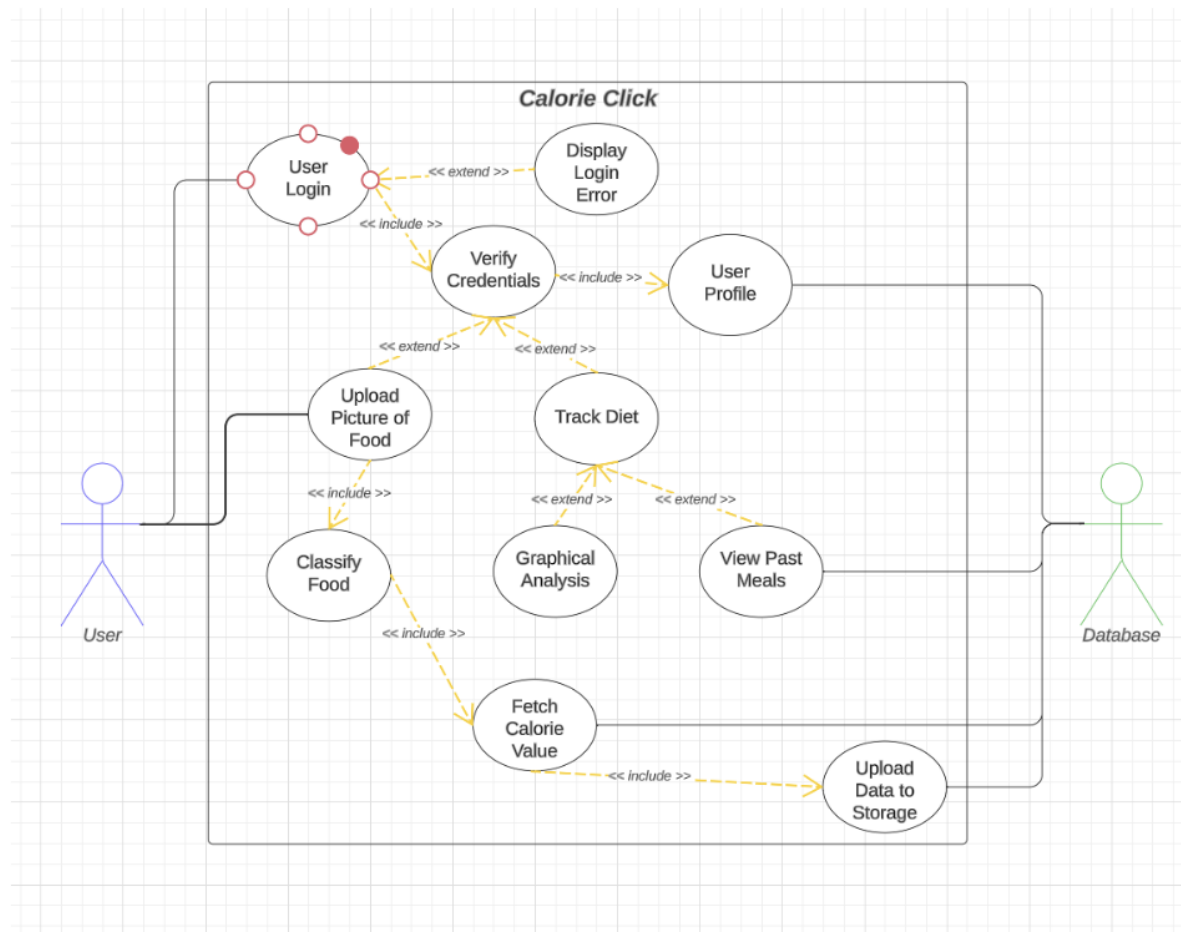
Ensure that the image recognition technology does not infringe on patents or intellectual property rights.

## 5. Conclusion

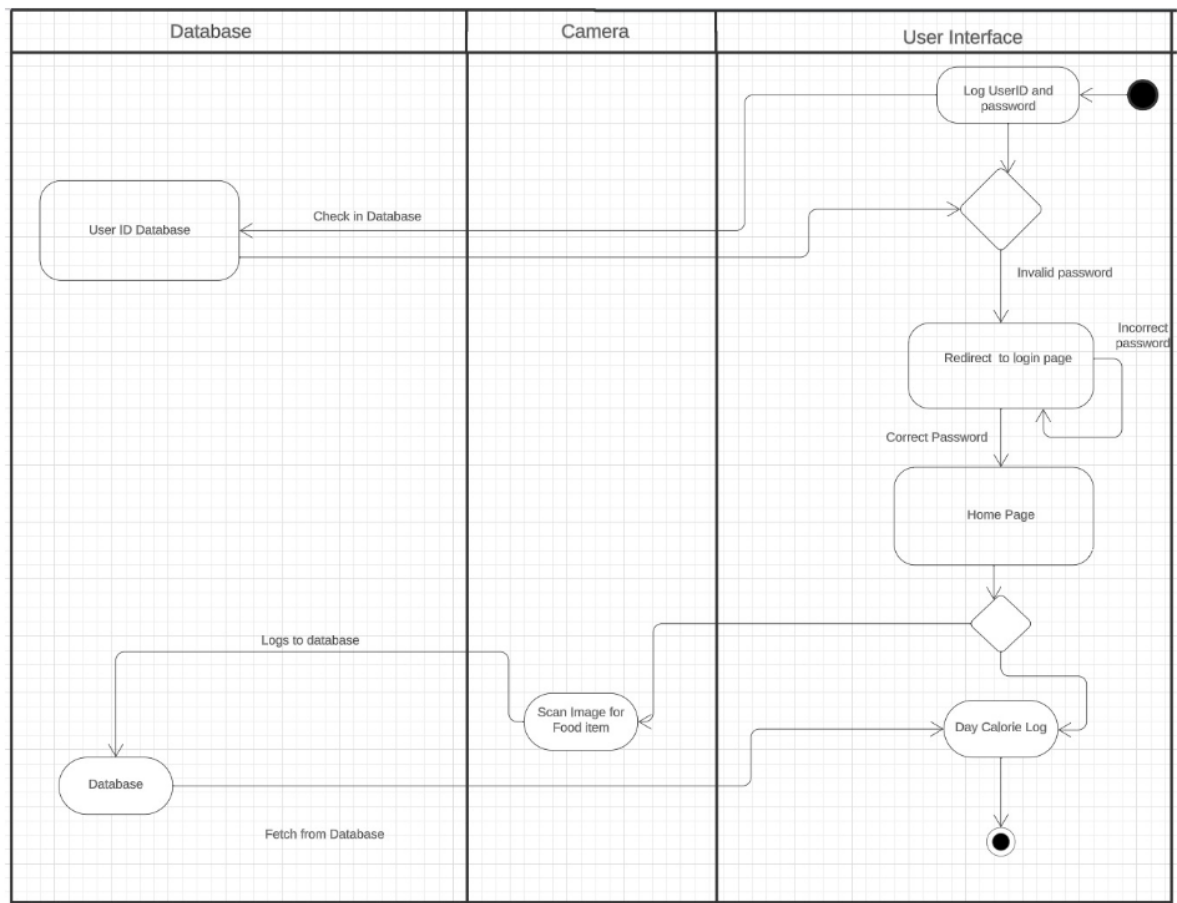
Developing a food image logging web application is technically feasible but requires careful consideration of image recognition accuracy, user authentication, and database management. Financial feasibility will depend on development and operating costs balanced against revenue streams. Market feasibility is promising, given the growing interest in health and fitness. Legal and ethical considerations are critical to protect user data and avoid legal issues.

Before proceeding, conducting a comprehensive feasibility study, including prototype development and user testing, is advisable to validate the concept and assess potential challenges in real-world scenarios. If successful, this web app could serve as a valuable tool for individuals seeking to make informed dietary choices and improve their overall health.

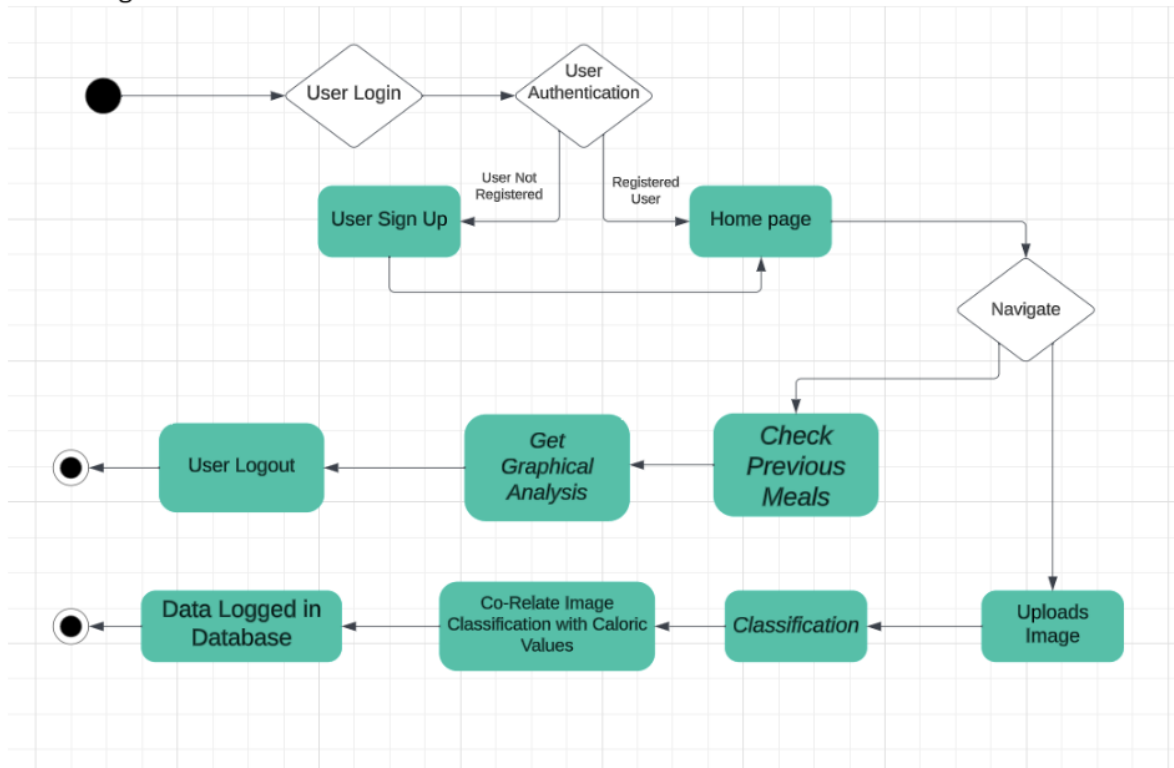
## Use Case Diagram



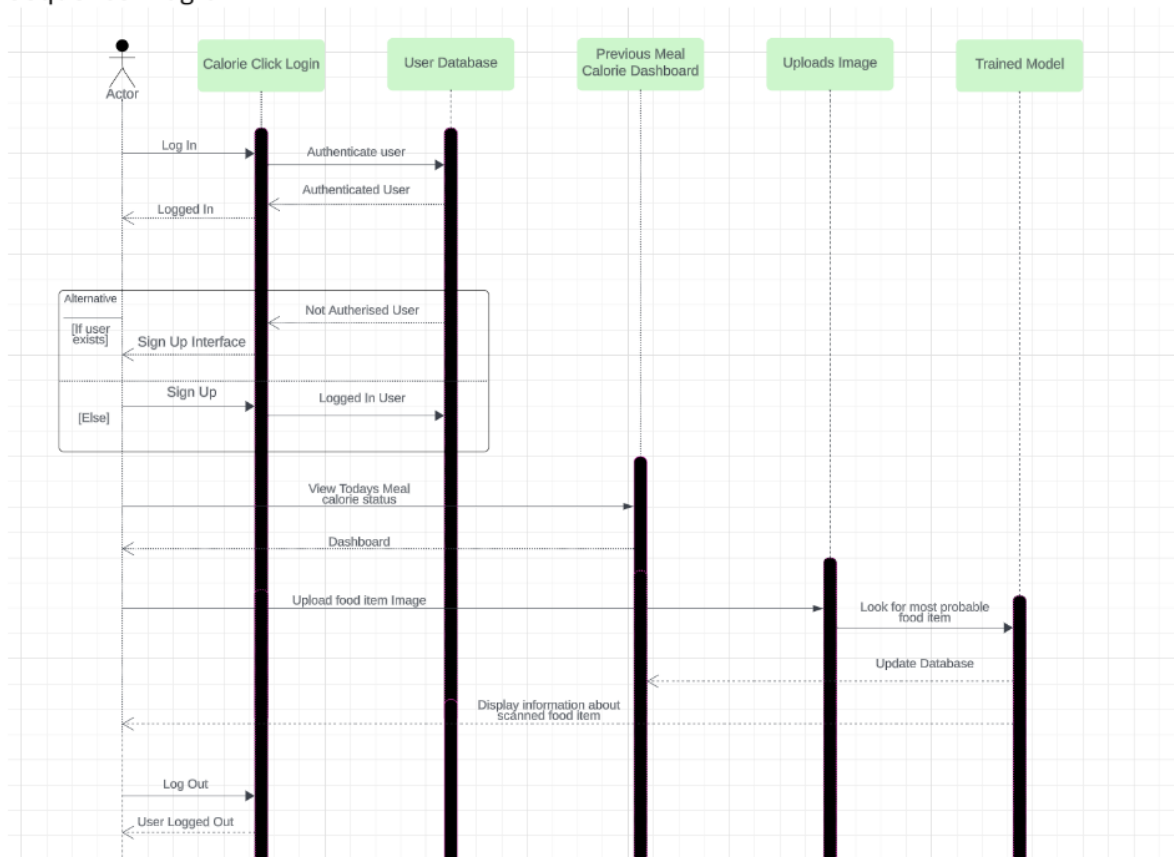
## Swimlane Diagram



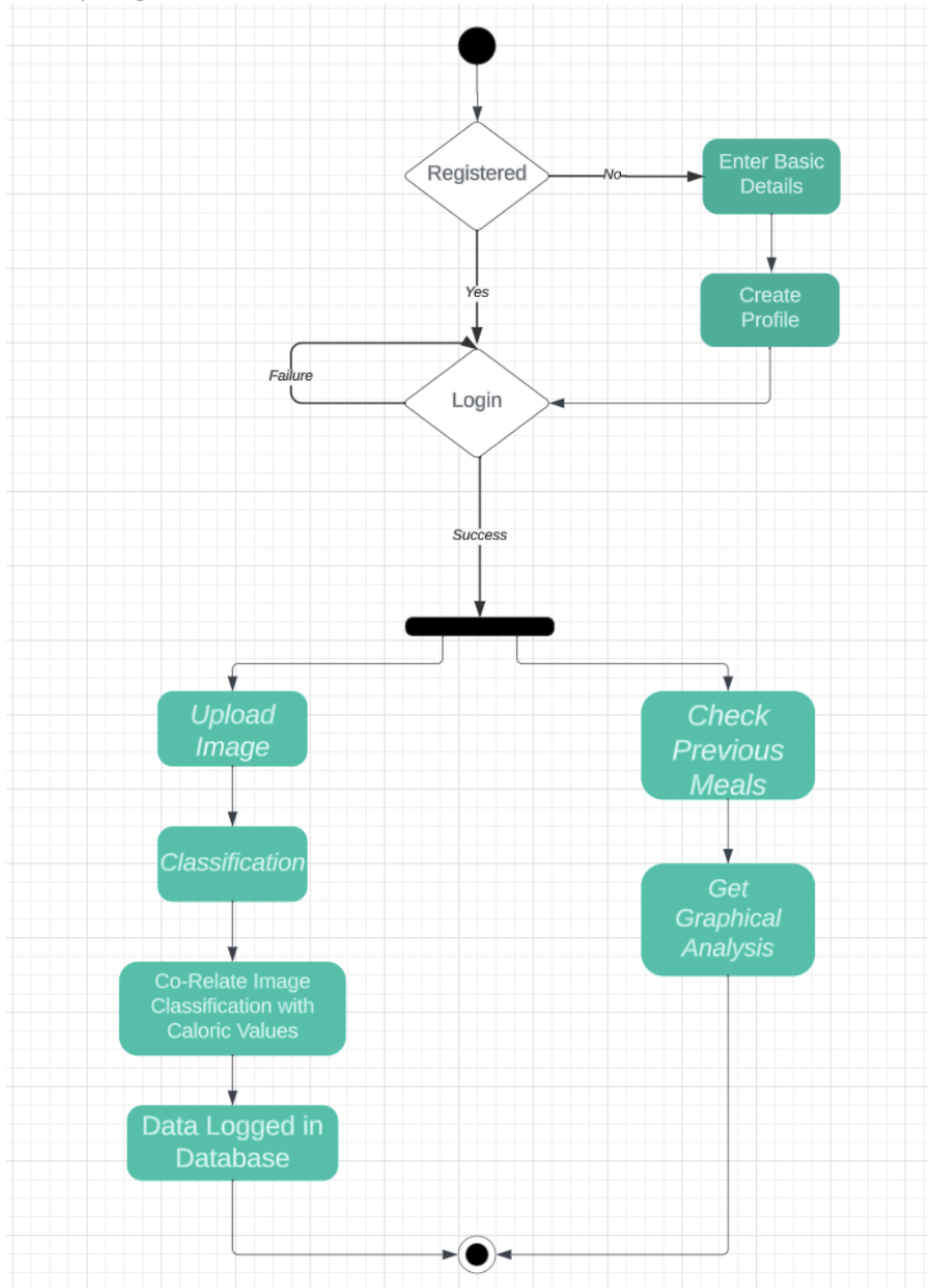
## State diagram



## Sequence Diagram

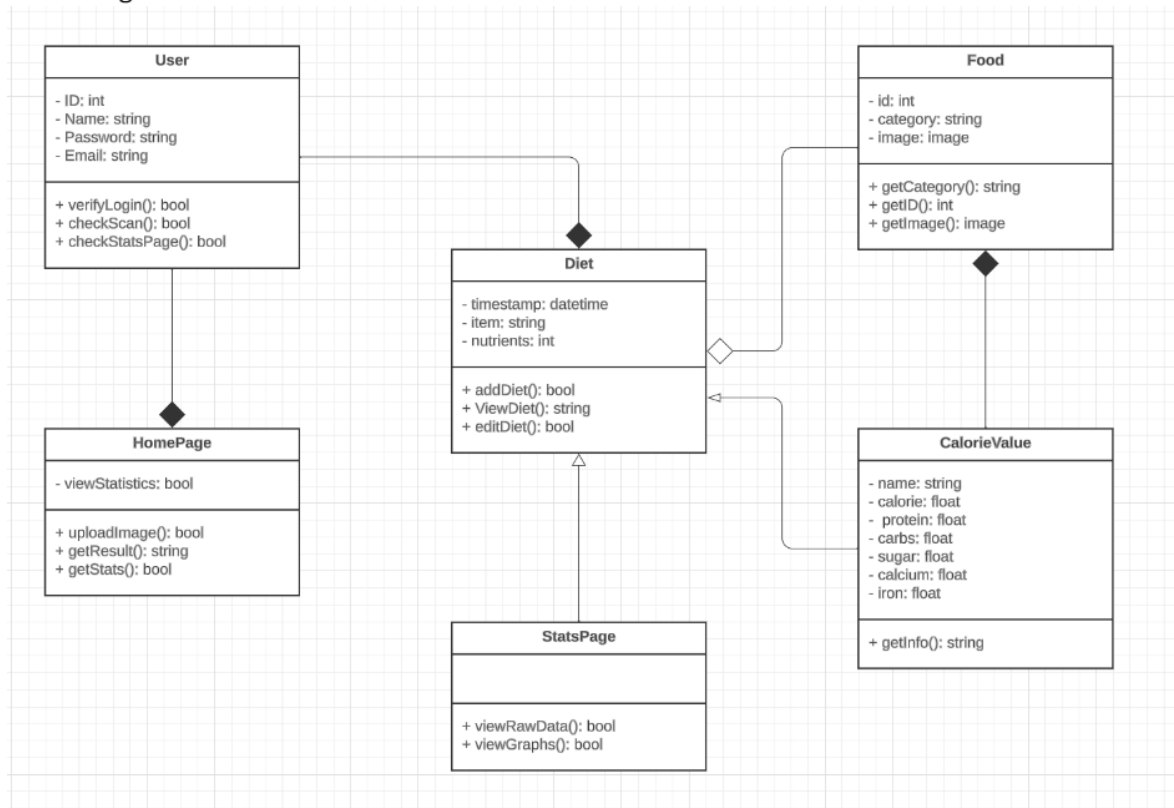


## Activity Diagram

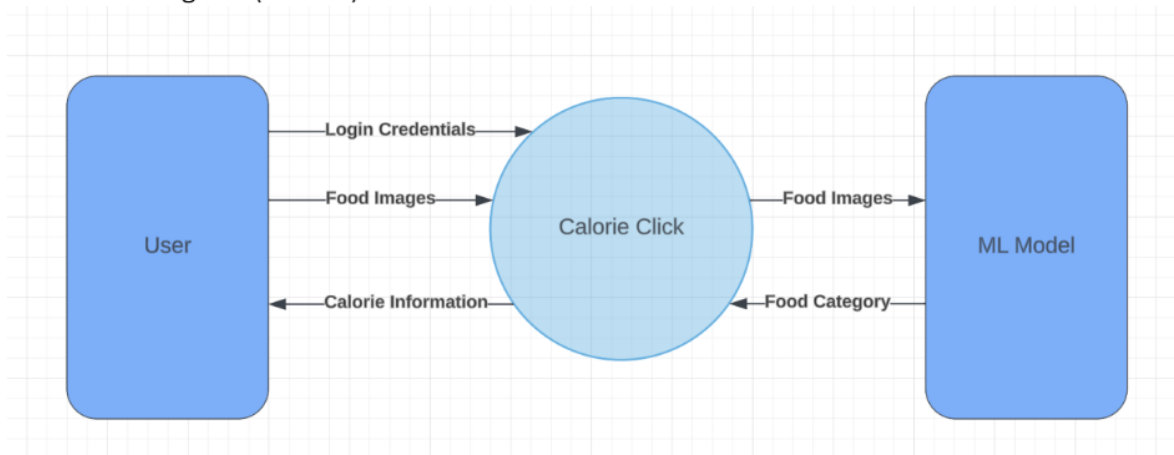




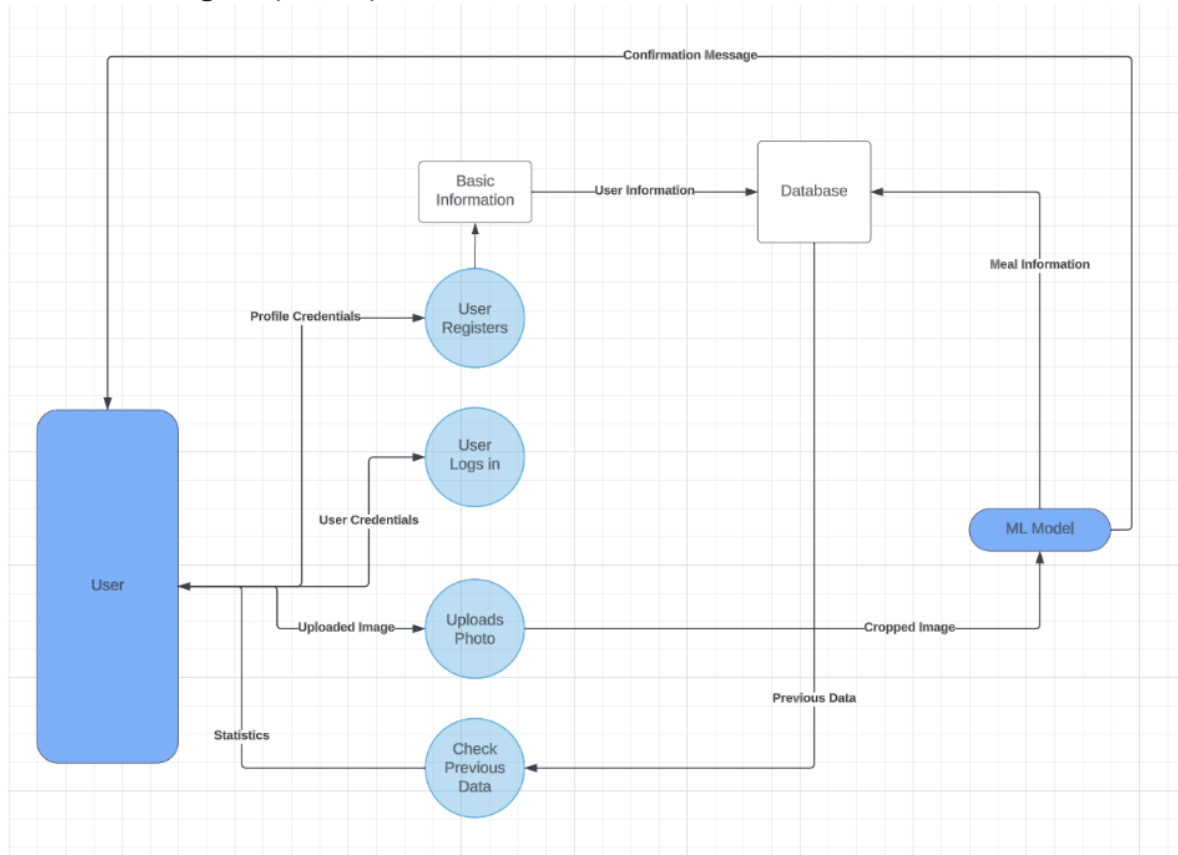
## Class Diagram



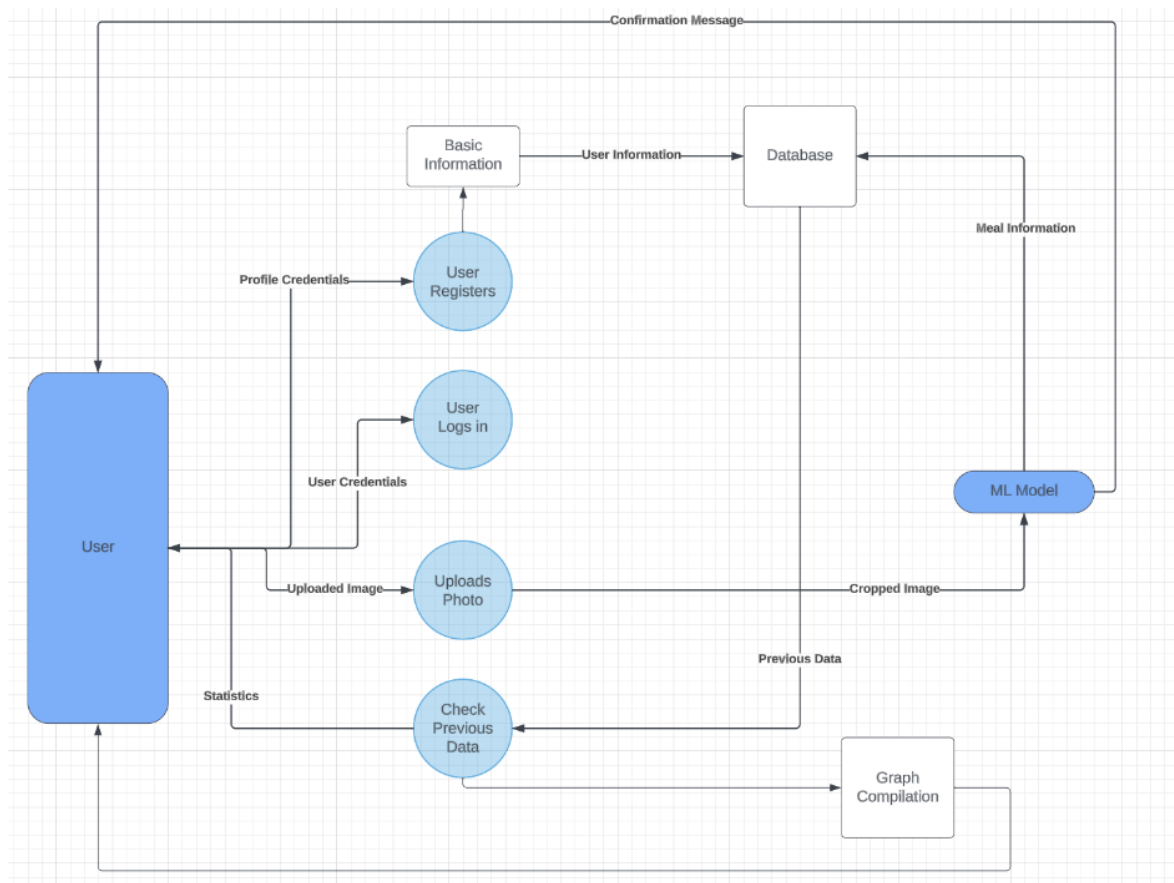
## Data Flow Diagram (Level 0)



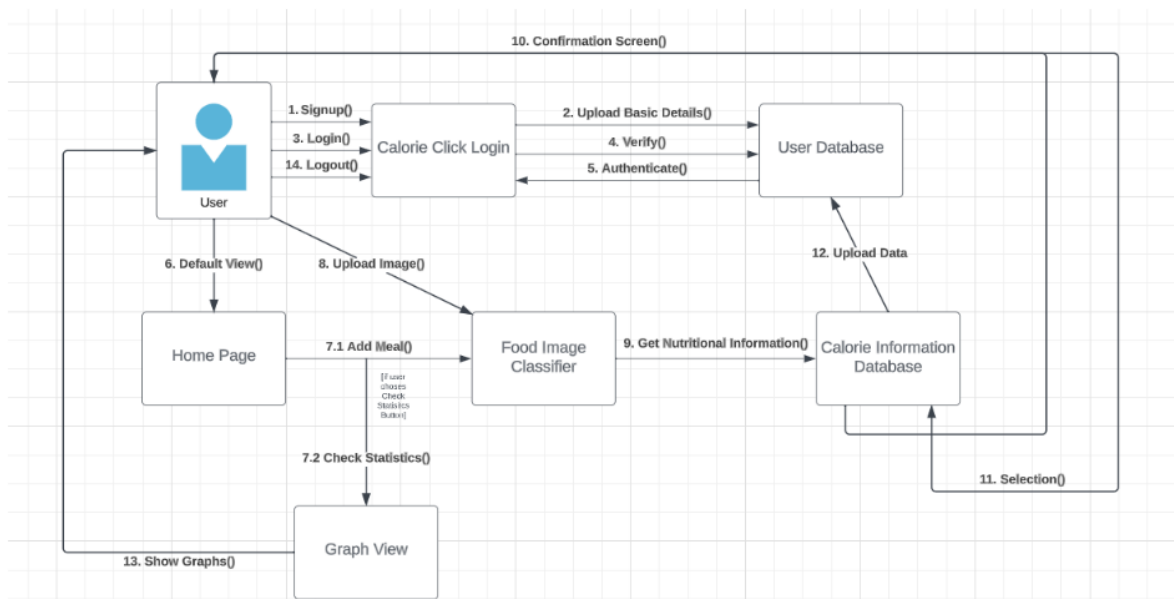
## Data Flow Datagram (Level 1)



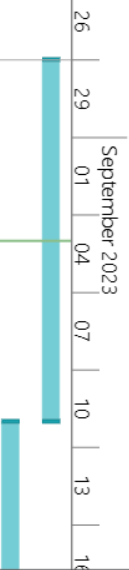





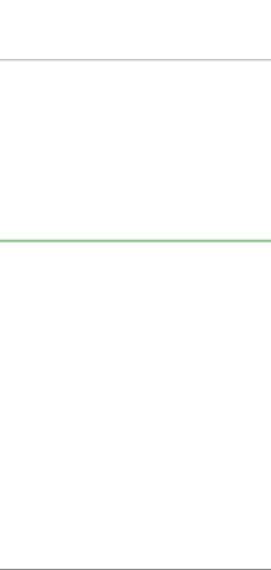


## Data Flow Diagram (Level 2)



## Collaboration Diagram



ID		Task Mode	Task Name	Duration	Start	Finish	Predecessors	Resource Names							
1			Dataset Creation	10 days	Tue 29-08-23	Mon 11-09-2									
2			Research	6 days	Tue 12-09-23	Tue 19-09-23									
3			Wireframing	7 days	Wed 20-09-2	Thu 28-09-23									
4			Model Design	20 days	Fri 29-09-23	Thu 26-10-23									
5			Front-end Development	20 days	Fri 29-09-23	Thu 26-10-23									
6			Model Connection	7 days	Fri 27-10-23	Mon 06-11-2									
7			Testing	7 days	Tue 07-11-23	Wed 15-11-2									
8			Documentation	9 days	Thu 16-11-23	Tue 28-11-23									

Task

Split

Milestone

Project Summary

Inactive Task

Inactive Milestone

Inactive Summary

Manual Task

Duration-only

Manual Summary Rollup

Manual Summary

Start-only

Finish-only

External Tasks

External Milestone

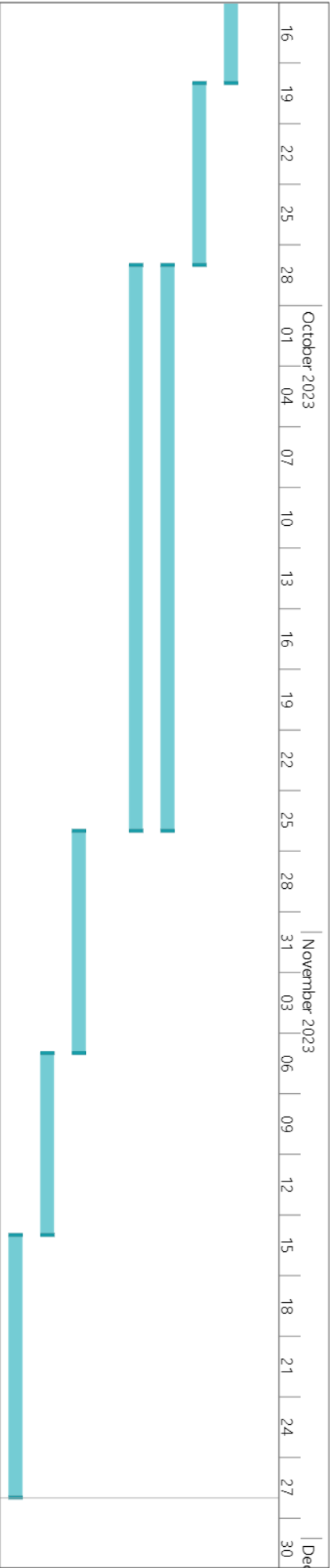
Deadline





Progress

Manual Progress

Project: GANTT\_7\_27

Date: Tue 05-09-23



Project: GANTT_7_27 Date: Tue 05-09-23	Task		Inactive Summary		External Tasks	
	Split		Manual Task		External Milestone	
	Milestone Summary		Duration-only		Deadline	
	Project Summary		Manual Summary Rollup		Progress	
	Inactive Task		Manual Summary		Manual Progress	
	Inactive Milestone		Start-only			

Project: GANTT_7_27 Date: Tue 05-09-23	Task		Inactive Summary		External Tasks	
	Split		Manual Task		External Milestone	
	Milestone Summary		Duration-only		Deadline	
	Project Summary		Manual Summary Rollup		Progress	
	Inactive Task		Manual Summary		Manual Progress	
	Inactive Milestone		Start-only			
			Finish-only			

## Use Case Scenarios

1. Use Case Title	User Login
2. Abbreviated Title	User Login
3. Use Case ID	1
4. Actors	User
5. Description:	Login by filling in user email id and password to access the software.
5.1. Pre-Conditions:	User must be signed up and verified
5.2. Task Sequence:	<ul style="list-style-type: none"> <li>• Enter verified email id.</li> <li>• Enter set user password.</li> <li>• Verify by clicking on “Log In”.</li> </ul>
5.3. Post Conditions:	User is logged in and can now access the software.
6. Modification History:	Date 17-September-2023
7. Author:	Aryan, Samarth

1. Use Case Title	Upload Food Image
2. Abbreviated Title	Upload Picture Of Food
3. Use Case ID	2
4. Actors	User
5. Description:	User can upload any food item image and he/she will get the various nutrients information like protein, calories etc. present in food item.
5.1. Pre-Conditions:	User must be logged in to the software.
5.2. Task Sequence:	<ul style="list-style-type: none"> <li>• Click on ‘Upload Image’ to upload an image.</li> </ul>
5.3. Post Conditions:	User will get a confirmation message about the food item image he/she uploads.
6. Modification History:	Date 17-September-2023
7. Author:	Aryan, Samarth

1. Use Case Title	Full Day Diet Nutrient log
2. Abbreviated Title	Track Diet
3. Use Case ID	3
4. Actors	User
<b>5. Description:</b>  The user can review their entire day's dietary intake, including an analysis of various nutrients consumed, to determine whether they have met their daily nutritional requirements.	
<b>5.1. Pre-Conditions:</b>  <ul style="list-style-type: none"> <li>User must have an active account and must be logged in.</li> </ul>	
<b>5.2. Task Sequence:</b>  <ul style="list-style-type: none"> <li>User need to click on “Nutrient Log”.</li> </ul>	
<b>5.3. Post Conditions:</b>  The user will receive nutritional information for their different meals.	
<b>6. Modification History:</b> Date 17-September-2023	
<b>7. Author:</b> Aryan, Samarth	