



YEDİTEPE UNIVERSITY



# CSE 344

## Software Engineering

# TERM PROJECT DESIGN REPORT

Course Instructor : Mert ÖZKAYA

Group 12

Ezgi Kara

Nilhan Süer

Berat Ilgaz Dursun

Yalın Gürsoy

Ceren Yazgan

## Table of Content

<b>1. Introduction.....</b>	<b>3</b>
1. 1.1.Purpose of the document.....	3
2. 1.2.Purpose of the system.....	3
3. 1.3.Structure of the document.....	3
<b>2. Detailed design class diagram.....</b>	<b>4</b>
1. 2.1.Class Diagram.....	4
1. 2.1.1.Detailed Classes.....	4
2. 2.1.2.Detailed Relationships.....	4
<b>3. Dynamic Models.....</b>	<b>5</b>
1. 3.1.Sequence Diagram.....	5
2. 3.2.State Diagrams.....	8
3. 3.3.Activity Diagrams.....	9
<b>4. Software Architecture.....</b>	<b>10</b>
1. 4.1.UML Package Diagram.....	10
2. 4.2.UML Component Diagram.....	11
<b>5. Entity Relationship Diagram.....</b>	<b>11</b>
<b>6.Glossary and References.....</b>	<b>12</b>
1. 6.1 Glossary.....	12
2. 6.2 References.....	12

# 1.INTRODUCTION

## 1.1 Purpose of the Document

This document's goal is to provide a concise explanation of class diagrams, including their characteristics and methods. The affiliations, compositions, aggregations, relationship names, and multiplicities of the relationships between the classes will all be discussed. State, sequence, and activity diagrams are examples of dynamic models that will be discussed in this document. And the UML Package Diagram and UML Component Diagram will be used to handle software architecture. The purpose of this paper is to explain every component of the software product, as well as the background mechanisms underlying the software class operations and the program's architecture.

## 1.2 Purpose of the System

Our purpose of developing this system is to help people who are discouraged to follow a dietary plan by the lack of expert support. Our program does not only create a meal plan that is suited for the user's health condition as any other program in the market does, it also ensures the user for that the program has taken the right actions by enabling the user's expert to keep track of what is happening with their personalized dietary plans.

Even though our target audience is people with special dietary needs, our program is suitable for anyone that is looking for a way to be healthier/fitter.

All those being said, experts are not only a supporter in our system design. They also can benefit from using our program with their own customers to be more organized and for the ease of accessibility. We believe that being able to access any patient's informations(such as medical history, weight, height, age etc.) and meal plan of the day with one click is going to be incredibly efficient.

## 1.3.Structure of the Document

Information regarding classes and their characteristics, processes, and relationships in terms of associations, compositions, aggregations, relationship names, and multiplicities will be provided in another section of the paper.

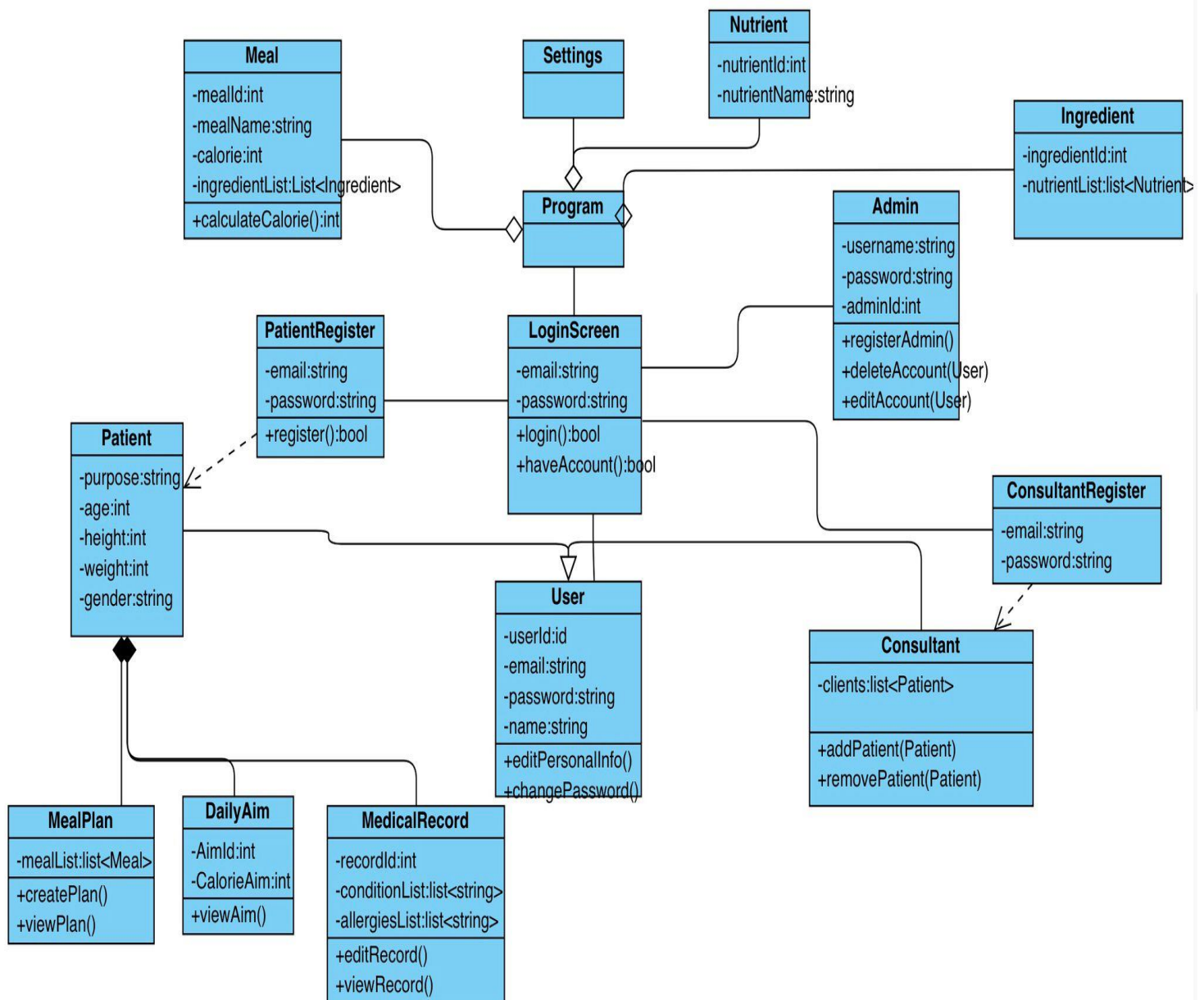
Following that, the document's "Dynamic Models" section, which includes "Sequence Diagrams," "State Diagrams," and "Activity Diagrams," is present.

Under the heading "Software Architecture," after the section on "Dynamic Models," there will be "UML Package Diagram" and "UML Component Diagram," then "Entity Relationship Diagram" and "Glossary & References."

## 2. DETAILED CLASS DIAGRAM

### 2.1.Class Diagram

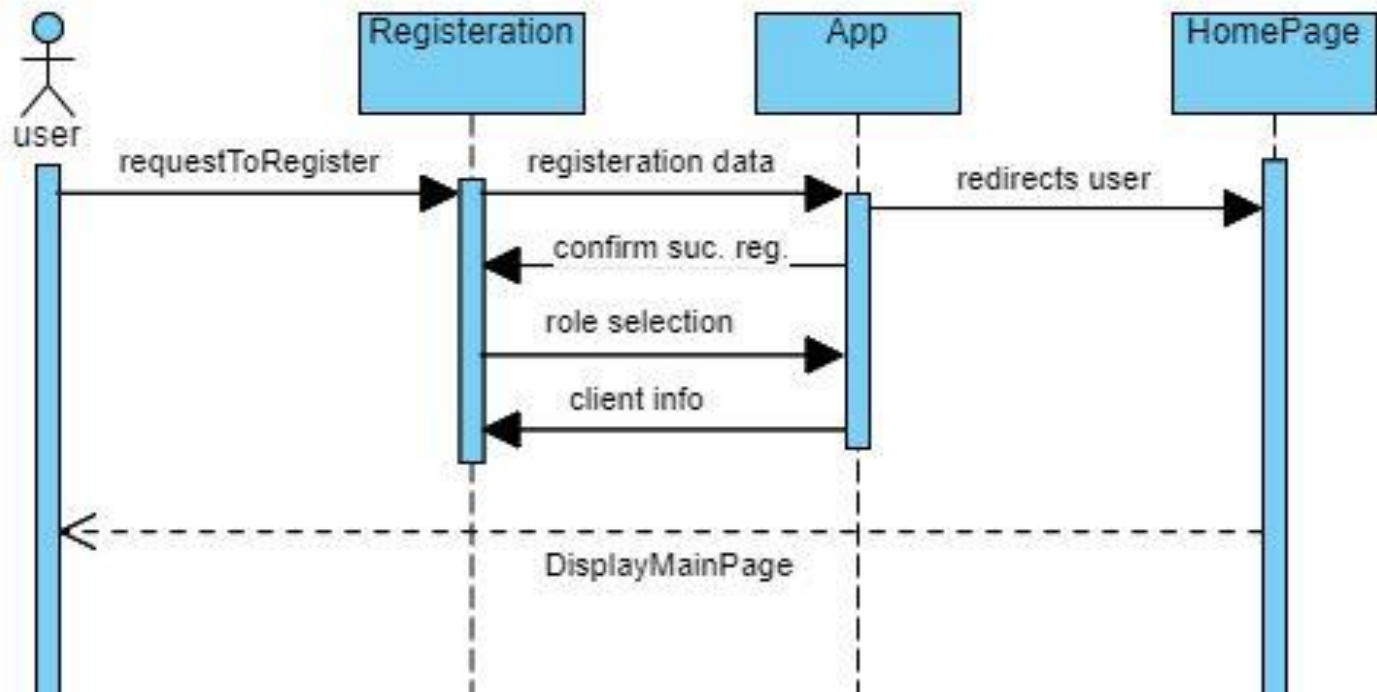
#### 2.1.1 & 2.1.2 Detailed Classes and Detailed Relationships



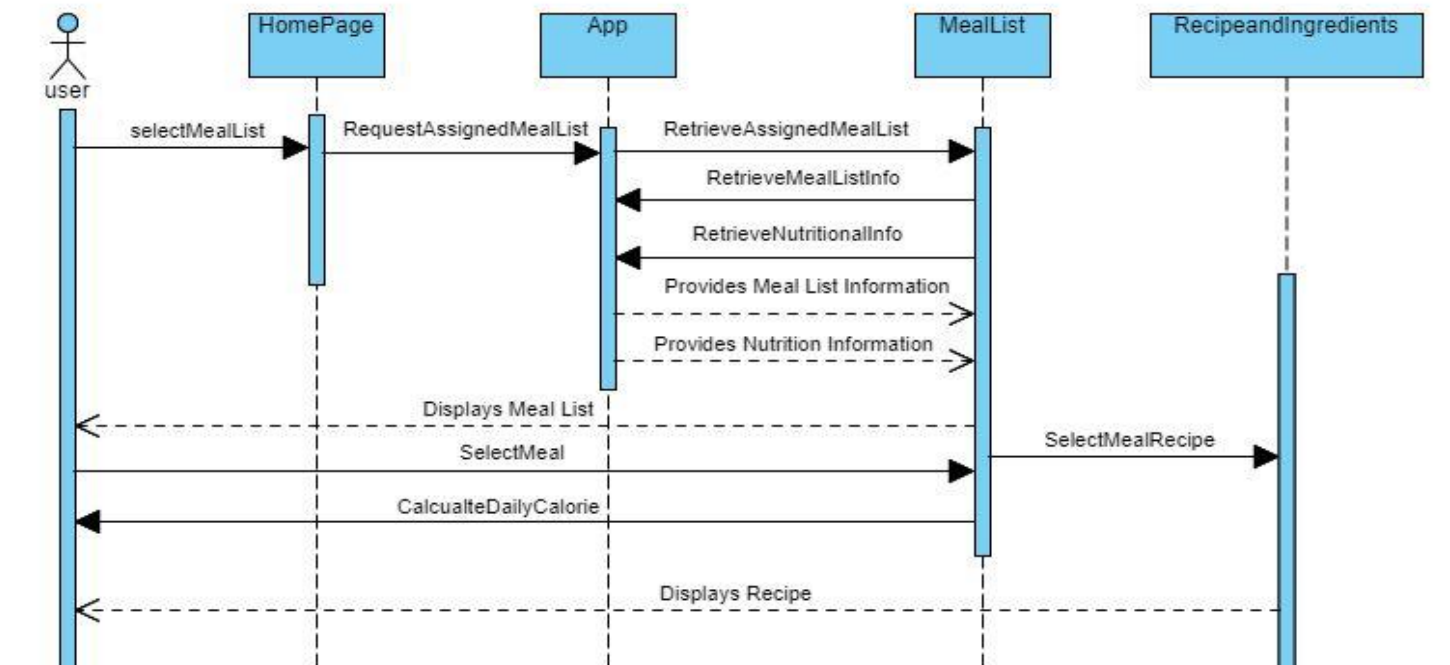
### 3.DYNAMIC MODELS

#### 3.1) Sequence Diagrams

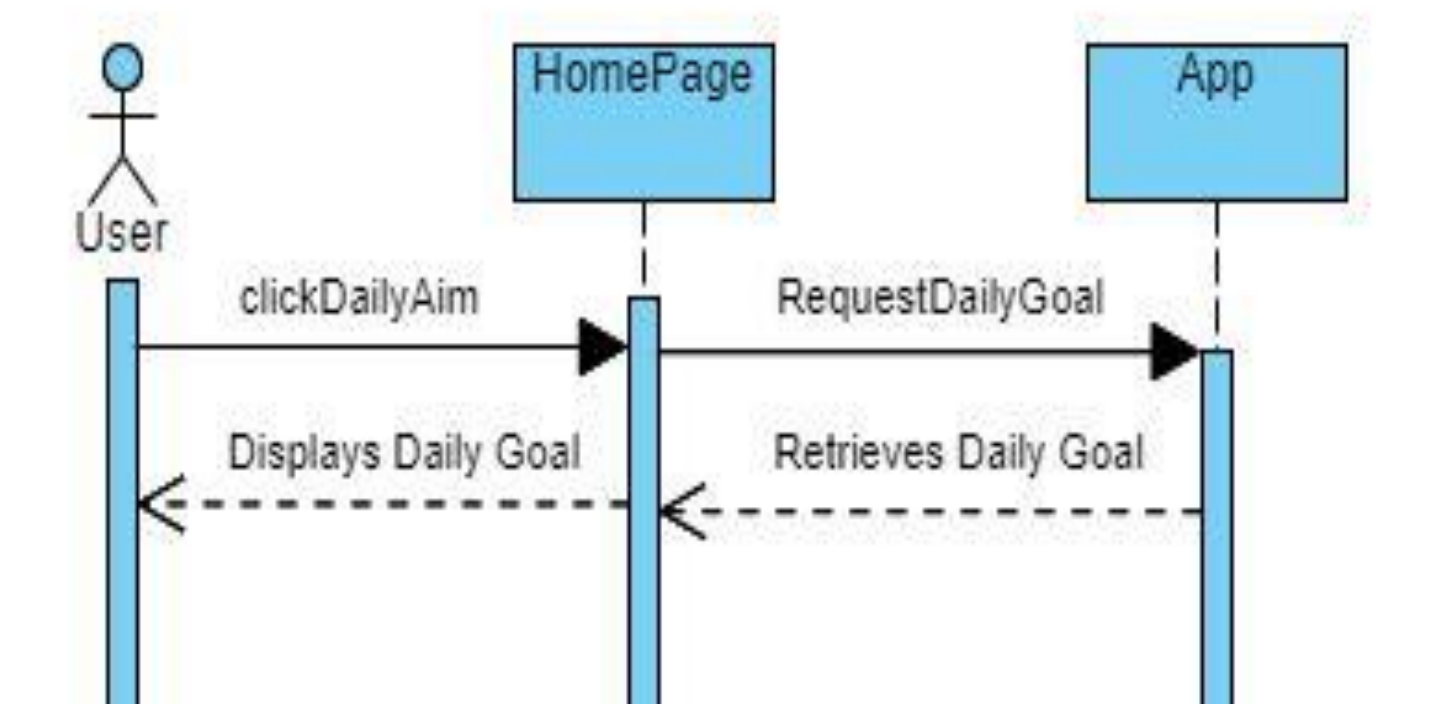
##### Registration



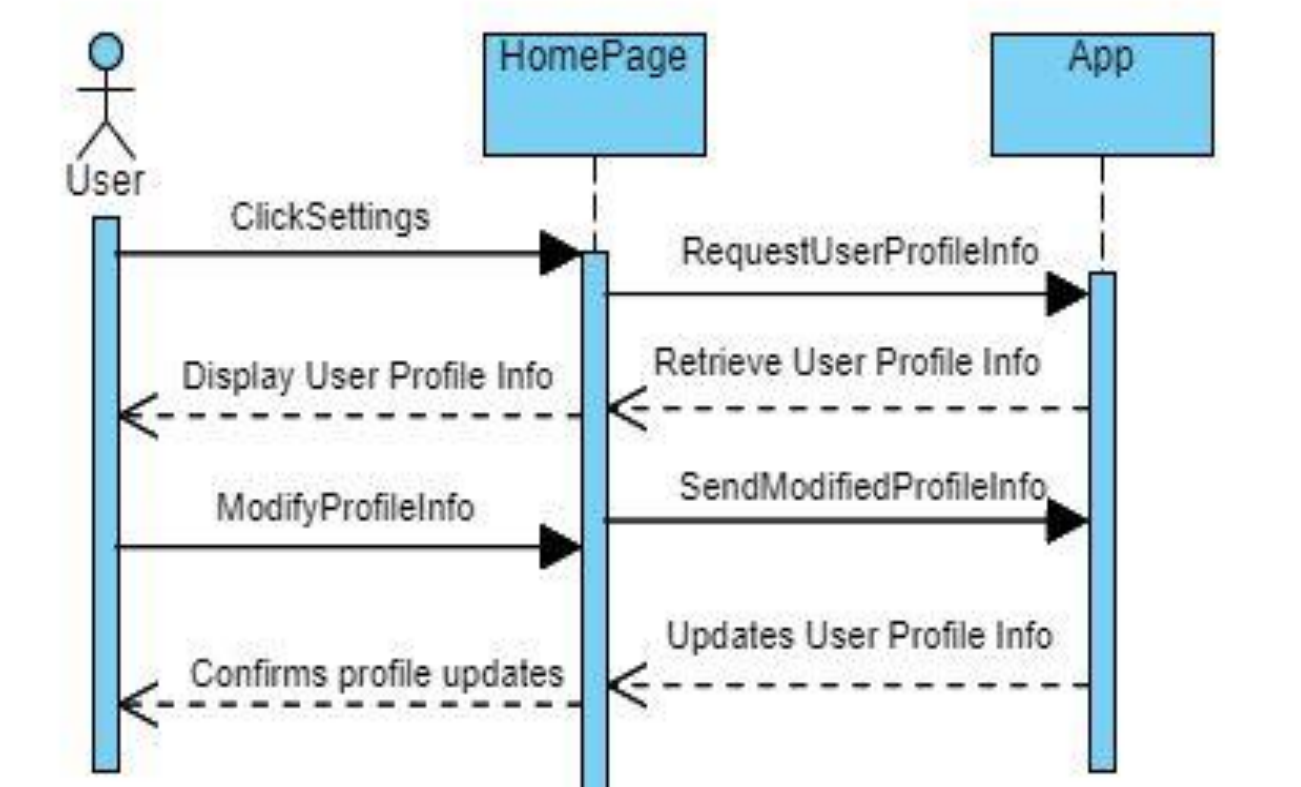
##### Home page, meal list , ingredients



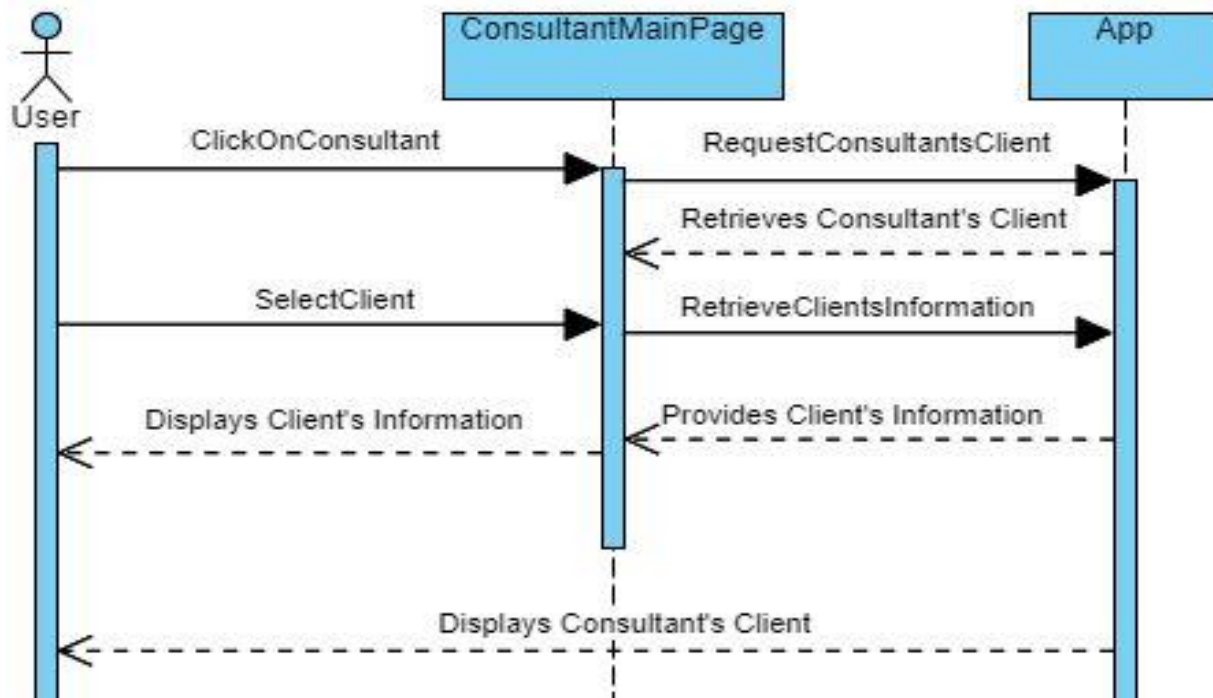
## Daily aim



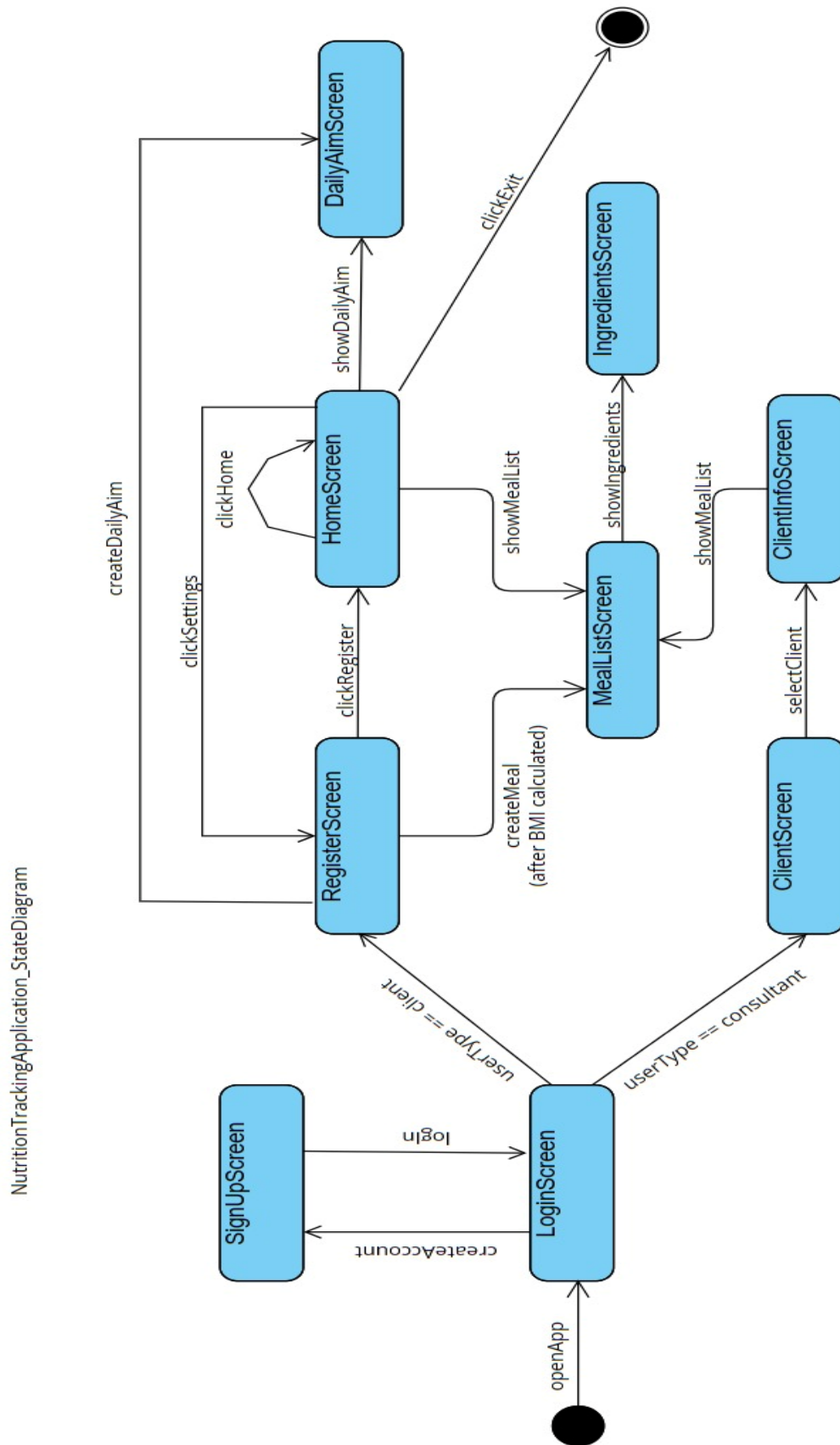
## Settings



## Consultant's Page

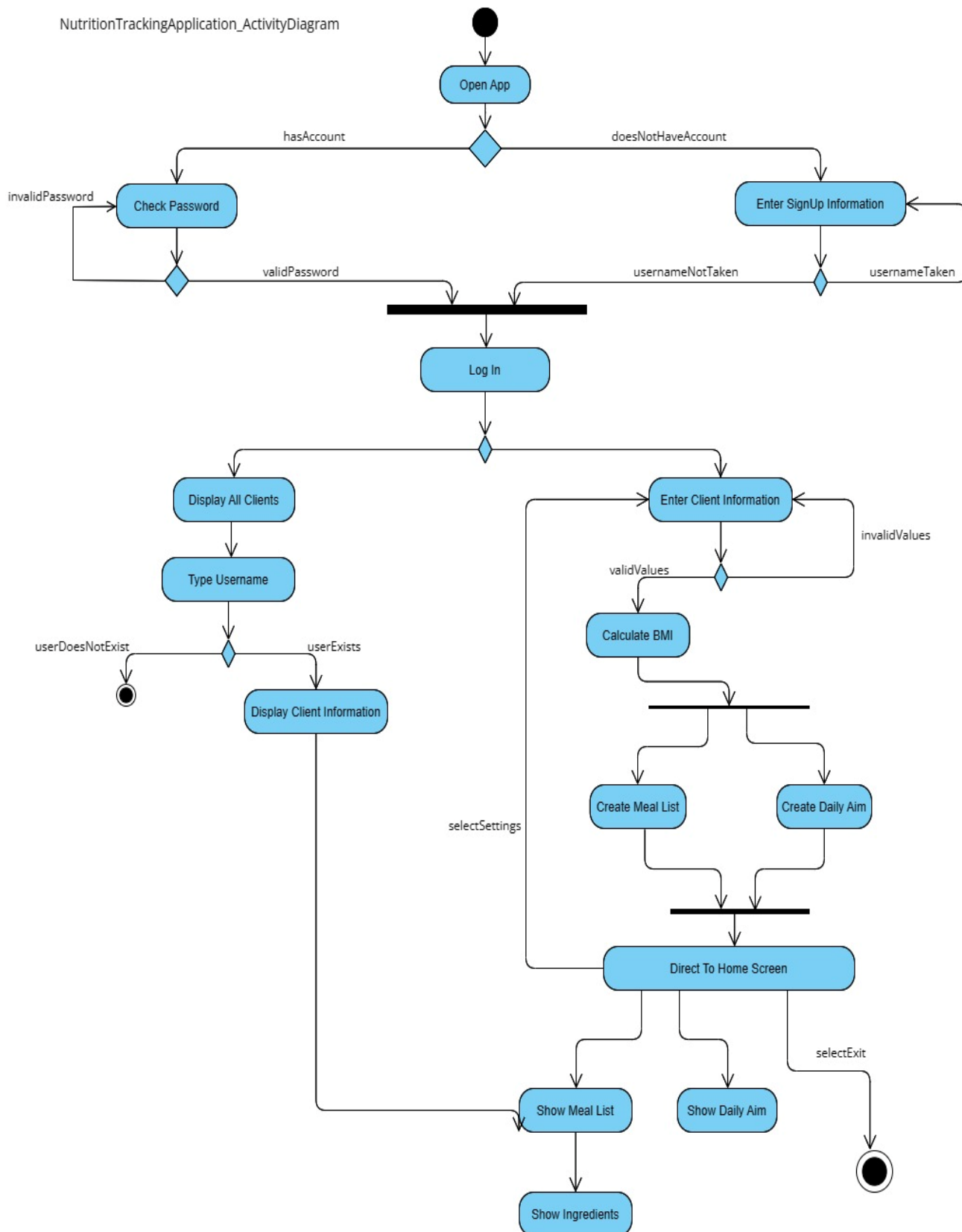


## 3.2) State Diagram



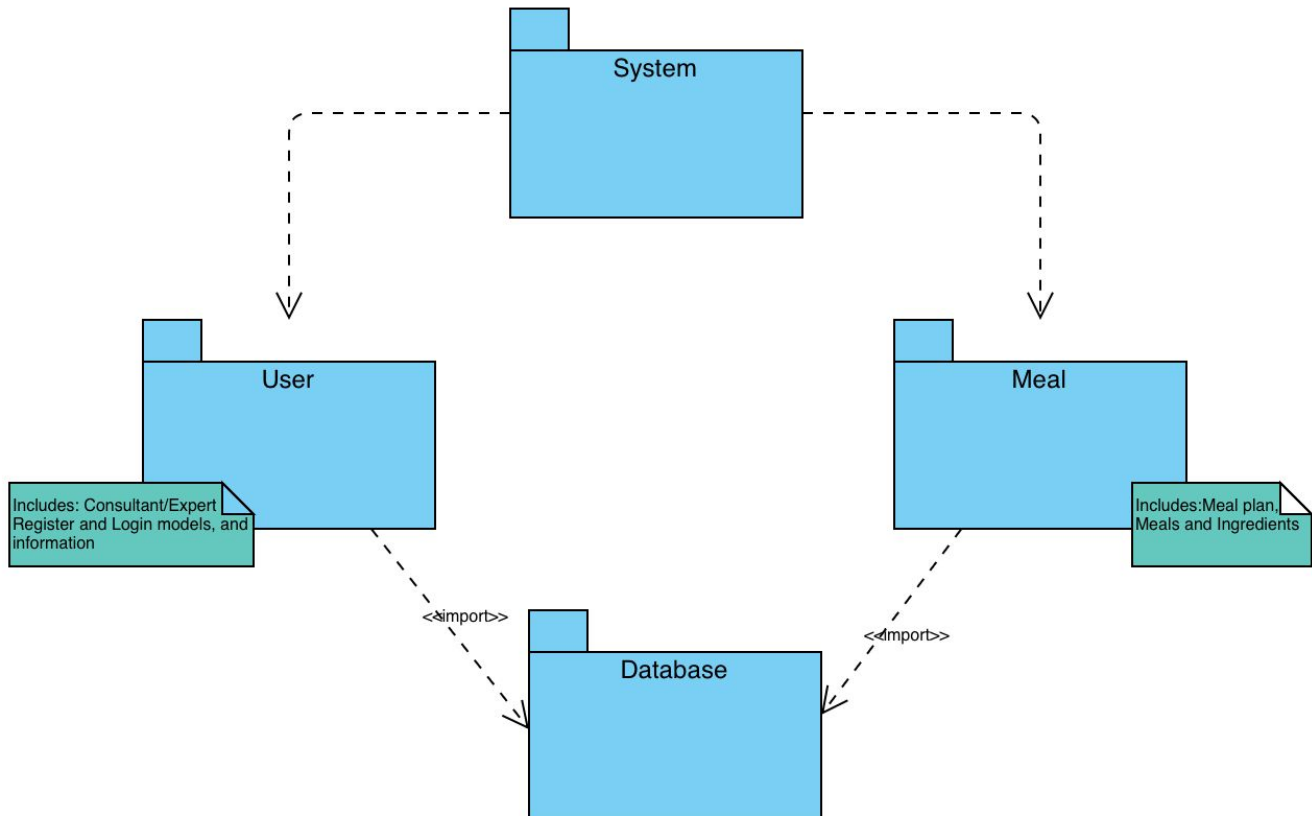


### 3.3) Activity Diagram



## 4) SOFTWARE ARCHITECTURE

### 4.1) UML Package Diagram

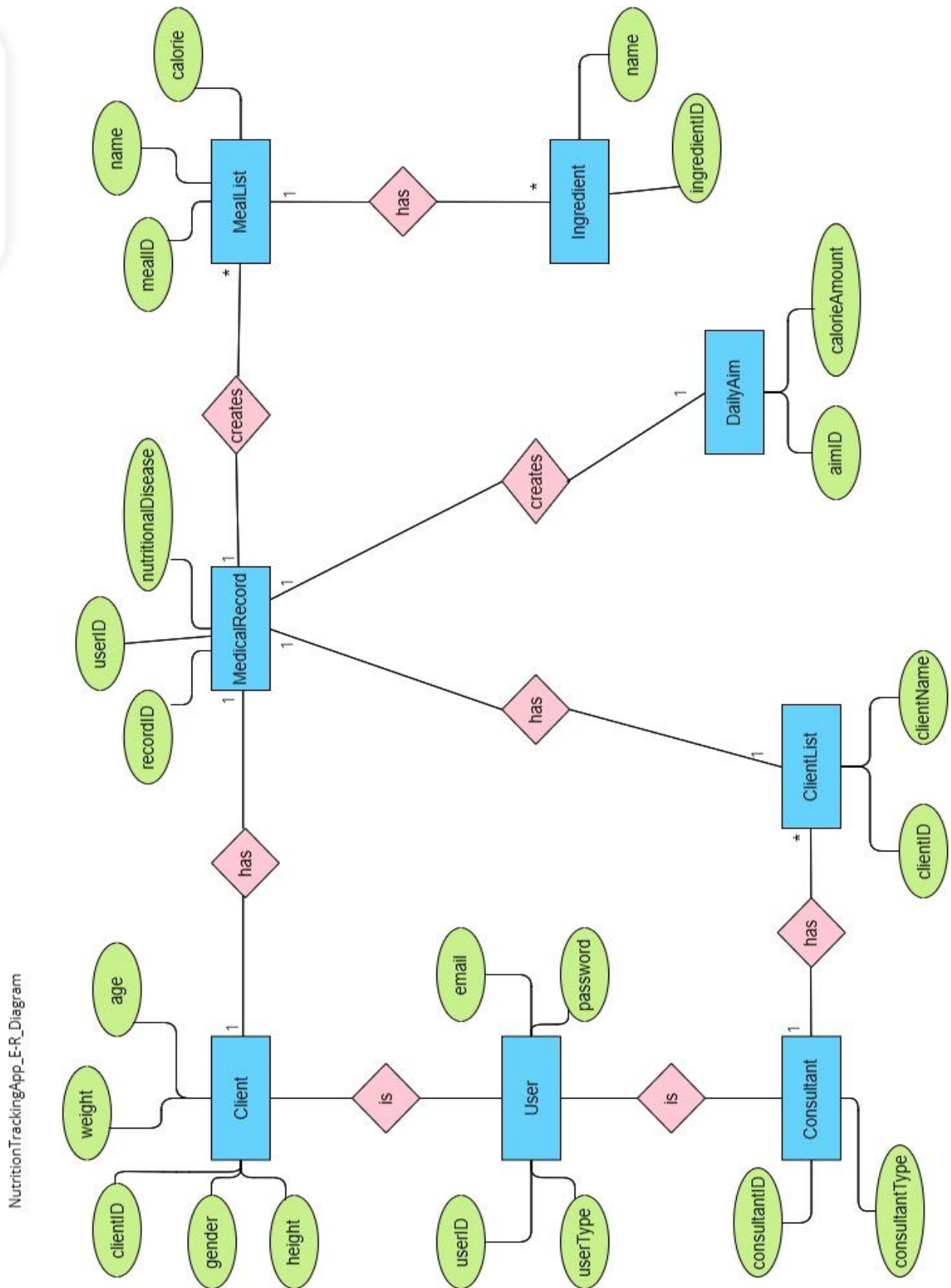


### 4.2) UML Component Diagram

NutritionTrackingApp\_UML\_ComponentDiagram



## 5) ENTITY RELATIONSHIP DIAGRAM



## 6. Glossary and References

### 6.1. Glossary

**Activity Diagram:** an activity diagram provides a view of the behavior of a system by describing the sequence of actions in a process.

**Aggregation:** an aggregation relationship shows a classifier as a part of or subordinate to another classifier.

**Associations:** is a relationship between two classifiers, such as classes or use cases, that describes the reasons for the relationship and the rules that govern the relationship.

**Attribute:** An attribute describes a range of values for that data definition. Attributes describe the structure and value of an instance of a class.

**Component Diagram:** Component diagram breaks down the actual system under development into various high levels of functionality. Each component is responsible for one clear aim within the entire system and only interacts with other essential elements on a need-to-know basis.

**Multiplicity:** Multiplicity in UML allows to specify **cardinality -number of elements** - of some collection of elements.

**Package Diagram:** is used as a central view to represent a software system compile-time logical architecture.

**Retrieve:** Searching for, locating, and returning data.

**Sequence Diagram:** Sequence Diagrams are interaction diagrams that detail how operations are carried out.

**State Diagram:** state diagrams to illustrate the dynamic view of a system. They are especially important in modeling the behavior of an interface, class, or collaboration.

### 6.2. References

1. Visual paradigm
2. <https://cdn-images.visual-paradigm.com/guide/uml/what-is-component-diagram/05-component-diagram-example-using-interface.png>
3. <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-package-diagram/>
4. <https://developer.ibm.com/articles/the-sequence-diagram/>
5. <https://www.geeksforgeeks.org/introduction-of-er-model/>