| HIGH LEVEL DESIGN DOCUMENT  Recipe Recommendation Based On Ingredients  Recognition Using CV and ML  UE19CS390A – Capstone Project Phase – 1  ***Submitted by:***   | **Name**  **A R Manyatha**  **Amulya S Dinesh**  **Manasi Swain**  **Mihir Soni** | **<SRN >**  **PES2UG19CS002**  **PES2UG19CS035**  **PES2UG19CS216**  **PES2UG19CS232** | | --- | --- |   Under the guidance of   | **Prof. Swati Pratap Jagdale**  Assistant Professor  PES University | | --- |   **January - May 2022**  **DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**  FACULTY OF ENGINEERING  **PES UNIVERSITY**  (Established under Karnataka Act No. 16 of 2013)  Electronic City, Hosur Road, Bengaluru – 560 100, Karnataka, India |
| --- | --- | --- | --- |

TABLE OF CONTENTS

| 1. Introduction | 4 |
| --- | --- |
| 1. Current System | 4 |
| 1. Design Considerations | 4 |
| 3.1 Design Goals | 4 |
| 3.2 Architecture Choices | 4 |
| 3.3 Constraints, Assumptions and Dependencies | 4 |
| 1. High Level System Design | 5 |
| 1. Design Description | 5 |
| * 1. Master Class Diagram | 6 |
| 5.2 Reusability Considerations | 6 |
| 1. ER Diagram / Swimlane Diagram / State Diagram | 6 |
| 1. User Interface Diagrams | 6 |
| 1. Report Layouts | 6 |
| 1. External Interfaces | 7 |
| 1. Packaging and Deployment Diagram | 7 |
| 1. Help | 7 |
| 1. Design Details | 7 |
| 12.1 Novelty | 7 |
| 12.2 Innovativeness | 7 |
| 12.3 Interoperability | 7 |
| 12.4 Performance | 7 |
| 12.5 Security | 7 |
| 12.6 Reliability | 7 |
| 12.7 Maintainability | 7 |
| 12.8 Portability | 7 |
| 12.9 Legacy to Modernization | 7 |
| 12.10 Reusability | 7 |
| 12.11 Application Compatibility | 7 |
| 12.12 Resource Utilization | 7 |
| Appendix A: Definitions, Acronyms and Abbreviations | 8 |
| Appendix B: References | 8 |
| Appendix C: Record of Change History | 8 |
| Appendix D: Traceability Matrix | 8 |

# Note:

| **Section – 1 & Section 2** | **Common for Product Based and Research Projects** |
| --- | --- |
| **Section 3 to Section 11** | **High-Level Design for Product Based Projects.** |
| **Section 12** | **High-Level Design for Research Projects.** |
| **Appendix** | **Provide details appropriately** |

# Introduction

Recipes are the universal language of cooking. Through a recipe you quickly understand how a certain dish can be made. In the world revolving around technology, people are always on the run, leading to a busy schedule with no time for self-cafe. Food, nutrition and fitness plays a vital role to help keep up. It is indeed sad to see that people are losing interest in spending time learning and cooking new recipes. There are many websites with plenty of recipes available online, however, they lack real time analysis of ingredients used for cooking. The main objective of the proposed system is to assist users to decide what they can cook with the available resources. By pointing the camera at the food ingredients, users can immediately build a plan or have an idea of what they will be cooking, based on our recommendations along with the nutritional value of the recipe.

There are three main parts to this project: Computer vision to scan the ingredients; Machine learning to label the ingredients; Search/Web Scraping to search for recipes from the dataset based on the ingredients, if the recipe is not present, perform web scraping.

# Current System

We reviewed how visual content, context and external knowledge can be integrated effectively into food-oriented applications, with special focus on recipe analysis and retrieval, food recommendation as emerging direction.

We largely rely on contextual and prior information. Similarly, context and prior knowledge can be integrated in automatic food analysis systems.

People enjoy food photography because they appreciate food. Behind each meal there is a story described in a complex recipe and, unfortunately, by simply looking at a food image we do not have access to its preparation process, this inspired the project on inverse cooking.

Our project does the opposite of this i.e. we take pictures of ingredients and provide the user with the list of recipes.

1. **Design Considerations** 
   1. **Design Goals**
2. The existing system proposed a cooking recipe recommendation system by employing object recognition for food ingredients such as vegetables and fruits. By pointing a mobile phone camera towards food ingredients, a user receives a recommendation list.

The newly proposed project aims to recognise various items in the pantry which includes fruits, vegetables, meat etc. The model also provides a choice of diet (diabetics, vegetarian, vegan, etc) to the user. On recognizing these, a recipe list will be generated and provided to the user with calories. User has the option to choose the desired recipe.

1. *Availability*:

The server should be available on specified time as many users are waiting for the recipe to be generated.

1. *Security and privacy:*

Authentication of user whenever he/she logs into the system. The users will have private user profiles which will store data of the the recipes that they’ve used from our application previously on a MongoDB server which will be kept hidden from the public. MongoDB has a multitude of security features, such as encryption, authentication, role-based access control, TLS/SSL encryption and many more. We will have a user\_id and a password for authentication which will be logged in in the mobile application.

1. *Speed:*

The model requires a server with high speed internet capability. The user should be equipped with fair network bandwidth to send images and receive recipes.

The system will be designed to be *highly scalable*, ie, it will be able to process a wide range of ingredients efficiently.

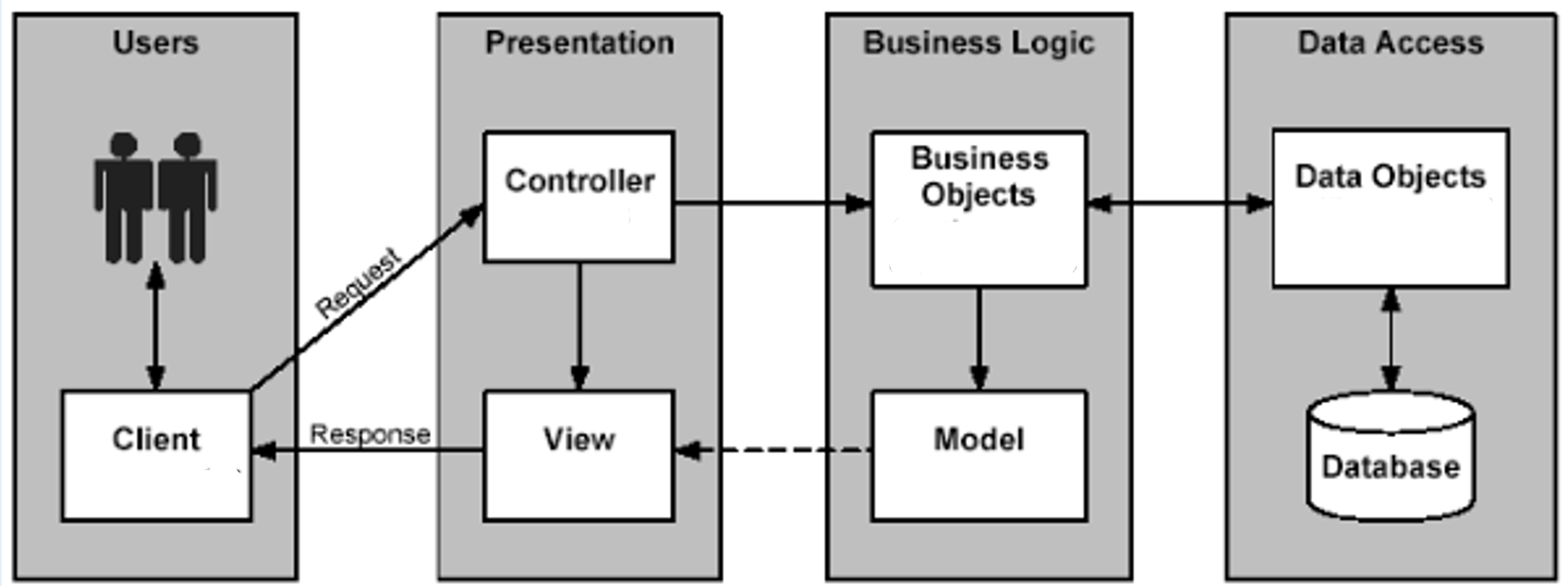
* 1. **Architecture Choices**

*Choice of Architecture considered* : Layered (n-tier) architecture

This architecture can be used to structure programs that can be decomposed into groups of subtasks, each of which is at a particular level of abstraction. Each layer provides services to the next higher layer.

The 4 layers of our system are as follows :

1. Presentation Layer – responsible for user interactions with the software system
2. Application Layer – handles aspects related to accomplishing functional requirements
3. Domain Layer – responsible for algorithms, programming components
4. Database Layer – responsible for handling databases and data.



Pros of layered architecture:

1. The layered architecture is relatively easier to understand and implement. The layers are self-explanatory. They help in communicating with other layers in the architecture.
2. All the operation is dependent on one another in the architecture and is consistent with all the layers in the system, ie, it is sequential.
3. Data transfer is consistent in layered architecture.
4. Layering helps to differentiate between the tasks assigned to each layer of the architecture so that when a task has to be identified, it is easy to figure it out using the layering structure.
5. Within a layer, scalability is possible as the objects increase in the project. This helps to identify the objects working within the layer and to assign tasks to the layers.
6. This type of system architecture improves agility.

Cons of layered architecture:

1. No dependency inversion: In a layered architecture, the dependencies are direct and conceptually changed into essential higher layers from a low-level infrastructure layer.
2. Parallel processing is not possible.
3. Hidden use cases: It is difficult to determine the use cases of your project by simply checking the code organization. You need to refer to the class names and in most cases, implementation.

*Alternate choices of architecture:*

Microservice Architecture: It is a variant of the service-oriented architecture structural style. It arranges an application as a collection of loosely-coupled services.

Pros of microservice architecture:

1. Scaling up becomes easier
2. Leads to Improved Fault Tolerance

Cons of microservice architecture:

1. Increased Complexity of Communication Between the Services
2. Requires More Resources
3. Relatively Complex Deployment

# Constraints, Assumptions and Dependencies

1. Interoperability requirements

We will create our project such that the server with the model, the database, as well as the user interface will be able to converse with each other and share data flawlessly. It will ideally support all the latest versions of the operating system from the users side. The only constraint may be that older versions of the operating system from the users side may not have some of the modules inbuilt that we may use for our project rendering then unable to use our system.

1. Interface/protocol requirements

We will be using a mobile phone camera to take the pictures/video and send it to our computer using wifi. It will only be using standard wifi to match the compatibility standards of modern handheld devices and for reliability.

1. Distribution requirements

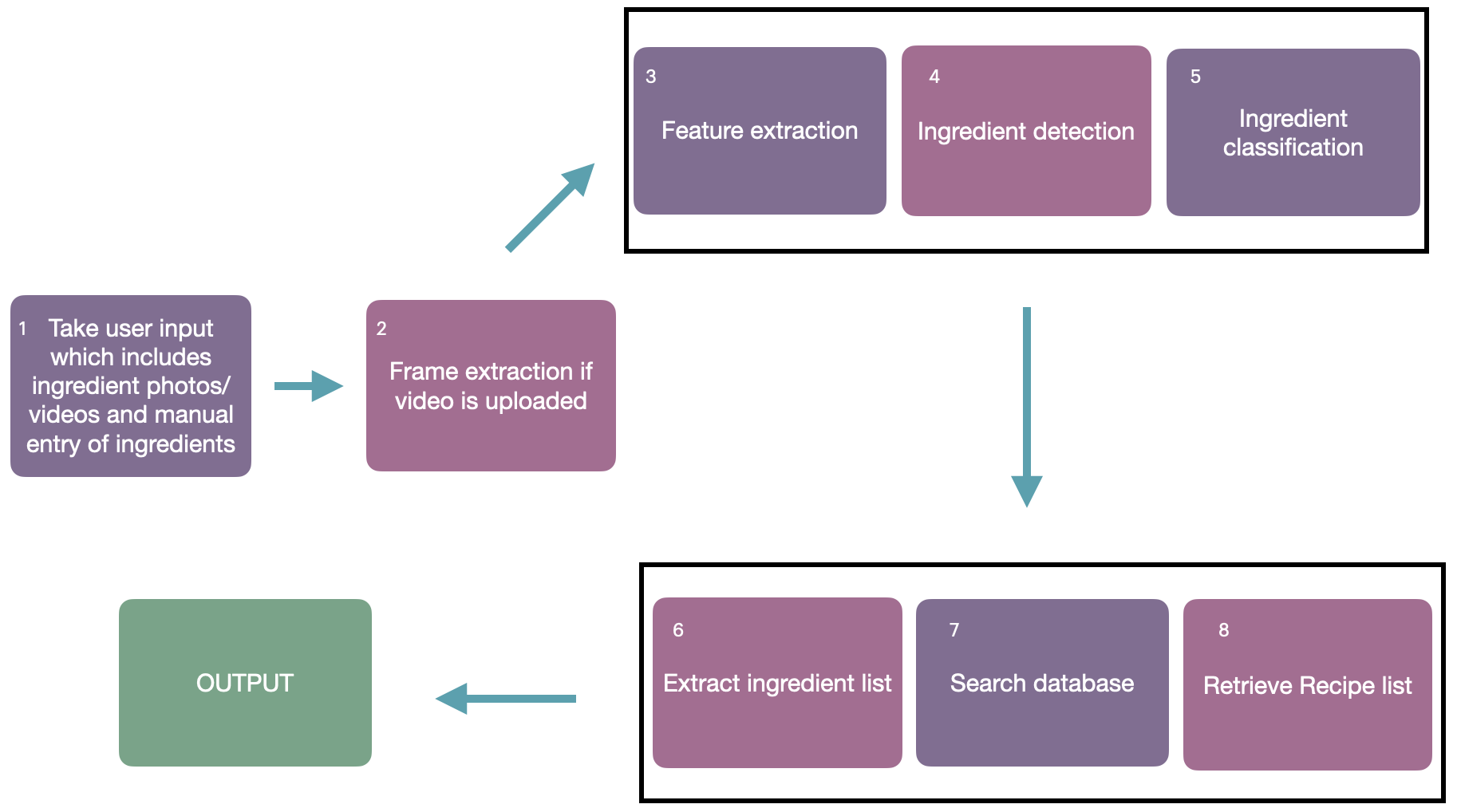
It ensures the availability of the right amount of materials (quantity) at the right time.

1. End-user environment.

The model requires a server with high speed internet capability. The end user should be equipped with fair network bandwidth to send images and receive recipes, without which the user might not have a pleasant experience using the application.

1. Availability of Resources
2. Server - server should be up in running along with being scalable at all times. It should be able to handle a large number of requests in real time and process it accordingly.
3. Storage - the back-end of our system should be able to hold a large database that will be useful for recommending recipe lists.
4. Android device - the device should be of newer compatible versions with a good working camera and should have enough storage space to accommodate our app.
5. Hardware or software environment:
6. Enough storage in server
7. Able to connect to all users requesting for recipes
8. Software run has no latency
9. Hardware limitations:
10. low camera quality
11. low lighting (bad quality captured)
12. Storage limitations of the server
13. Criticality of application: The most important component of the application would be the code that contains the ML model. It is vital that this piece of software must not fail.
14. Safety and security consideration: There is no safety consideration needed as there is no physical component in the project. Security concerns may include server attacks that may inhibit normal functioning of the application.

# High Level System Design



The system elements are identified from different perspectives:

1. Physical/Conceptual – Refer to Section 9 (Packaging and deployment diagram)
2. Security – Authentication of user whenever he/she logs into the system. The users will have private user profiles which will store data of the recipes. We will have a user\_id and a password for authentication which will be logged in in the mobile application.

# Design Description

* 1. **Master Class Diagram**

The following figure is the representation of a master class diagram of the system, which is given at a high level and then broken down into sub levels. Each class has attributes and methods defined.

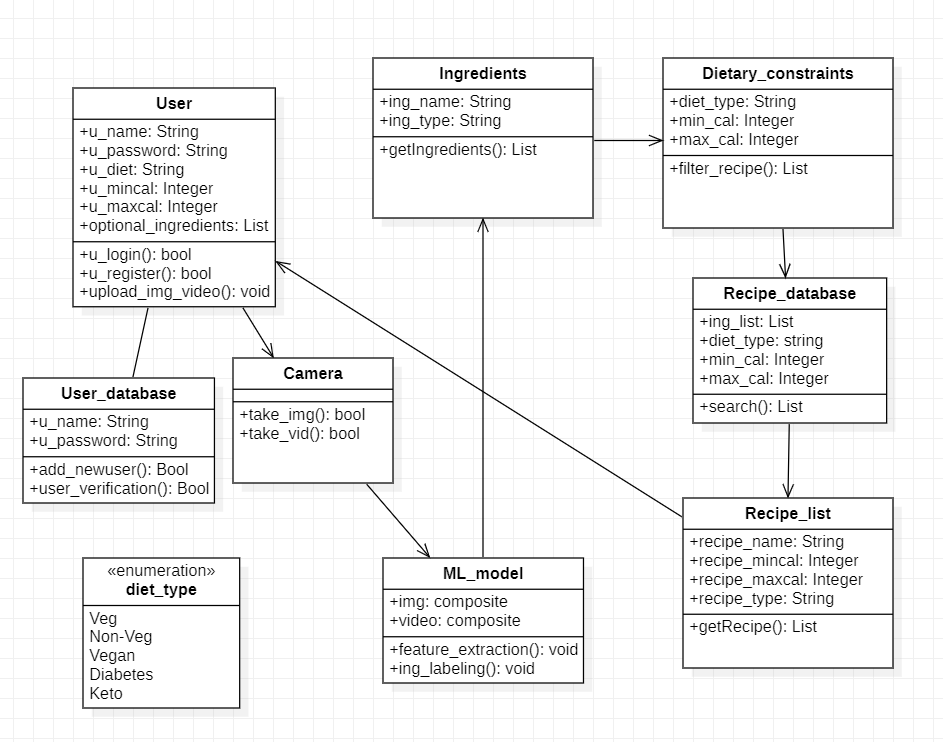


figure 5.1

* 1. **Reusability Considerations**

1. Project Components that are and can be generated with available reusable components.
2. Components that can be built in the project for reuse in other projects.

Object detection algorithms can be reused. On training the model on different datasets, it can be able to identify different kinds of objects other than food ingredients.

1. **ER Diagram / Swimlane Diagram / State Diagram (include as appropriate)**

We have represented our system using a swimlane diagram. A swimlane is used in process flow diagrams, or flowcharts, that visually distinguishes job sharing and responsibilities for sub-processes of a business process.

It delineates who does what in a process. There three vertical swimlanes, namely,

1. User
2. Model
3. Database

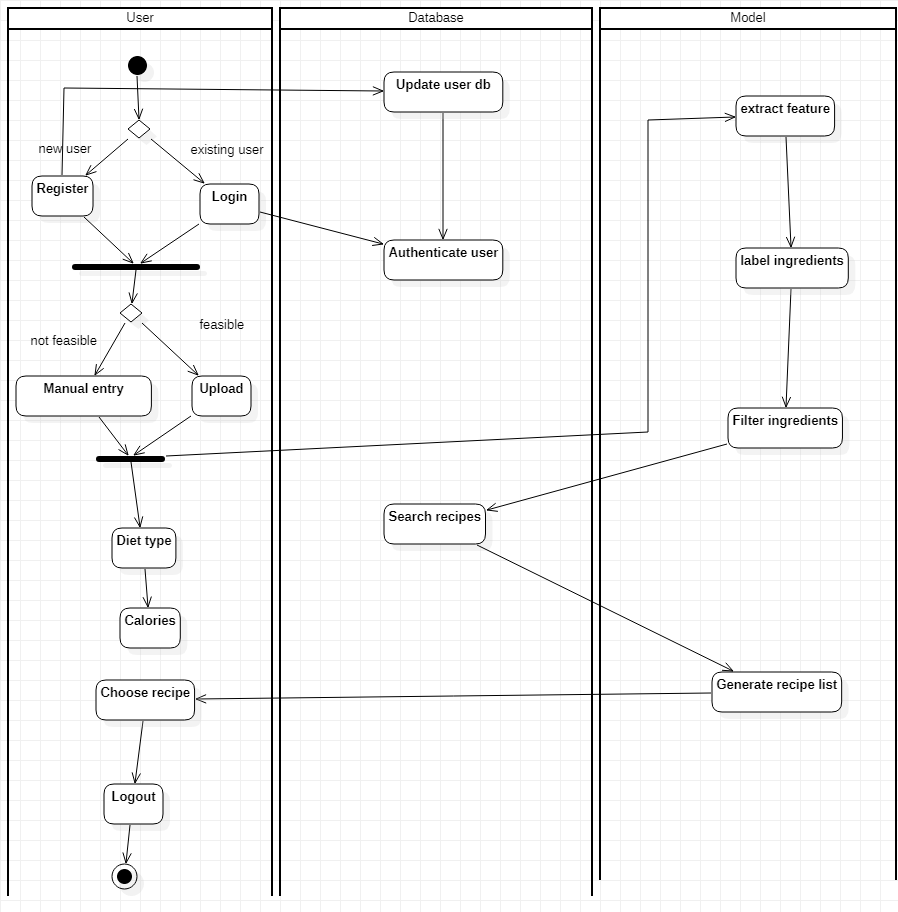


figure 6

1. **User Interface Diagrams**

A User Interface diagram is modeled in terms of its internal structure and objects comprising it, the same as the rest of the application.

The two actors in our case are:

1. User
2. Database

This diagram shows the actors along with a rough list of their actions and their relationships.

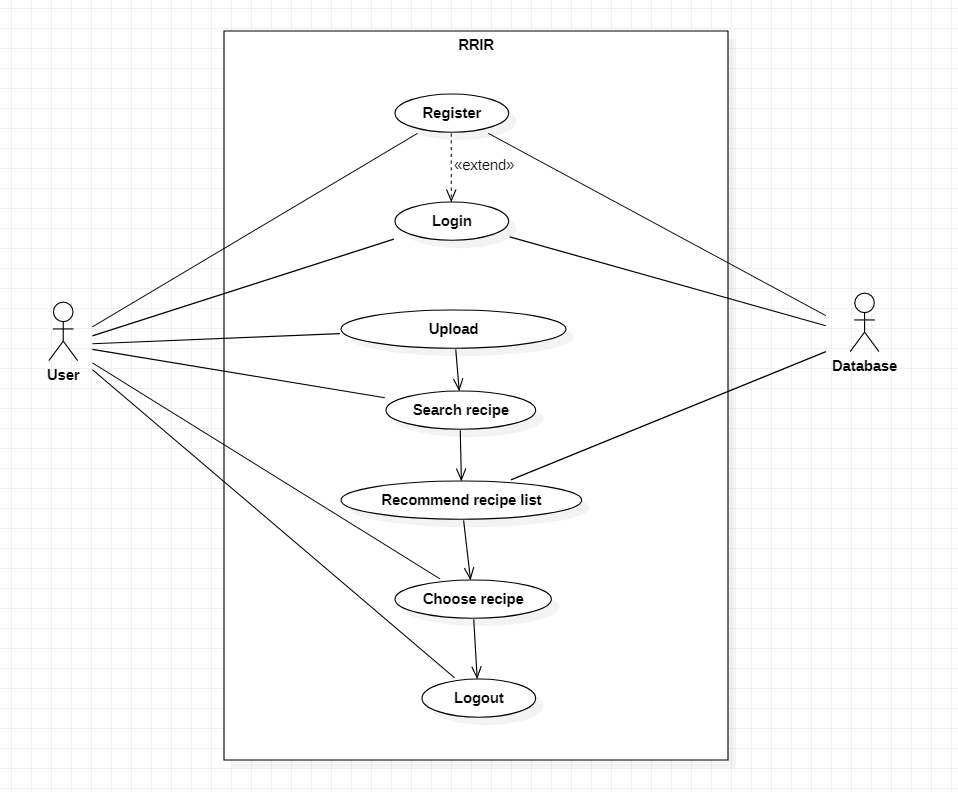


figure 7.1

This diagram is the expansion of the use case diagram which expands onto the login that the user provides.

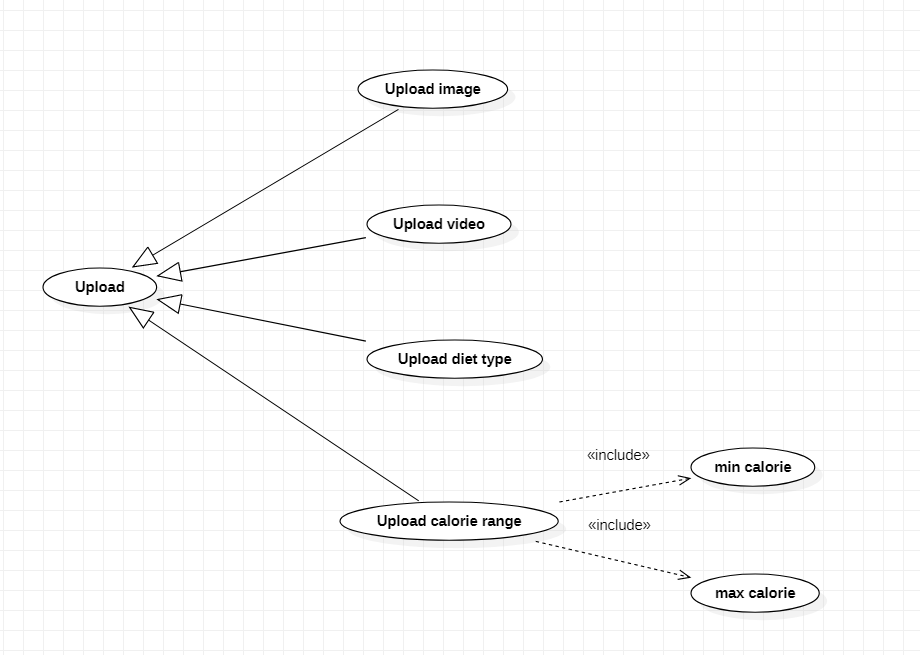


figure 7.2

This diagram is also a part of the use case diagram which expands onto the rough flow of the program with the actors being User and Database.

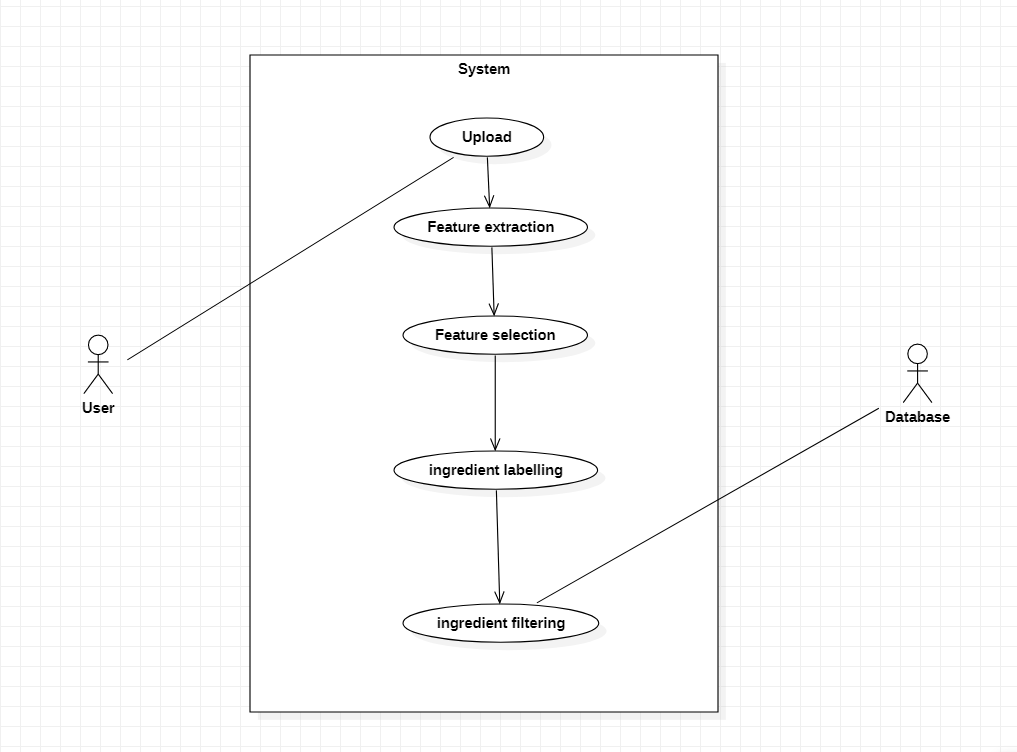


figure 7.3

1. **External Interfaces**

The external interface diagram shows the hardware and software interfaces and their relationships with each other. One side is the real time processing and the other is the user interface. It also shows how the actors (User and Database) converses with the two interfaces.

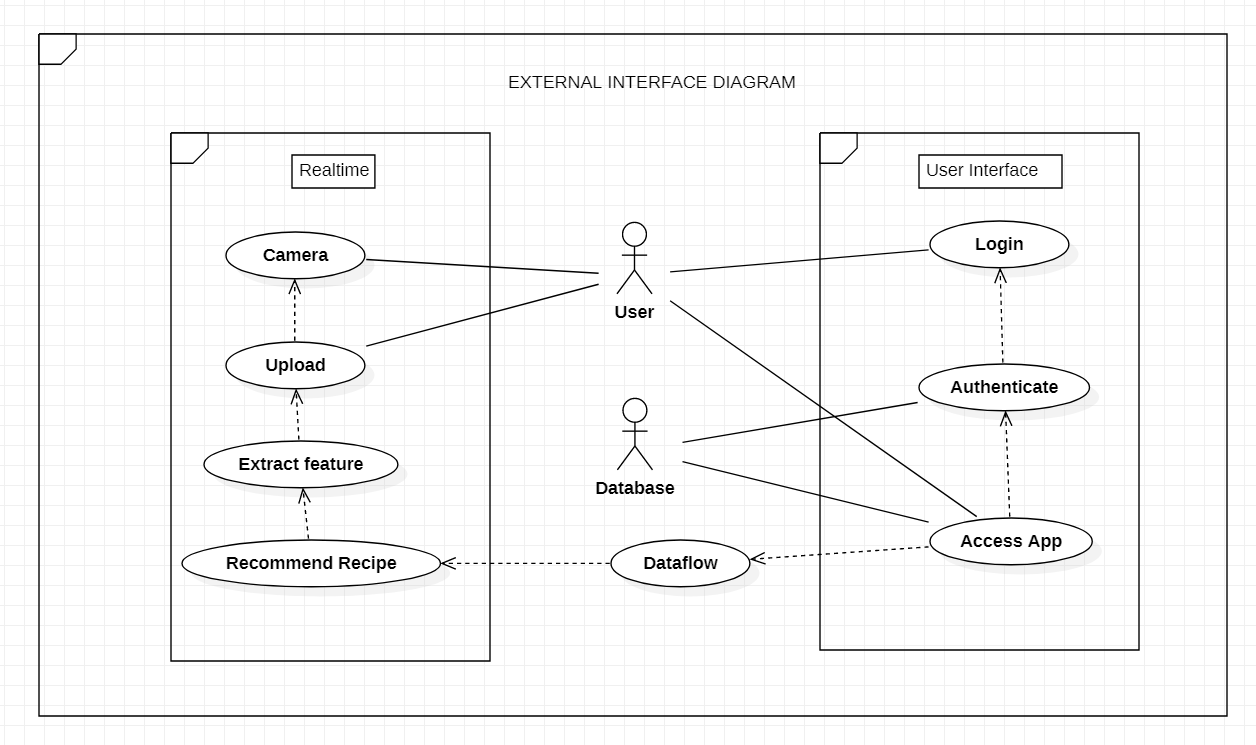


figure 8

1. **Packaging and Deployment Diagram**

A deployment diagram is a UML diagram type that shows the execution architecture of a system, including nodes such as hardware or software execution environments, and the middleware connecting them.

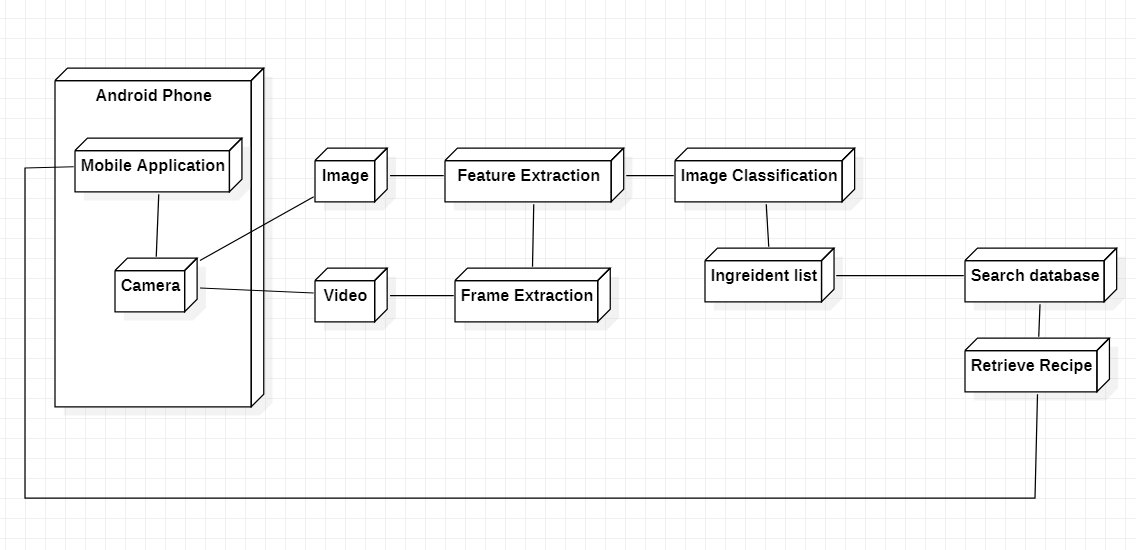


figure 9

# Help

An Upload button will be available on the UI. On clicking on this, the camera on the phone will be accessed and the user will have to click images or shoot videos. Good and high speed internet should be ensured for better experience on the application. Camera quality of more than 10 megapixel is advisable. The ingredients that cannot be scanned, should be manually entered with correct spellings for more accurate recommendations. The diet type, if necessary can be chosen from the predefined list. The user also has the freedom to enter the minimum and maximum calorie value that is desired. On successfully uploading the details, a list of matching recipes will be generated, for the user to choose from.

1. **Design Details**
   1. **Novelty**

Our project not only focuses on recommending a recipe list, but also, caters to the needs of the user by considering diet choice and calorie range.

* 1. **Innovativeness**

As far as our literature survey is concerned, we haven't come across any project which considered the type of diet along with calorie count. This will be inculcated into our project to make it more practical for everyday use.

* 1. **Interoperability**

We will create our project such that the server with the model, the database, as well as the user interface will be able to converse with each other and share data consistently. It will ideally support all the latest versions of the operating system from the users side.

* 1. **Performance**

Availability: The server should be available on specified time as many users are waiting for the recipe to be generated .

Correctness: Accurate recipes should be recommended to the user in real time.

Usability: The model should handle multiple requests from multiple users.

* 1. **Security**

Authentication of user whenever he/she logs into the system. The users will have private user profiles which will store data of the the recipes that they’ve used from our application previously on a MongoDB server which will be kept hidden from the public. MongoDB has a multitude of security features, such as encryption, authentication, role-based access control, TLS/SSL encryption and many more. We will have a user\_id and a password for authentication which will be logged in in the mobile application.

* 1. **Reliability**

The model requires a server with high speed internet capability. The user should be equipped with fair network bandwidth to send images and receive recipes

* 1. **Maintainability**

The administrators should ensure that the server is up and running and not overloaded, ensuring it is well maintained.

* 1. **Portability**

Portability is the ease with which a software system can be transferred from its current hardware or software environment to another environment. Our system is portable on all android devices released after 2015.

* 1. **Reusability**

Object detection is the first phase of our project that can be reused. On training the model on different datasets, it can be able to identify different kinds of objects other than food ingredients. Hence, object detection algorithm can be reused.

* 1. **Application compatibility**

The mobile application is Android based, and can be used on all Android devices released after 2015.

* 1. **Resource utilization**

Server - server should be up in running along with being scalable at all times. It should be able to handle a large number of requests in real time and process it accordingly.

Storage - the back-end of our system should be able to hold a large database that will be useful for recommending recipe lists.

Android device - the device should be of newer compatible versions with a good working camera and should have enough storage space to accommodate our app.

# Appendix A: Definitions, Acronyms and Abbreviations

1. ML : Machine Learning
2. CV : Computer Vision
3. RRIR : Recipe Recommendation Based On Ingredients Recognition Using CV and ML
4. Image labeling: It is the process of identifying and marking various details in an image. Image labeling is useful when automating the process of generating metadata or making recommendations to users based on details in their images

# Appendix B: References

1. Keiji Yanai, Takuma Maruyama, Yoshiyuki Kawano, “A Cooking Recipe Recommendation System with Visual Recognition of Food Ingredients”, 2014
2. Gangothri R. Sanil , Anusha Anchan , Vijetha Achar, Kavya Hegde, “KNN based Recipe Retrieval using Ingredient Recognition”, 2018
3. W. Ling et al. , “Finding function in form: Compositional character models for open vocabulary word representation,” in Proceedings of the
4. 2015 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, 2015, pp. 1520-1530
5. Yu-Hsien Ting Qiangfu Zhao, Rung-Ching Chen,”Dietary Recommendation Based on Recipe Ontology”, pp. 1-6, ISSN: 23255986, 6th International Conference on Awareness Science and Technology (iCAST) ,IEEE, 2014
6. Nayani R1 , Pooja KR2 , Ranjani Bahadur3 , Vibha Y4, “RECIPE MANAGEMENT SYSTEM”, International Journal For Technological Research In Engineering Volume 3, Issue 9, May-2016 ISSN (Online): 2347 – 4718.
7. Pathanjali 1 , Vimuktha E Salis , Jalaja G , Latha A, “A Comparative Study of Indian Food Image Classification Using K-Nearest-Neighbour and Support-Vector-Machines”, International Journal of Engineering &Technology, 7 (3.12) (2018) 521-525.
8. Anthimopoulos, M.M.; Gianola, L.; Scarnato, L.; Diem, P.; Mougiakakou, S.G. “A Food Recognition System for Diabetic Patients Based on an Optimized Bag-of-Features Model.” JBHI, 2014 ,18, 1261–1271

# Appendix C: Record of Change History

| **#** | **Date** | **Document Version No.** | **Change Description** | **Reason for Change** |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |

# Appendix D: Traceability Matrix

| **Project Requirement Specification Reference Section No. and Name.** | **DESIGN / HLD Reference Section No. and Name.** |
| --- | --- |
| 1. Introduction | 1. Introduction |
| 6.3 Security Requirements | 3.1 Design goals |
| 6.1 Performance Requirements | 1.4 Performance |