IR@AIKTC aiktcdspace.org

A PROJECT REPORT

ON

"AUTOMATED DIET PLANNER"

Submitted to

UNIVERSITY OF MUMBAI

In Partial Fulfilment of the Requirement for the Award of

BACHELOR'S DEGREE IN
COMPUTER ENGINEERING

BY

MALBARI SABIYA ABDUL RASHID SHAHEEN	**15CO06**
ALEKAR IFAT SALIM MEHTAB	**15CO02**
SHAIKH MARIYA IRFAN AHMED REHANA	**14CO07**

UNDER THE GUIDANCE OF
PROF. SHAIKH ABDUL SALAM

DEPARTMENT OF COMPUTER ENGINEERING
Anjuman-I-Islam's Kalsekar Technical Campus
SCHOOL OF ENGINEERING & TECHNOLOGY
Plot No. 2 3, Sector - 16, Near Thana Naka,
Khandagaon, New Panvel - 410206

2018-2019

AFFILIATED TO

UNIVERSITY OF MUMBAI

Service By KRRC (Central Library)

IR@AIKTC aiktcdspace.org

A PROJECT II REPORT
ON

‘AUTOMATED DIET PLANNER”

Submitted to
UNIVERSITY OF MUMBAI

In Partial Fulfilment of the Requirement for the Award of

BACHELOR’S DEGREE IN
COMPUTER ENGINEERING

BY

MALBARI SABIYA ABDUL RASHID SHAHEEN	**15CO06**
ALEKAR IFAT SALIM MEHTAB	**15CO02**
SHAIKH MARIYA IRFAN AHMED REHANA	**14CO07**

UNDER THE GUIDANCE OF
PROF. SHAIKH ABDUL SALAM

DEPARTMENT OF COMPUTER ENGINEERING
Anjuman-I-Islam’s Kalsekar Technical Campus
SCHOOL OF ENGINEERING & TECHNOLOGY
Plot No. 2 3, Sector - 16, Near Thana Naka,
Khandagaon, New Panvel - 410206

2018-2019
AFFILIATED TO

UNIVERSITY OF MUMBAI

Service By KRRC (Central Library)

IR@AIKTC aiktcdspace.org

Anjuman-i-Islam's Kalsekar Technical Campus

Department of Computer Engineering
SCHOOL OF ENGINEERING & TECHNOLOGY
Plot No. 2 3, Sector - 16, Near Thana Naka,
Khandagaon, New Panvel - 410206

CERTIFICATE

This is certify that the project entitled

"AUTOMATED DIET PLANNER"

submitted by

MALBARI SABIYA ABDUL RASHID SHAHEEN	**15CO06**
ALEKAR IFAT SALIM MEHTAB	**15CO02**
SHAIKH MARIYA IRFAN AHMED REHANA	**14CO07**

is a record of bonafide work carried out by them, in the partial fulfilment of the requirement for the award of Degree of Bachelor of Engineering (Computer Engineering) at ***Anjuman-I-Islam's Kalsekar Technical Campus, Navi Mumbai*** under the University of MUMBAI. This work is done during year 2018-2019, under our guidance.

Date: / /

(Prof. SHAIKH ABDUL SALAM)
Project Supervisor

(Prof. KALPANA BHODKE)
Project Coordinator

(Prof.TABREZ KHAN)
HOD, Computer Department

DR. ABDUL RAZAK HONNUTAGI
Director

External Examiner

Service By KRRC (Central Library)

Acknowledgements

We would like to take the opportunity to express my sincere thanks to our guide **Prof. SHAIKH ABDUL SALAM**, Assistant Professor, Department of Computer Engineering, AIKTC, School of Engineering, Panvel for his invaluable support and guidance throughout my project research work. Without his kind guidance & support this was not possible.

We are grateful to him for his timely feedback which helped us track and schedule the process effectively. His/her time, ideas and encouragement that he gave is help me to complete my project efficiently.

We would like to express our deepest appreciation towards **DR. ABDUL RAZAK HONNUTAGI**, Director, AIKTC, Navi Mumbai, **Prof. TABREZ KHAN**, Head of Department of Computer Engineering and **Prof. KALPANA BODKE**, Project Coordinator whose invaluable guidance supported us in completing this project.

At last we must express our sincere heartfelt gratitude to all the staff members of Computer Engineering Department who helped us directly or indirectly during this course of work.

MALBARI SABIYA ABDUL RASHID SHAHEEN
ALEKAR IFAT SALIM MEHTAB
SHAIKH MARIYA IRFAN AHMED REHANA

Project I Approval for Bachelor of Engineering

This project entitled ***"Automated Diet planner"*** by ***Malbari Sabiya Abdul Rashid Shaheen,Alekar Ifat Salim Mehtab,Shaikh Mariya Irfan Ahmed Rehana*** is approved for the degree of ***Bachelor of Engineering in Department of Computer Engineering.***

Examiners

1. .....

2. .....

Supervisors

1. .....

2. .....

Chairman

.....

Declaration

We declare that this written submission represents my ideas in my own words and where others ideas or words have been included,We have adequately cited and referenced the original sources. We also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission.We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

MALBARI SABIYA ABDUL RASHID SHAHEEN
15CO06

ALEKAR IFAT SALIM MEHTAB
15CO02

SHAIKH MARIYA IRFAN AHMED REHANA
14CO07

ABSTRACT

In today's fast paced and busy world, it becomes very difficult for people to take proper care of their health. In developing countries like India, majority of working-class people suffer from nutritional disorders such as malnutrition and over- nutrition because of consuming unhealthy, processed foods. Since the individual's health mainly depends on the food intake, it is vital to have a proper diet based on the BMI value along with any existing health issues such as diabetes, cardiac, kidney,etc. Users will require different nutritional intake for their body due to variation in height, weight, age, eating habits, daily activity level.

Our proposed system calculates the earned calories based on daily eating habits along with calorie requirements for the given user based on the BMI value and generates a diet plan by using machine learning techniques. A comparison of machine learning algorithms such as Decision tree regressor, Random Forest, J48 and CART will be applied for recommending the diet for users between 15- 40 age groups.

Keywords: Random forest, Decision tree regressor,j48,CART,BMI,

Contents

List of Figures

List of Tables

Chapter 1

Introduction

People nowadays are very concerned about their health. They care about what they eat. However, most of them lack knowledge about how to choose and what kind of food servings is suitable for their body. This is because, different people will have different needs of diet,according to variety of age, weight, height, gender, and lifestyle. Since it is difficult to maintain a healthy lifestyle,it is quite important to have a healthy balanced diet.

India is one of the countries that have diet issue.Currently, typical Indian meal comprises of good carbs, healthy natural protein and resistant starch which is beneficial in weight loss. Lower amounts of fats and animal proteins are consumed. Currently, the influence of westernization, fast food chains and processed foods with refined carbohydrate and high animal protein intake, as well as inactivity, have contributed to the problems of obesity and cardiovascular diseases.

1.1 Purpose

The purpose of our project is to generate nutrition plan for particular age groups to improve their health and lifestyle.Being healthy is not merely the absence of disease or infirmity, but it is a state of complete physical, mental, and social well-being. Healthy diet is the healthy eating habit. A good diet is the well-balanced food intake that fulfills all the body's need.

1.2 Project Scope

There is a need in the society of a well balanced,healthy diet in order to sustain a healthy lifestyle.The proposed system is the Diet Recommendation system which gets the raw data by calculating the required calorie using height, weight and age.Further food recommendation will be done using classifiers such as ID3, Random Forest, Decision tree regressor which will recommend the diet plan for the user.The provided diet plan will be beneficial for users if any user is suffering from diabetes,heart disease or kidney disease

Service By KRRC (Central Library)

1.3 Project Goals and Objectives

1.3.1 Goals

The diet management system is for quick generation of diet plan for individuals according to their BMI,calorie needs and categorized mainly based on diseases such as heart,kidney disease or diabetes.

1.3.2 Objectives

In this project we will help individual maintain their health by giving them proper diet plan using current technology such as machine learning algorithms.This new technology is very Effective in terms of recommendation of different food item. Following are the objectives of our system:

- To help user select a proper food item.
- To help users understand what type of food serving is suitable for them
- To help nutrition and health outcome of young and working generation.

1.4 Organization Of Report

- Chapter 1: This chapter consists of introduction related to project such as purpose,project scope,along with project goals and objective.
- Chapter 2:This chapter consists of Literature survey in which we did study of technical papers related to our project.
- Chapter 3:It describes planning of the project along with the capabilities,roles and responsibilities of members of the project.
- Chapter 4:In this chapter we generated a software requirement specification which includes the hardware and software requirement.
- Chapter 5:This chapter consists of system requirements definition.
- Chapter 6:This chapter consists of implementation of entire project, along with modular flow.It also consists of source-code of the project.
- Chapter 7: Here,system testing is done.Various scenarios are checked against different behaviours.
- Chapter 8: This chapter consists of screenshot of the working project.
- Chapter 9: This chapter provides conclusion as well as future scope of the project

Service By KRRC (Central Library)

Chapter 2

Literature Survey

2.1 Automated Menu Planning Algorithm For Children Using ID3 for Indian Food Database.

Adequate nutrition is essential in early childhood for the proper body growth and organ formation, to have a strong immune system, cognitive and neurological development. Children in India are mostly suffered from malnutrition. It happens because most of the mothers don't have proper knowledge about nutrition facts, which is to be feed to her child. To give proper diet to children as per their profile, Dietary Management System using ID3 is proposed. In this paper, ID3 is implemented with an example of Beverages using Weka tool and proposed work will be implemented in JAVA.

2.1.1 Advantages of Paper

a. Food items are assigned in a proper portion for Morning beverages breakfast,lunch,evening snacks,dinner

b. Food likability factor is considered.

2.1.2 Disadvantages of Paper

a. Only specific age group between 1-7 years is considered.

b. Only one classifier is used that is ID3

2.1.3 How to Overcome the Problems Mentioned in Paper

a. Age group between 15 to 30 is considered.

b. Different types of classifiers will be used.

Service By KRRC (Central Library)

2.2 A Dietary Menu-Generating System to Promote Healthy Life

In Japan, it has been pointed out that the number of patients afflicted with lifestyle diseases such as hypertension,diabetes and hyperlipemia is rapidly growing. Theses lifestyle diseases are caused from Habitual excessive drinking/eating, biased nutrition, little physical activity, sleep deprivation or stress. It is well-known that bad lifestyle habit provokes lifestyle diseases. In other words, improvement of lifestyle habit prevents these diseases from progressing.To improve eating habits (alimentary therapy), knowledge of medicine and nutrition is required . There are many studies which are focused on health care,however, none of them developed functions to assist planning balanced healthy meals. This paper describes a designing and implementation of alimentary therapy assist system. The system can record users' individual data, such as gender, age, height, weight and so on. Also, this system has a function to accumulate clinical history. And this system will record medical information. Furthermore, this system has a function to accumulate user's meal history and market information, and provide the user with healthy menu based on the meal history, individual information and market information.

2.2.1 Advantages of Paper

a. They have three main modules such as Detection module in order to detect any life threatening disease,Medical diagnosis module,Alimentary therapy support module.

b. Menu can be generated based on user's preference

2.2.2 Disadvantages of Paper

a. Accuracy of the model is not tested.

b. The user interface is not simple for elderly people to use easily.

2.2.3 How to Overcome the Problems Mentioned in Paper

a. Simple and easy user interface will be designed

b. Accuracy will be tested.

Service By KRRC (Central Library)

2.3 Nutrients Facts Analysis Using Supervised Learning Approaches.

A healthy lifestyle in people is achieved by Classification Algorithms in Data Mining – A Survey having balanced and nutritional food. In today's world we do not know with absolute certainty what foods can be consumed and what cannot be consumed, that is, we do not know for sure what foods have good nutritional value and what foods do not. The nutritional facts label is printed on food products all over the world and they are represented using a similar structure. These nutritional facts give data about some of the major nutrients present in the food product such as carbohydrates, protein and so on. These nutrition fact labels are not easily understood by common people. People who are careful about their diet such as those who exercise and diet regularly, trainers, and nutritionists may understand these nutritional facts, but not the common people. To make this information accessible in an easier way by classifying these food products into five levels of healthiness ranging from very healthy to very dangerous is the aim of this project work. This is done by a sequential process of data retrieval, data cleaning, data labeling and supervised learning.

2.3.1 Advantages of Paper

a. Accuracy testing is done

b. Food fact analysis is done using web scraping.

2.3.2 Disadvantages of Paper

a. Simple analysis of nutrient content is done.

b. No generation of diet plan

2.3.3 How to Overcome the Problems Mentioned in Paper

a. Specific diet plan will be generated according to individual's requirement.

Service By KRRC (Central Library)

2.4 Classification Algorithms in Data Mining – A Survey

Data Mining or Knowledge Discovery is the latest emerging trend in the information technology. It is the process of analyzing data from different perspectives and summarizing it into useful information. One of the function of data mining is classification, is a process of generalizing data sets based on different instances. There are various classification techniques which help as to group the data sets. Some the algorithms that this paper will be analyzing are Linear Regression, Multi Layer Perceptron, CART, J48, C4.5, ID3, Random forest and KNN.

2.4.1 Advantages of Paper

a. Detailed study of classification algorithm is done in this paper

b. Study of each algorithm along with their advantages and disadvantages is mentioned in this paper.

2.5 Technical Review

The technology that we are using here is python,along with django as it's web framework.

2.5.1 Scikit-learn

Sci-kit-learn is one of the most popular ML libraries for classical ML algorithms. It is built on top of two basic Python libraries, viz., NumPy and SciPy. Scikit-learn supports most of the supervised and unsupervised learning algorithms. Scikit-learn can also be used for data-mining and data-analysis, which makes it a great tool who is starting out with ML.

2.5.2 Advantages of Technology

a. It has a clean API, is robust, fast, easy to use,and comprehensive.

b. It is well documented and supported and is released under a permissive license and has a very active developers community.

c. It is usually the first choice in Machine Learning when implementing any project in Python.

2.5.3 Reasons to Use This Technology

a. It is simple,fast and scalable.

Service By KRRC (Central Library)

b. It is comparatively better than other libraries.

c. It has many algorithms which can be used for classification.

2.5.4 Pandas

Pandas is an open source, BSD-licensed library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language.

2.5.5 Advantages of Technology

a. It can easily represent data in a form naturally suited for data analysis via its Data Frame and Series data structures in a concise manner.

b. It provides for easy sub setting and filtering of data, procedures that are a staple of doing data analysis.

c. Its concise and clear API allows the user to focus more on the core goal at hand, rather than have to write a lot of scaffolding code in order to perform routine tasks. For example, reading a CSV file into a Data Frame data structure in memory takes two lines of code, while doing the same task in Java/C/C++ would require many more lines of code or calls to non-standard libraries

2.5.6 Reasons to use this Technology

a. Pandas is built upon the NumPy libraries and hence, inherits many of the performance benefits of this package, especially when it comes to numerical and scientific computing.

b. Fast and easy to use.

c. It is naturally suited to data analysis

‘

Service By KRRC (Central Library)

Chapter 3

Project Planning

3.1 Members and Capabilities

Table 3.1: Table of Capabilities

SR. No	Name of Member	Capabilities
1	Alekar Ifat Salim	UI Design
2	Malbari Sabiya Abdul Rashid	UI Design
3	Shaikh Mariya Irfan	Database,UI Design

3.2 Roles and Responsibilities

Table 3.2: Table of Responsibilities

SR. No	Name of Member	Role	Responsibilities
1	Alekar Ifat Salim	Team Leader	UI Design,Integration
2	Malbari Sabiya Abdul Rashid	Team Member	Back-end,Documentation
3	Shaikh Mariya Irfan	Team Member	UI design,database.

3.3 Assumptions and Constraints

Assumption:Assumption is that user will know about the pre existing diseases he or she is suffering from such as diabetes,heart disease,or kidney disease.Also,another assumption is that user likes a particular food which will be provided in a diet plan.

Constraints:If a model is not trained properly entire system might generate a wrong diet plan,thus accurate models must be built.

3.4 Project Management Approach

Prototyping Model is a system development method (SDM) in which a prototype (an early approximation of a final system or product) is built, tested, and then reworked

Service By KRRC (Central Library)

as necessary until an acceptable prototype is finally achieved from which the complete system or product can now be developed. This model works best in scenarios where not all of the project requirements are known in detail ahead of time. It is an iterative, trial-and-error process that takes place between the developers and the users. There are several steps in the Prototyping Model:

- The new system requirements are defined in as much detail as possible. This usually involves interviewing a number of users representing all the departments or aspects of the existing system.
- A preliminary design is created for the new system.
- A first prototype of the new system is constructed from the preliminary design. This is usually a scaled-down system, and represents an approximation of the characteristics of the final product.
- The users thoroughly evaluate the first prototype, noting its strengths and weaknesses, what needs to be added, and what should to be removed. The developer collects and analyzes the remarks from the users.
- The first prototype is modified, based on the comments supplied by the users, and a second prototype of the new system is constructed.
- The second prototype is evaluated in the same manner as was the first prototype.
- The preceding steps are iterated as many times as necessary, until the users are satisfied that the prototype represents the final product desired.
- The final system is constructed, based on the final prototype.

3.5 Ground Rules for the Project

Following ground rules have been considered:

1. We always pitch in to help where necessary to help solve problems and catch-up on behind schedule work.
2. Additional meetings can be scheduled to discuss critical issues.
3. All project team members have access to project plan and project logs (in a standard document format) and are aware of the assigned tasks and due dates.
4. All project members have the responsibility to proactively notify the project manager about tasks, duration or dependencies they believe are missing (or any other needed changes to the plan) and confront issues directly and promptly.
5. Constructive feedback of team leader or team member is valued.

Service By KRRC (Central Library)

6. All project team members understand the scope of Project. Any work performed must be in the project plan and is in the project scope.

3.6 Project Budget

Scikit-learn: Open Source
Pandas: Open Source
Django: Open Source

3.7 Project Timeline

Views Clipboard

	Name	Duration	Start	
1	Analysis	23 days	3/8/18 8:00 AM	4/9/
2	Feasibility study	4 days?	5/9/18 8:00 AM	10/9
3	Technology study	4 days?	11/9/18 8:00 AM	14/9
4	Financial feasibilty stu	2.5 days?	17/9/18 8:00 AM	19/9
5	Identifying market pot	5 days?	19/9/18 1:00 PM	26/9
6	Finalizing the Project	1.5 days?	26/9/18 1:00 PM	27/9
7	Design	5.5 days?	26/9/18 1:00 PM	3/10
8	High level Design	6 days?	4/10/18 8:00 AM	11/1
9	Low level Design	5 days?	28/9/18 8:00 AM	4/10
10	Coding	23 days?	5/10/18 8:00 AM	6/11
11	Front-End	22.75 da...	7/11/18 8:00 AM	7/12
12	Back-End	40.068 d...	7/12/18 3:00 PM	1/2/
13	Integration of Modules	3 days?	16/1/19 8:00 AM	18/1
14	Testing	9 days?	7/2/19 8:00 AM	19/2

Figure 3.1: Project Timeline

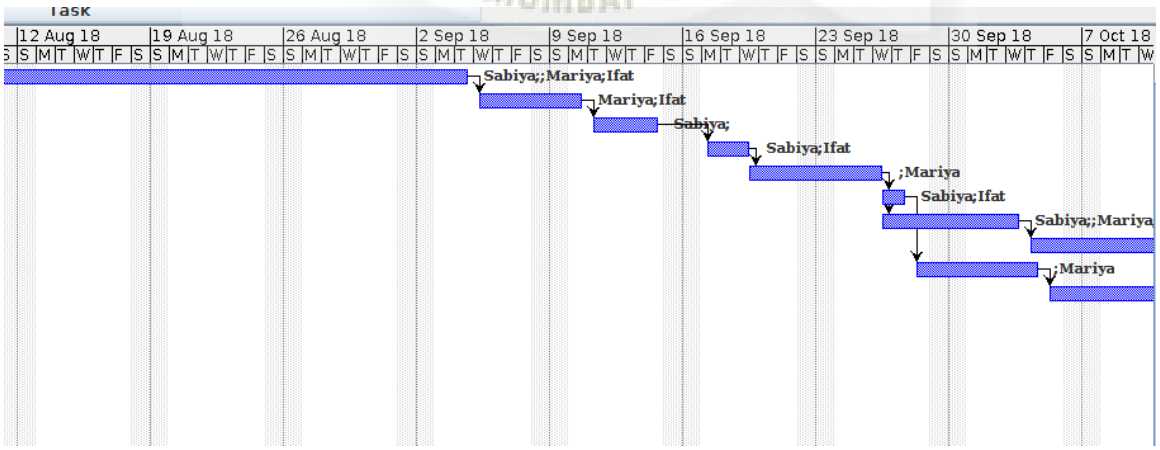

Figure 3.2: Project Timeline/Gantt chart

Service By KRRC (Central Library)

Chapter 4

Software Requirements Specification

4.1 Overall Description

4.1.1 Product Perspective

Automated Diet planner application consists of user perspective as well as admin's perspective.
User's or Customer's Perspective:

- User will be able to interact with the application over the internet.
- According to the inputs given by user,diet plan will be generated for them after the calculation of BMI as well as required calorie.

Admin's perspective:

- According to feedback given by user, admin can modify the diet as well as keep a track of user's weight and whether the diet is effective or not.

4.1.2 Product Features

The features of the project include:

- Diet plan will be provided using a web interface.
- User can calculate BMI as well as required calorie.
- Registered user can get a plan which matches according to their requirement.

4.1.3 User Classes and Characteristics

Users of the application includes individuals between 15 to 45 years.A particular individual will be able to generate a diet plan based on the inputs provided by them.Another user of the application will be admin who will keep a track of all the individuals regarding diet plans,whether it is beneficial for user or not.A particular user will interact with class of product such as Registration,assignment of diet plans and calculation of required calorie.

Service By KRRC (Central Library)

4.1.4 Operating Environment

User must have an access to web browser where the user can use this system on web efficiently.The environment in which system will operate is platform-independent.

4.1.5 Design and Implementation Constraints

Hardware Requirement:

- Minimum i3 processor
- 4gb Ram

Software Requirement:

- Python
- Django

Data pre-processing is a very crucial step as well as critical,because data needs to be very accurate in order to train it.problem might be encountered while training the model,because it might happen that their might be some missing values in it.The main decision was to select which language to use for front-end development.

4.2 System Features

The major feature of our system is to generate a diet plan based on parameters provided by user.

4.2.1 System Feature

- Generation of diet plan
- Sign up
- Calculation of required calorie along with BMI calculation.

Description and Priority

1. Calculation of required calorie: It is a feature having higher priority,It allows user to calculate the required calories.
2. Sign up: This feature allows new user to register themselves and become an authenticated user
3. Generation of Diet plan: this is an important module which will generate a diet plan based on the input provided by user.This feature has the highest priority

Service By KRRC (Central Library)

Stimulus/Response Sequences

1. The user needs to sign up into system.
2. The user will enter all the necessary details as required in the application.
3. Generation of diet plan

4.2.2 Functional Requirements

User Interface:

1. The software provides a good graphical interface for the user.
2. Any user can calculate the required calorie.
3. Generation of diet plan is done.

Hardware Interface:
Hard disk:50 GB
Ram:4GB
Processor:i5

Software Interface:
Back end:Python
Front end:Django
Spyder,Sublime text editor

4.3 External Interface Requirements

4.3.1 User Interfaces

The web server must provide a user interface that will be accessible through any web browser such as Google chrome,Internet Explorer,Mozilla firefox.

4.3.2 Hardware Interfaces

For hardware interface any system with 4GB ram and i5 or i3 processor is required

4.3.3 Software Interfaces

Operating System: Linux,Windows
Database:SQLite

Service By KRRC (Central Library)

4.3.4 Communications Interfaces

Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.

4.4 Nonfunctional Requirements

4.4.1 Performance Requirements

The system must be interactive and the delays must be less.Delays might ocur due to server error,since all the functions are done online.

4.4.2 Safety Requirements

The data that is stored in database is critical and must be secured.Since the data is accessed by admin and authorized user,security of data is ensured.

4.4.3 Security Requirements

The server on which Online data is stored will have it's own security to prevent unauthorized write/delete access.There is no restrictions on read access.

Service By KRRC (Central Library)

Chapter 5

System Design

5.1 System Requirements Definition

In this project we generated a diet plan which will help user to maintain a balanced weight along with healthy body.The system is built upon machine learning algorithm such as decision tree regresser, random forest and so on.Training and test data is provided to system for building a machine learning model.Individual will get a diet plan based on the parameters provided by them such as height,weight which will help in calculation of BMI as well as calorie calculation.The objective of requirement definition phase is to derive two types of requirement:
Model Training:It plays a significant role in our project.Decision tree algorithm has been used in order to train the model regarding the diet plan generation.
Data Pre processing:In this module various techniques such as handling missing values, development of proper diet plan has been done.
Integration:Integration of machine learning model along with the front-end is also the important module of this project.

5.1.1 Functional Requirements

- User must be able to sign up into the application.
- User must be able to generate a diet plan.
- User should be able to calculate a required calorie.
- The administrator should be able to view all the users who have generated a diet plan.

Service By KRRC (Central Library)

5.1.2 Use-case Diagram

A use case diagram is a dynamic or behavior diagram in UML. Use case diagrams model the functionality of a system using actors and use cases. Use cases are a set of actions, services, and functions that the system needs to perform. In this context, a ”system” is something being developed or operated, such as a web site. The ”actors” are user of the system.User will provide enter details such as Age,weight,height,Diseases and activity level.

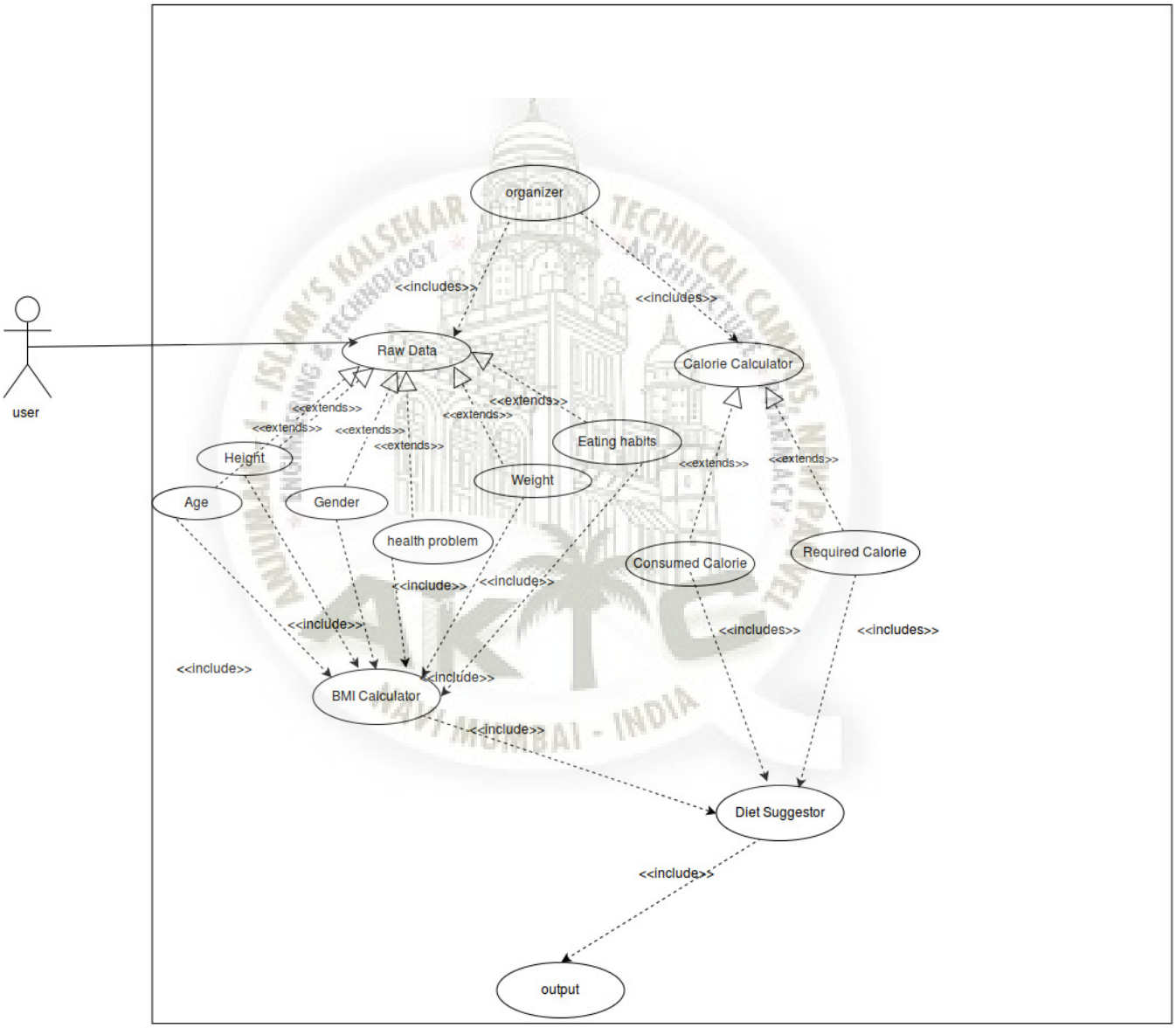

Figure 5.1: Use-case Diagram for Diet Recommendation system

Service By KRRC (Central Library)

Data-flow Diagram

A data-flow diagram (DFD) is a way of representing a flow of a data of a process or a system (usually an information system).Level 0 DFD consists of main automated diet planner as well as users and diet suggestor.Level 1 DFD consists of bmi,calorie calculation, which will be provided to diet planner. The refined representation of a process can be done in another data-flow diagram, which subdivides this process into sub-processes such as diet plan,exercise schedule.

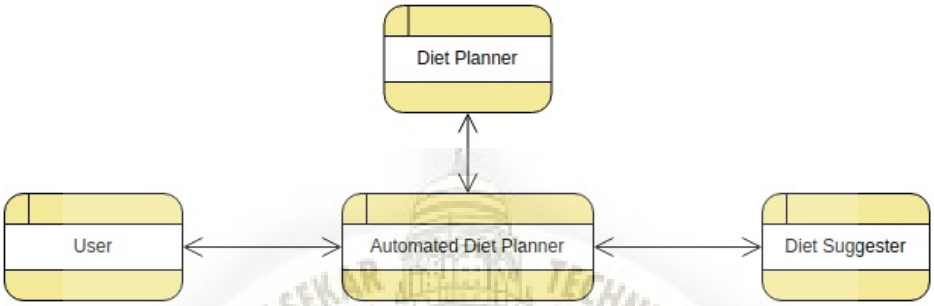

Figure 5.2: DFD-level 0

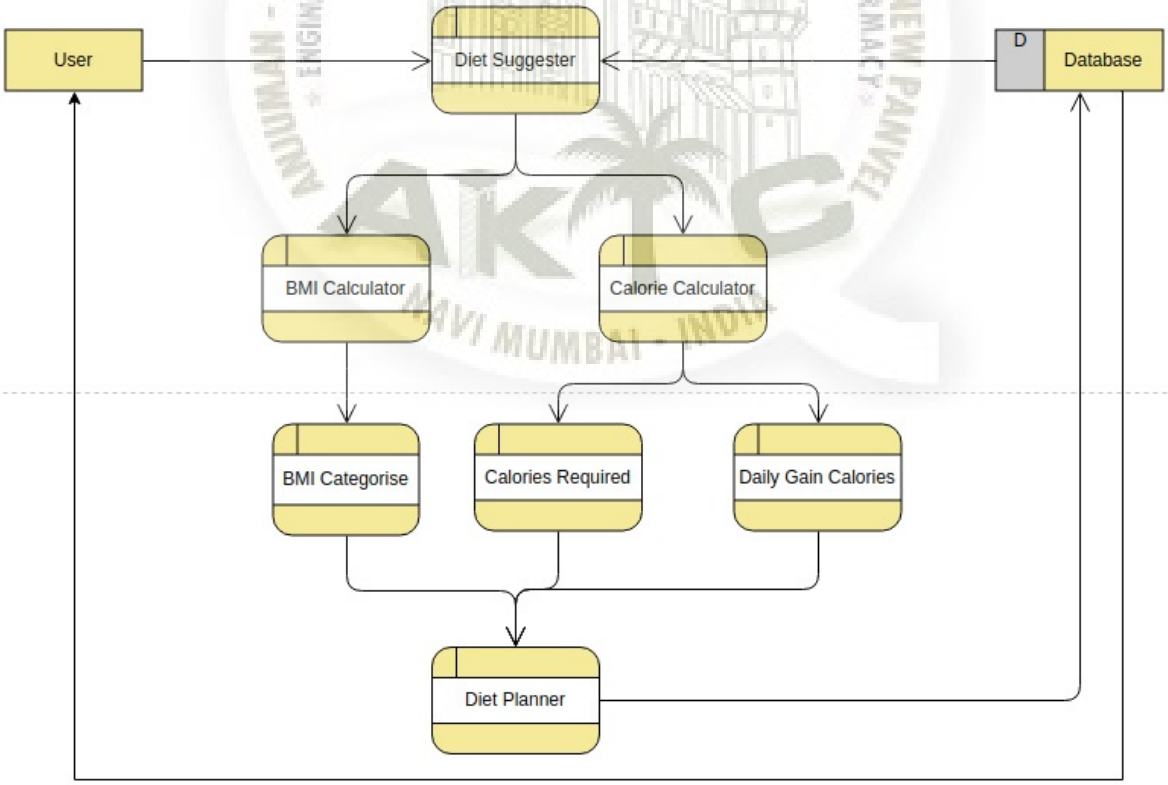

Figure 5.3: DFD -level 1

Service By KRRC (Central Library)

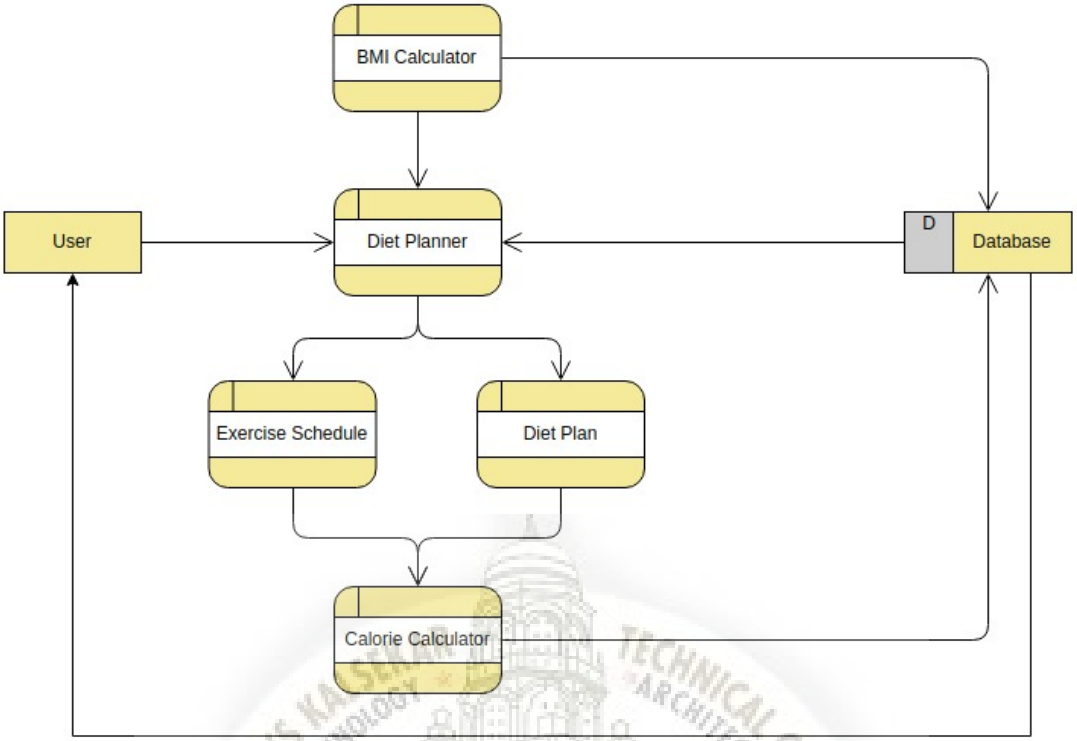

Figure 5.4: DFD-level 2

Service By KRRC (Central Library)

5.1.3 System Requirements (Non-Functional Requirements)

These are non-functional system properties such as availability, performance and safety etc. They define functions of a system, services and operational constraints in detail.

Performance Requirements: The system must be interactive and the delays must be less.Delays might occur due to server error,since all the functions are done online.

Safety Requirements: The data that is stored in database is critical and must be secured.Since the data is accessed by admin and authorized user,security of data is ensured.

Security Requirements: The server on which Online data is stored will have it's own security to prevent unauthorized write/delete access.There is no restrictions on read access.

Service By KRRC (Central Library)

Database Schema/ E-R Diagram

The ER or (Entity Relational Model) is a high-level conceptual data model diagram. Entity-Relation model is based on the notion of real-world entities and the relationship between them. ER diagram consists of various entities such as User,BMI,Calorie required and Food.

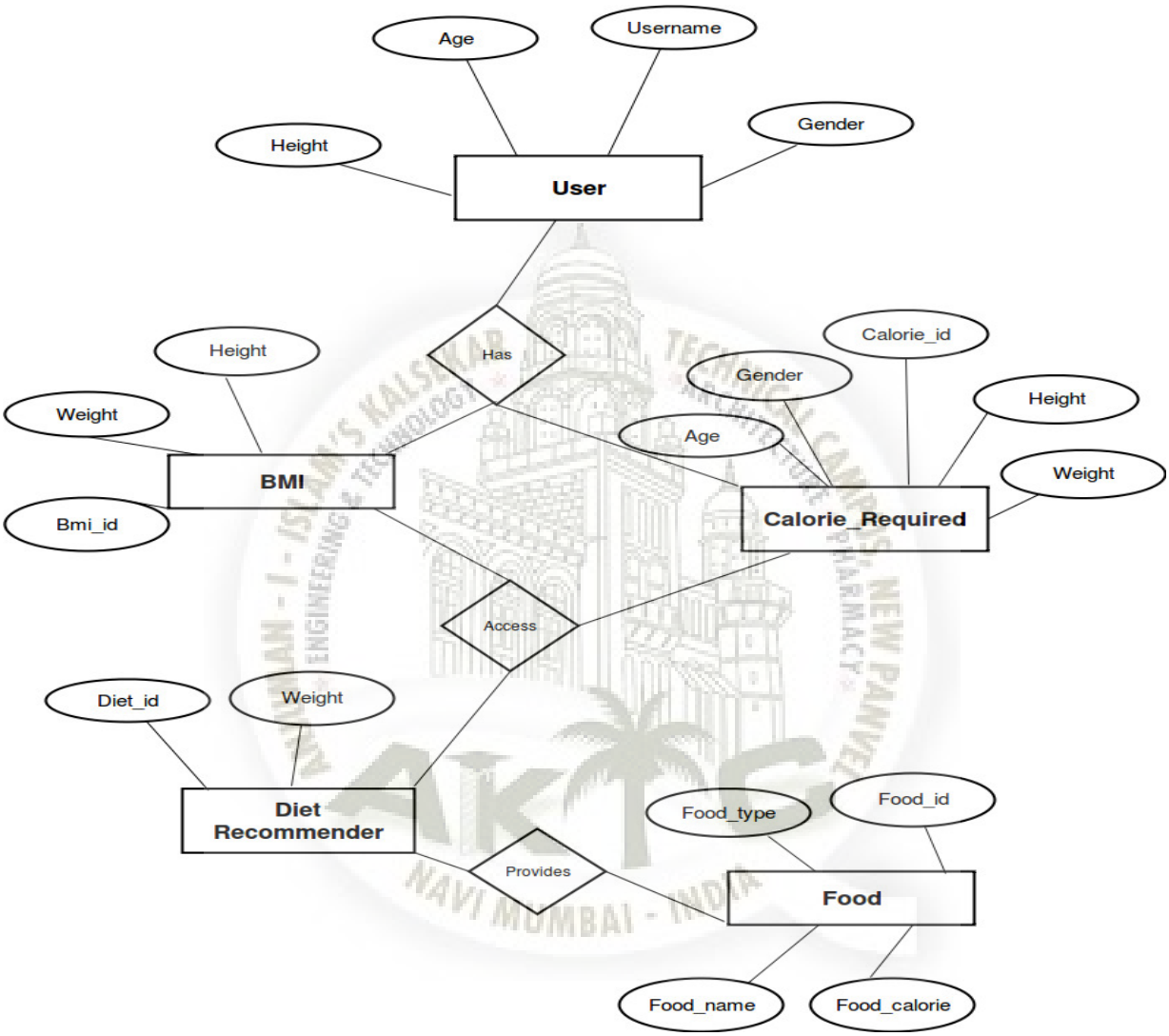

Figure 5.5: ER diagram

Service By KRRC (Central Library)

5.2 System Architecture Design

A system architecture or systems architecture is the conceptual model that defines the structure, behavior, and more views of a system.An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system.

A system architecture can consist of system components and the sub-systems developed, that will work together to implement the overall system.This system architecture consists of various modules such as calorie calculation module,diet planner as well as diet suggestor.

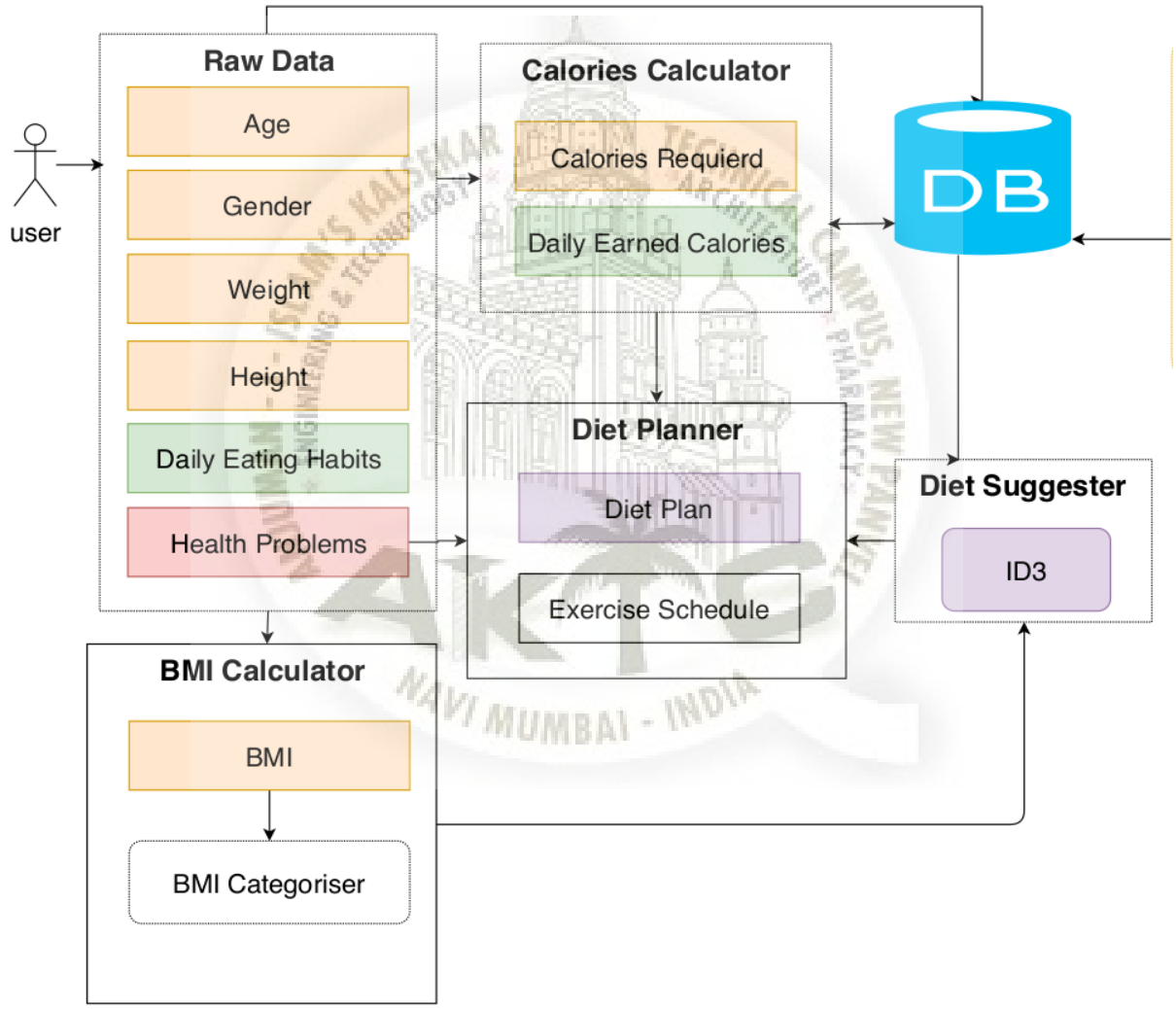

Figure 5.6: System Architecture

Service By KRRC (Central Library)

5.3 Sub-system Development

A modular description provides detailed information about the module and it's supported components which is accessible in different manner.Following are the modules of our system.

5.3.1 Bmi Calculation Module

In this module Body mass index will be calculated.Apart from that the categorization of BMI such as overweight,underweight,normal weight will be done.

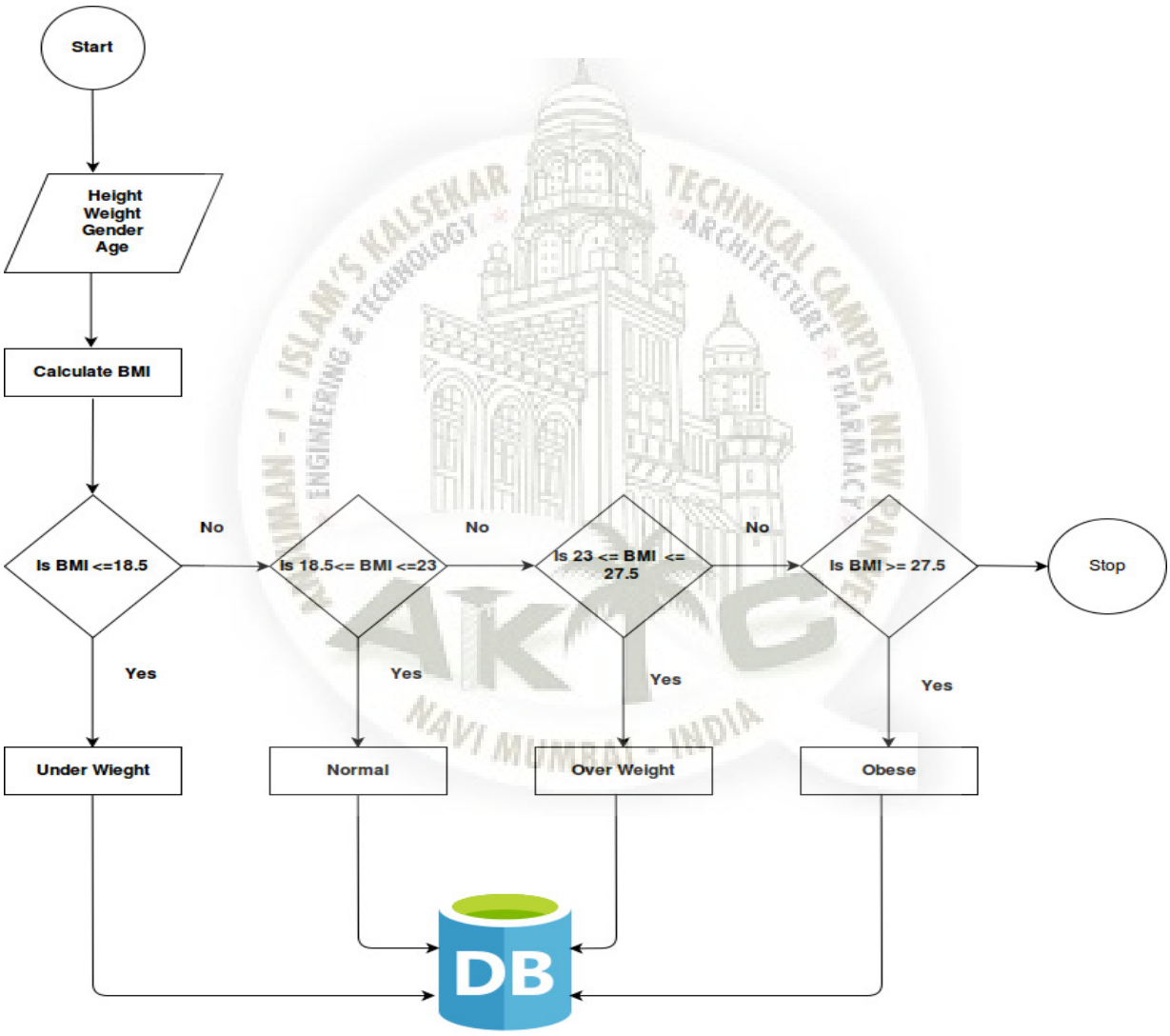

Figure 5.7: Module 1: BMI Calculation

Service By KRRC (Central Library)

5.3.2 Calorie Calculation

In this module calorie calculation will be done, using inputs such as height,weight and age. a formula will be applied which will calculate required calorie of an individuals.

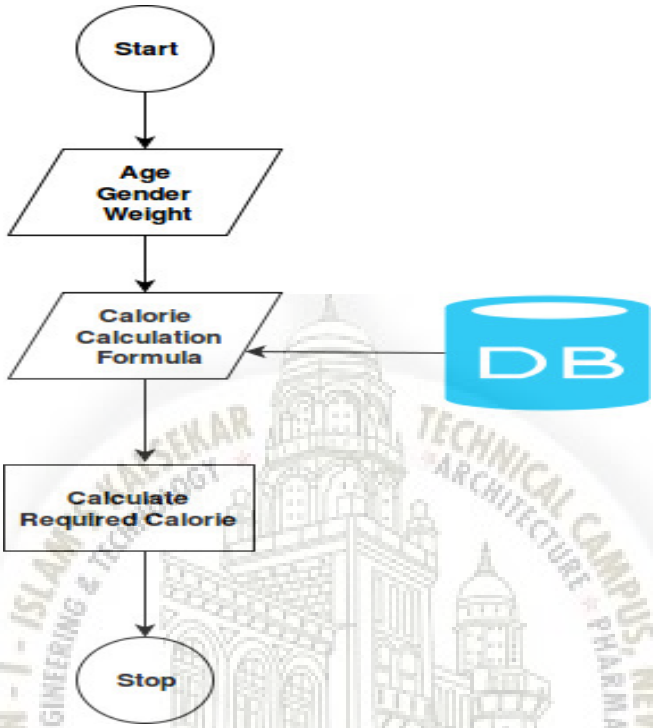

Figure 5.8: Module 2:Calorie Calculation

5.3.3 Diet generation module.

This is the main module of our system in which diet plan will be generated.It includes following steps such as data pre processing,data training,testing and then generation of diet plan.

Service By KRRC (Central Library)

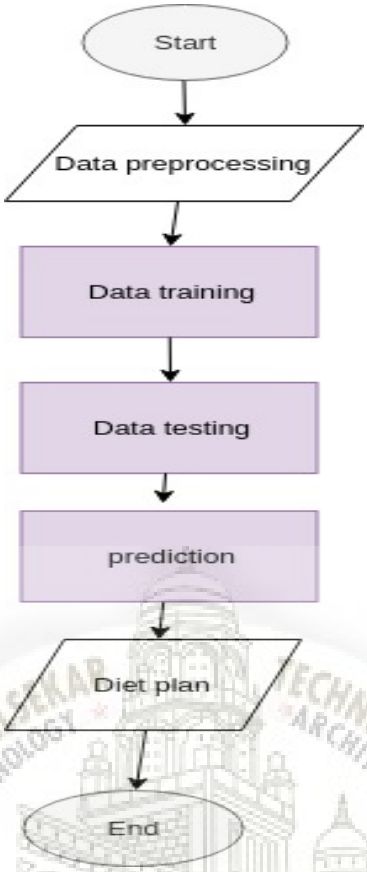

Figure 5.9: Module 3:Diet generation

5.4 System Integration

5.4.1 Class Diagram

The class diagram is the main building block of object-oriented modeling.It is used for general conceptual modeling of the structure of the application and for detailed modeling translating the model into programming code.Class Diagram consists of attributes such as User,Calorie calculator,BMI,Calorie Calculator

Service By KRRC (Central Library)

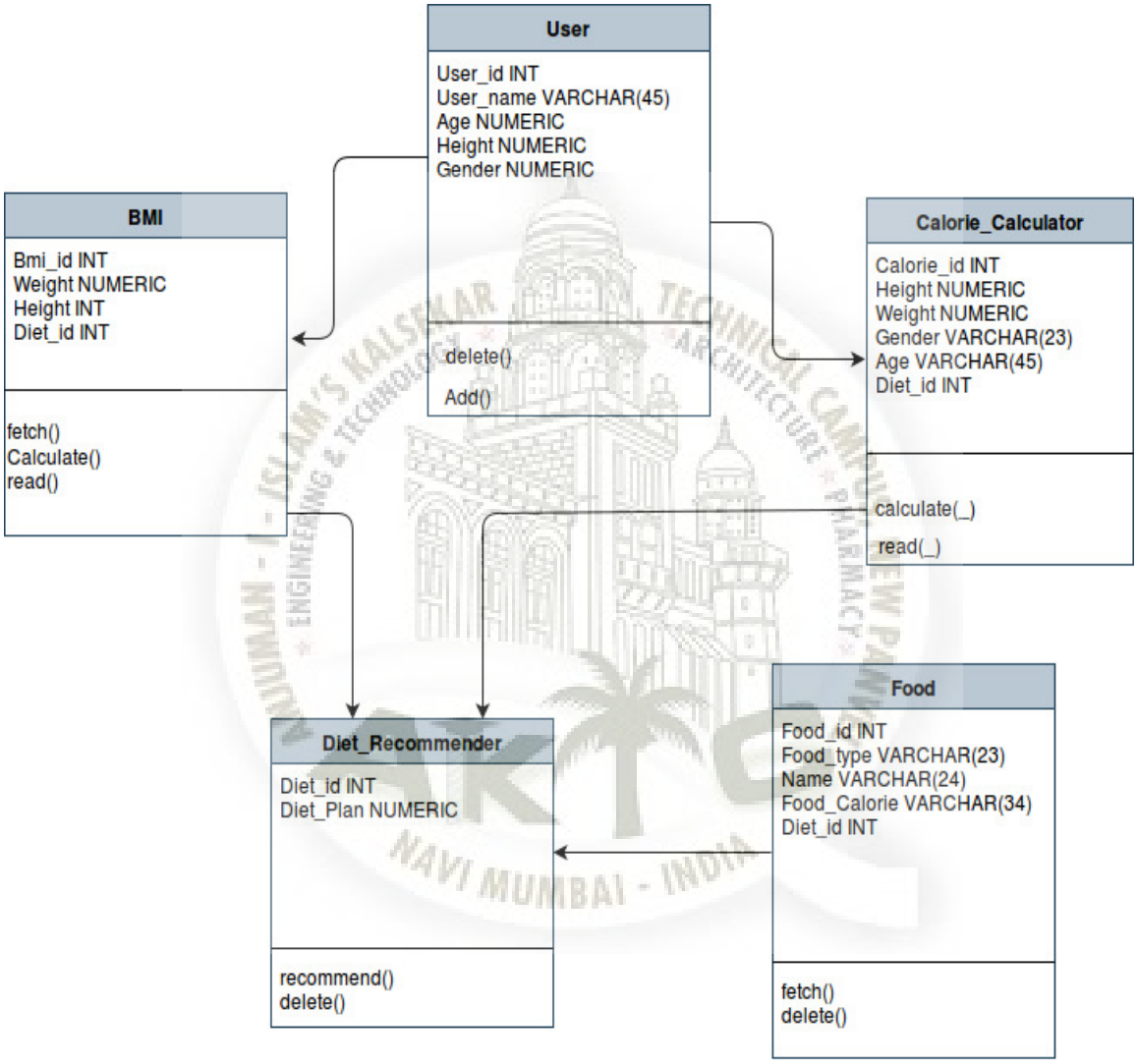

Figure 5.10: Class diagram

Service By KRRC (Central Library)

5.4.2 Sequence Diagram

A sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function.User will provide input to the application which will then validate it from server.In this way the flow of information is carried out in sequential manner.

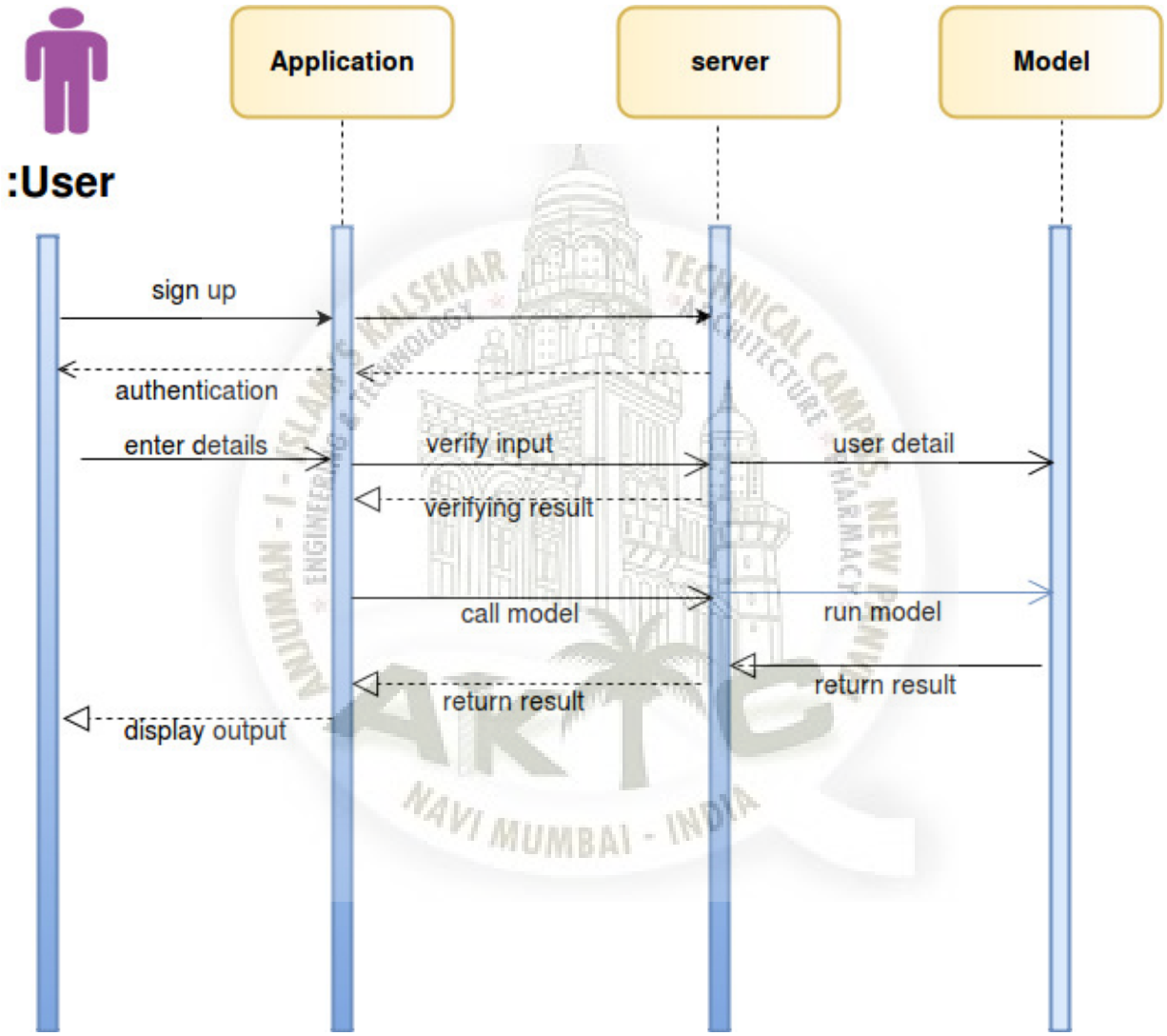

Figure 5.11: Sequence diagram

Service By KRRC (Central Library)

5.4.3 Component Diagram

Component diagrams are different in terms of nature and behavior. Component diagrams are used to model the physical aspects of a system. Now the question is, what are these physical aspects? Physical aspects are the elements such as executables, libraries, files, documents, etc. which reside in a node.component diagrams are used to visualize interactions between various components such as bmicalulator.py,plan.py,train.py

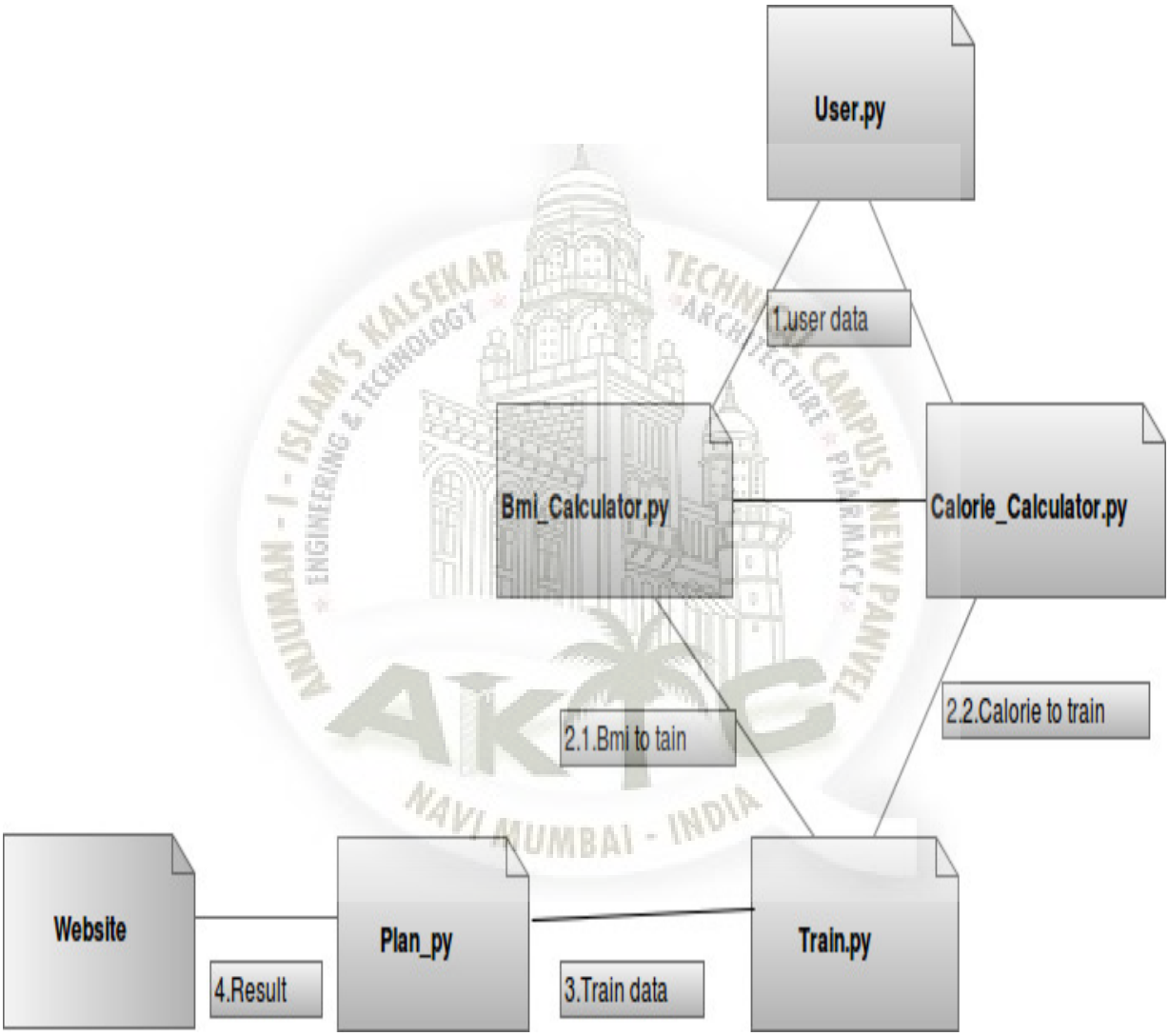

Figure 5.12: Component diagram

Service By KRRC (Central Library)

5.4.4 Deployment Diagram

A deployment diagram consists of Web server,Database server,Client.It shows the execution architecture of a system, including nodes such as hardware or software execution environments, and the middle ware connecting them. Deployment diagrams are typically used to visualize the physical hardware and software of a system. Deployment diagrams help model the hardware topology of a system compared to other UML diagram types which mostly outline the logical components of a system.

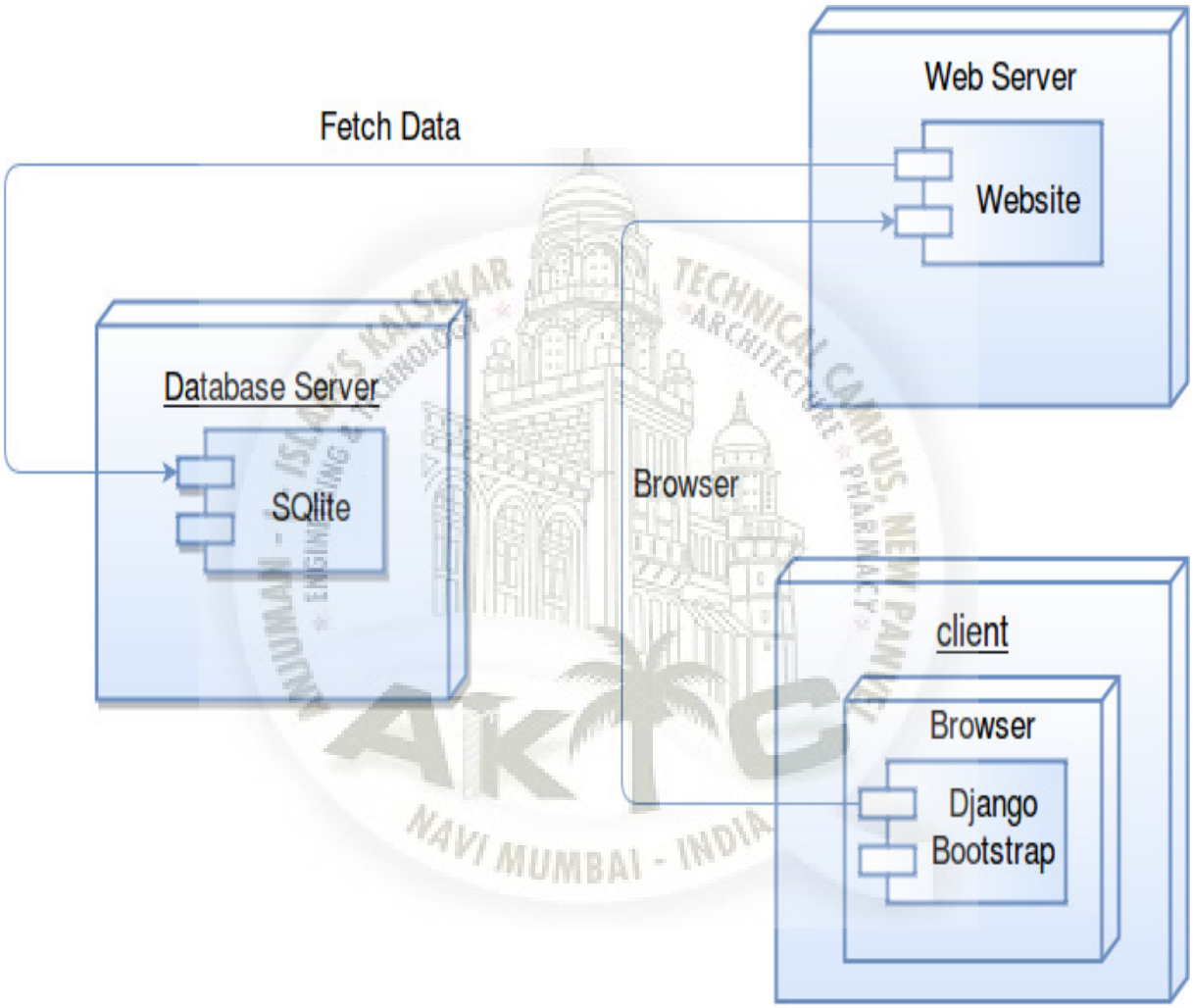

Figure 5.13: Deployment diagram

Service By KRRC (Central Library)

Chapter 6

Implementation

6.1 BMI calculation

```
def bmi(request):
    if request.method=="POST":
        height=request.POST['height']
        height=float(height)
        weight=request.POST['weight']
        weight=int(weight)
        pip=pow(height,height)
        BMI = weight / pip
        inss=BodyMassIndex(BMI=BMI)
        inss.save()

        if BMI <= 18.5:
            return render(request, 'diets/bmi.html', {'xyz':'Your BMI is', 'bmi'
                :f'{BMI:.4}', 'abc':'And  You are Under weight'})
        elif 18.5 < BMI <= 25 :
            return render(request, 'diets/bmi.html', {'xyz':'Your BMI is', 'bmi'
                :f'{BMI:.4}', 'abc':'And You are Normal Weight'})
        elif 25 < BMI <= 29.9 :
            return render(request, 'diets/bmi.html', {'xyz':'Your BMI is', 'bmi'
                :f'{BMI:.4}', 'abc':'And  You are Over Weight'})
        else : BMI >=30
        return render(request, 'diets/bmi.html', {'xyz':'Your BMI is', 'bmi':f'{
            BMI:.4}', 'abc':'And You are Obese Weight'})
        return render(request, 'diets/bmi.html', {'bmi':BMI})



    else:
        return render(request, 'diets/bmi.html')
```


Service By KRRC (Central Library)

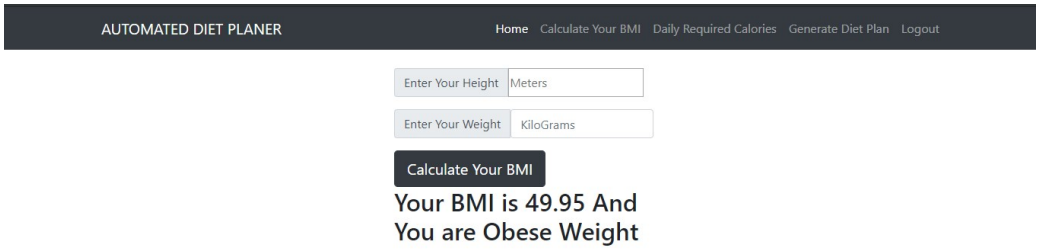

Figure 6.1: BMI Calculation.

Service By KRRC (Central Library)

6.2 Calorie calculation module

```
def calorie(request):
    if request.method == "POST":
        activity = request.POST["activity"]
        print("heeeelelxcvkxdvjfuk")
        #print(get_type("activity"))
        weight = request.POST['weight']
        weight = int(weight)
        abc = weight * 22
        if activity == "Sedentary":
            defg = abc * 1.6
        elif activity == "Lightly Active":
            defg = abc * 1.8
        elif activity == "Moderately Active":
            defg = abc * 2.0
        else:
            defg = abc * 2.2
        inx = Calories(defg=defg)
        inx.save()
        return render(request, 'diets/calorie.html', {'xyz':'Your Daily Required
            Calories Is = ','calories':round(defg)})
        '''return render(request, 'diets/calorie.html', {'xyz':'Your Daily
           Required Calories Is = ', 'calories':f'{defg:.2}'})'''
    return render(request, 'diets/calorie.html')
```

Figure 6.2: Calorie Calculation

Service By KRRC (Central Library)

6.3 Prediction module

```
#!/usr/bin/env python2
# -*- coding: utf-8 -*-
"""Created on Thu Mar 28 16:25:46 2019
@author: sabiya
"""
import pandas as pd
from sklearn.tree import DecisionTreeRegressor
from sklearn.model_selection import train_test_split
#from sklearn.preprocessing import OneHotEncoder
from sklearn.preprocessing import LabelEncoder
from sklearn import metrics
#from numpy import argmax
import numpy as np
#import sklearn.cross_validation as cross_validation
print('k')
diet_file_path = '/home/sabiya/Desktop/project/train1.csv'
diet_data = pd.read_csv(diet_file_path)
#print(diet_data)
y=diet_data.iloc[:,12]
#print(y)
x = diet_data.iloc[:,[-8,-9,-10,-2]]
#print(x)
#x=diet_data.iloc[:,[3,4,5,11]]
print(type(diet_data))
print(type(x))
print(type(y))
#ohc1=pd.get_dummies(val_x)
#print(x)
new_diet_data = pd.read_csv('/home/sabiya/Documents/test.csv')
nx=new_diet_data.iloc[:,4]
ny=new_diet_data.iloc[:,[0,1,2,3]]
train_x,val_x,train_y,val_y=train_test_split(ny,nx,test_size=0.50,random_state =
    0)
#train_x,train_y,val_x,val_y=cross_validation.train_test_split(x, y, train_size
    =0.75, random_state=101)
'''ohc=pd.get_dummies(train_x)
ohc2=pd.get_dummies(val_x)
ohc3=pd.get_dummies(train_y)
ohc4=pd.get_dummies(val_y)'''
#print(ohc2)
print("=========================transformation=========================")
le=LabelEncoder()
'''================================kidney_disease================='''
kd=diet_data.iloc[:,-10]
nkd=np.array(kd)
print(type(nkd))
le.fit(nkd)
#list(le.classes)
tkd=le.transform(nkd)

pd.DataFrame(tkd).to_csv("/home/sabiya/Desktop/project/formatted.csv",index =
    False)
print(type(tkd))
print(tkd)
reversetkd=le.inverse_transform(tkd)
'''================================kidney_disease================='''
#print(reversex)
'''================================heart_disease================='''
hd=diet_data.iloc[:,-8]
```


Service By KRRC (Central Library)

```
nhd=np.array(hd)
le.fit(nhd)
#list(le.classes)
thd=le.transform(nhd)
pd.DataFrame(thd).to_csv("/home/sabiya/Desktop/project/formatted2.csv",index =
    False)
print(thd)
reversethd=le.inverse_transform(thd)
'''================================heart_disease================='''
'''================================diabetes================='''
dia=diet_data.iloc[:,-9]
ndia=np.array(dia)
le.fit(ndia)
#list(le.classes)
tdia=le.transform(ndia)
pd.DataFrame(tdia).to_csv("/home/sabiya/Desktop/project/formatted3.csv",index =
    False)
print(tdia)
reversetdia=le.inverse_transform(tdia)
'''================================diabetes================='''

print("=========================transformation complete
    !=========================")
#dec=enc.dot(OHC.active_features_).astype(int)
#print(dec)
#print(ohc1)
#print(val_x)
my_model=DecisionTreeRegressor(random_state=1)
my_model.fit(train_x,train_y)
#print("The predictions are")
#pred=my_model.predict(val_x)
pred=my_model.predict([[1,0,1,3600]])
gt = np.array(train_y)
pqr=print(pred)
#print(pred)
fpr,tpr,threshold = metrics.roc_curve(gt,pred,pos_label=2)
acc = metrics.auc(fpr, tpr)
print(acc)
#print(pred)
#b=pred[:,-1]
#print(b)
#rs=argmax(pred)
#print(rs)
#x=pred[-1,:]
#print(x)
#print(type(pred))
#rs=pd.get_dummies(pred).idmax(1)
#rs=np.reshape(pred,(1,1))
#fpr,tpr,thresholds=metrics.roc_auc_score(train_y,pred)
#print(pred)
#metrics.auc(fpr,tpr)
#print(my_model.predict(ohc2.head(1)))
```

6.4 Integration module

```
def input(request):
    if request.method == "POST":
```


Service By KRRC (Central Library)

```
        users= request.user.username
        age = request.POST["age"]
        height = request.POST["height"]
        gender= request.POST["gender"]
        activity= request.POST["activity"]
        diabetes= request.POST["diabetes"]
        heart= request.POST["heart"]
        kidney= request.POST["kidney"]

        weight = request.POST ["weight"]
        ins = Diet(users=users,age=age,height=height,gender=gender,activity=
            activity,weight=weight,diabetes=diabetes,heart=heart,kidney=kidney)
        ins.save()
        abc= weight * 22
        defg = abc * 2


        import pandas as pd
        from sklearn.tree import DecisionTreeRegressor
        from sklearn.model_selection import train_test_split
        #from sklearn.preprocessing import OneHotEncoder
        from sklearn.preprocessing import LabelEncoder
        from sklearn import metrics

        #from numpy import argmax
        import numpy as np
        #import sklearn.cross_validation as cross_validation
        #print('k')
        diet_file_path = '/home/sabiya/Desktop/diet/diets/train1.csv'
        diet_data = pd.read_csv(diet_file_path)
        #print(diet_data)
        y=diet_data.iloc[:,12]
        #print(y)
        x = diet_data.iloc[:,[-8,-9,-10,-2]]
        #print(x)
        #x=diet_data.iloc[:,[3,4,5,11]]
        #print(type(diet_data))
        #print(type(x))
        #print(type(y))
        #ohc1=pd.get_dummies(val_x)
        #print(x)
        new_diet_data = pd.read_csv('/home/sabiya/Desktop/diet/diets/test.csv')
        nx=new_diet_data.iloc[:,4]
        ny=new_diet_data.iloc[:,[0,1,2,3]]
        train_x,val_x,train_y,val_y=train_test_split(ny,nx,test_size=0.50,
            random_state = 0)
        '''ohc=pd.get_dummies(train_x)
        ohc2=pd.get_dummies(val_x)
        ohc3=pd.get_dummies(train_y)
        ohc4=pd.get_dummies(val_y)'''



        #print("==========================transformation
            ==========================")

        le=LabelEncoder()
        '''================================kidney_disease================='''
        kd=diet_data.iloc[:,-10]
        nkd=np.array(kd)
```

```
        #print(type(nkd))
        le.fit(nkd)
        tkd=le.transform(nkd)

        pd.DataFrame(tkd).to_csv("/home/sabiya/Desktop/diet/diets/formatted.csv"
            ,index = False)
        #print(type(tkd))
        #print(tkd)
        reversetkd=le.inverse_transform(tkd)
        '''================================kidney_disease================='''
        #print(reversex)
        '''================================heart_disease================='''
        hd=diet_data.iloc[:,-8]
        nhd=np.array(hd)
        le.fit(nhd)
        #list(le.classes)
        thd=le.transform(nhd)
        pd.DataFrame(thd).to_csv("/home/sabiya/Desktop/diet/diets/formatted2.csv
            ",index = False)
        #print(thd)
        reversethd=le.inverse_transform(thd)
        '''================================heart_disease================='''
        '''================================diabetes================='''
        dia=diet_data.iloc[:,-9]
        ndia=np.array(dia)
        le.fit(ndia)
        #list(le.classes)
        tdia=le.transform(ndia)
        pd.DataFrame(tdia).to_csv("/home/sabiya/Desktop/diet/diets/formatted3.
            csv",index = False)
        #print(tdia)
        reversetdia=le.inverse_transform(tdia)
        '''================================diabetes================='''

        print("==========================transformation complete
            !=========================")
        #dec=enc.dot(OHC.active_features_).astype(int)
        #print(dec)
        #print(ohc1)
        #print(val_x)
        my_model=DecisionTreeRegressor(random_state=1)
        my_model.fit(train_x,train_y)

        #print("The predictions are")
        #pred=my_model.predict(val_x)

        if diabetes=="Yes":
            diabetesz=1
        else:
            diabetesz=0

        if heart=="Yes":
            heartz=1
        else:
            heartz=0

        if kidney=="Yes":
            kidneyz=1
        else:
            kidneyz=0
```

```
        weight = int(weight)
        abc = weight * 22
        if activity == "Senedarty":
            defg = abc * 1.6
        elif activity == "Lightly Active":
            defg = abc * 1.8
        elif activity == "Moderately Active":
            defg = abc * 2.0
        else:
            defg = abc * 2.2


        #pred=my_model.predict([[1,0,1,3600]])
        pred=my_model.predict([[kidneyz,heartz,diabetesz,defg]])
        gt = np.array(train_y)
        print(pred)
        #print(pred)

        #fpr,tpr,threshold = metrics.roc_curve(gt,pred,pos_label=2)
        #acc = metrics.auc(fpr, tpr)
        #print(acc)

        if pred==1:
            ghgh
        elif pred==2:
            list[]
        elif pred==:
            bhhh
        elif pred==2:

        elif pred==2:

        elif pred==2:

        elif pred==2:

        elif pred==2:

        elif pred==2:


        return render(request, 'diets/input.html', {'input':defg,'users':users})
    else:
        return render(request, 'diets/input.html')


    return render(request, 'diets/input.html')


def adminpanel(request):
    return render(request,'diets/admin_base.html')



def bmi(request):
    if request.method=="POST":
        height=request.POST['height']
        height=float(height)
        weight=request.POST['weight']
        weight=int(weight)
        pip=pow(height,height)
```


Service By KRRC (Central Library)

```
        BMI = weight / pip
        inss=BodyMassIndex(BMI=BMI)
        inss.save()

        if BMI <= 18.5:
            return render(request, 'diets/bmi.html', {'xyz':'Your BMI is', 'bmi':f'{BMI:.4}', 'abc':'And  You are Under weight'})
        elif 18.5 < BMI <= 25 :
            return render(request, 'diets/bmi.html', {'xyz':'Your BMI is', 'bmi':f'{BMI:.4}', 'abc':'And You are Normal Weight'})
        elif 25 < BMI <= 29.9 :
            return render(request, 'diets/bmi.html', {'xyz':'Your BMI is', 'bmi':f'{BMI:.4}', 'abc':'And  You are Over Weight'})
        else : BMI >=30
        return render(request, 'diets/bmi.html', {'xyz':'Your BMI is', 'bmi':f'{BMI:.4}', 'abc':'And You are Obese Weight'})
        return render(request, 'diets/bmi.html', {'bmi':BMI})



    else:
        return render(request, 'diets/bmi.html')

def calorie(request):
    if request.method == "POST":
        activity = request.POST["activity"]
        print("heeeelelxcvkxdvjfuk")
        #print(get_type("activity"))
        weight = request.POST['weight']
        weight = int(weight)
        abc = weight * 22
        if activity == "Senedarty":
            defg = abc * 1.6
        elif activity == "Lightly Active":
            defg = abc * 1.8
        elif activity == "Moderately Active":
            defg = abc * 2.0
        else:
            defg = abc * 2.2

        inx = Calories(defg=defg)
        inx.save()
        return render(request, 'diets/calorie.html', {'xyz':'Your Daily Required Calories Is = ','calories':round(defg)})
        '''return render(request, 'diets/calorie.html', {'xyz':'Your Daily Required Calories Is = ', 'calories':f'{defg:.2}')'''
    return render(request, 'diets/calorie.html')

def detail(request):
    dobj = Diet.objects.all()
    #print(dobj)
    dobjs = BodyMassIndex.objects.all()
    dobjss = Calories.objects.all()
    lll = zip(dobj,dobjs,dobjss)
    print(dobj)
    print(dobjs)
    print(dobjss)
    print("heloooooooooooooooooooooooooooo")
    #print(lll)
    #for dobj,dobjs,dobjss in lll:
        #print(dobj,dobjs,dobjss)
```


Service By KRRC (Central Library)

```
234
235     return render(request,'diets/detail.html',{'lll':lll})
236
237
238
239 def your_view_name(request):
240   if request.method == 'GET':
241     form = your_form_name()
242   else:
243     if form.is_valid():
244       info = request.POST['info_name']
245       output = script_function(info)
246       #Here you are calling script_function,
247       #passing the POST data for 'info' to it;
248       return render(request, 'diets/input.html', {'info': info, 'output': output
              , })
249   return render(request, 'diets/input.html', {'form': form})
250
251
252 def script_function( post_from_form ) :
253   post_from_form
254   return subprocess.check_call(['/home/sabiya/Desktop/project/train.py',
        post_from_form])
255
256
257
258
259 def table(request):
260     return render(request, 'diets/table.html')
```


Service By KRRC (Central Library)

Chapter 7

System Testing

.

7.1 Test Cases and Test Results

Test ID	Test Case Title	Test Condition	System Behavior	Expected Result
T01	Login	User should be registered	User should login into system	User must be able to register s well as login into system
T02	Calorie calculation	Information entered must be real	Calorie required should be calculated	Calorie required should be calculated
T03	Generation of diet plan	User must know about the pre existing disease	generation of diet plan	Generation of diet plan

7.2 Sample of a Test Case

Title: Login Page – Authenticate Successfully on our application
Description: A registered user should be able to successfully login into our system. *Precondition:* the user must already be registered with an email address and password. *Assumption:* a supported browser is being used.

Test Steps:

1. Navigate to Signup
2. Register on our system.

Service By KRRC (Central Library)

3. Fill the necessary details.
4. Enter the password of the registered user
5. Click ‘Sign In’

Expected Result: Thereafter,user must be able to login into our application and must be redirected to home page.

Actual Result: Thereafter,user is able to login into our application and can redirected to home page.

Figure 7.1: Diet Plan

7.2.1 Software Quality Attributes

1. Availability:The system should not be down,it must be available as and when required.
2. Correctness:Correct output must be generated

Service By KRRC (Central Library)

Chapter 8

Screenshots of Project

8.1 Home page

Figure 8.1: Home Page

Service By KRRC (Central Library)

8.2 Bmi calculation

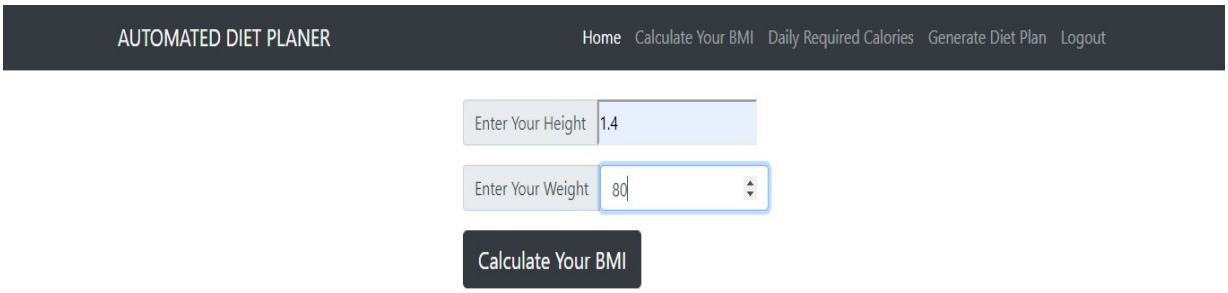

Figure 8.2: Calculation of BMI

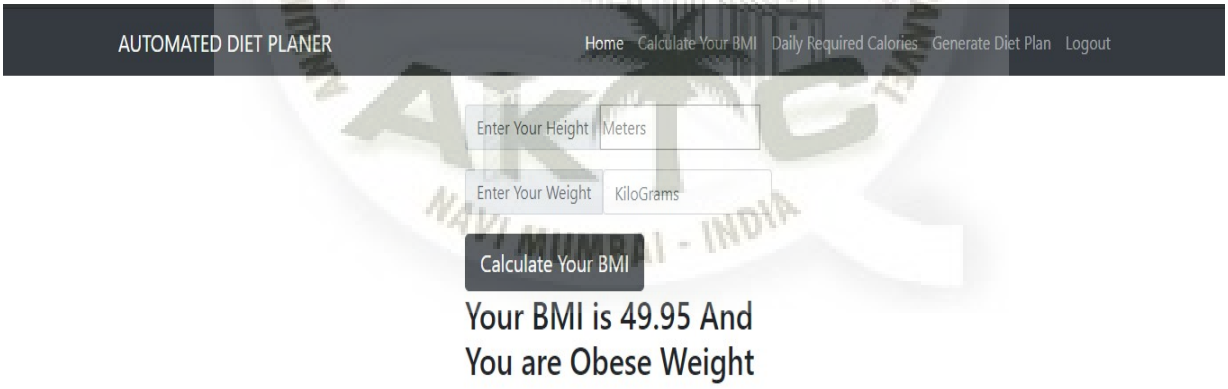

Figure 8.3: BMI Categorization

Service By KRRC (Central Library)

8.3 Calorie calculation

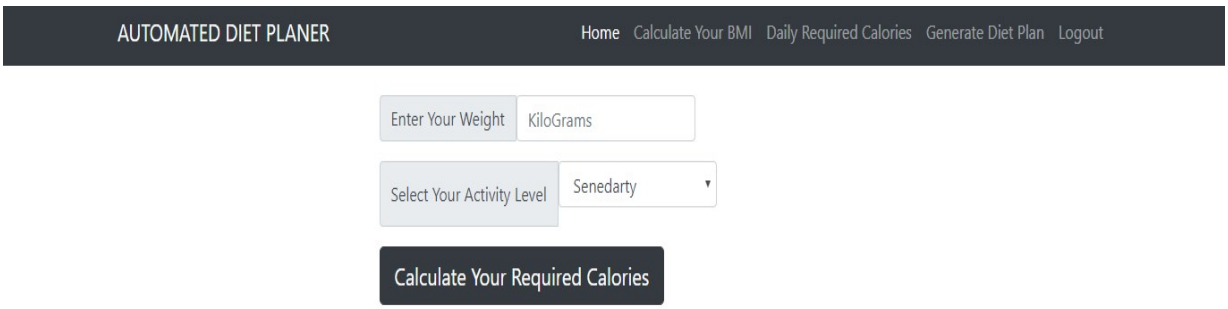

Figure 8.4: Calculation of Calorie

Service By KRRC (Central Library)

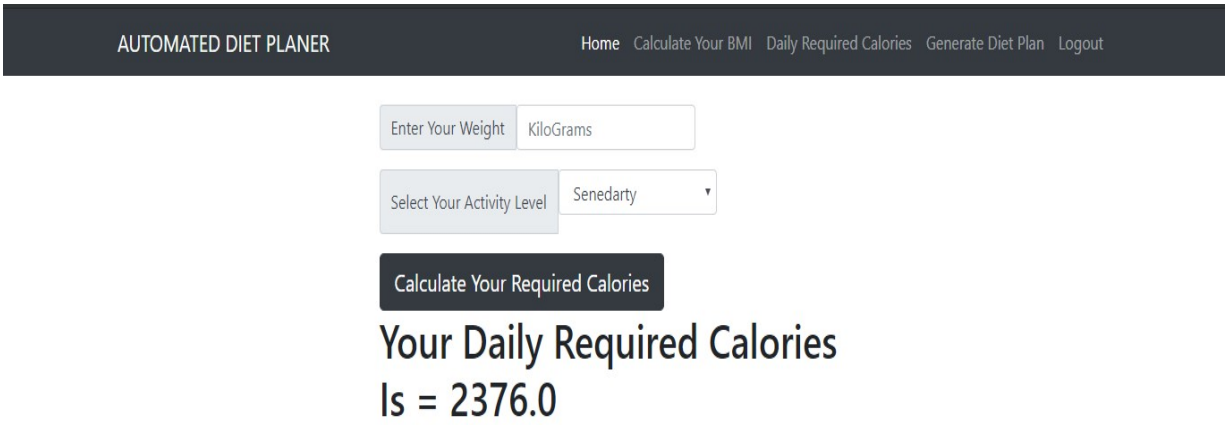

Figure 8.5: Calorie Categorization

8.4 Diet generation

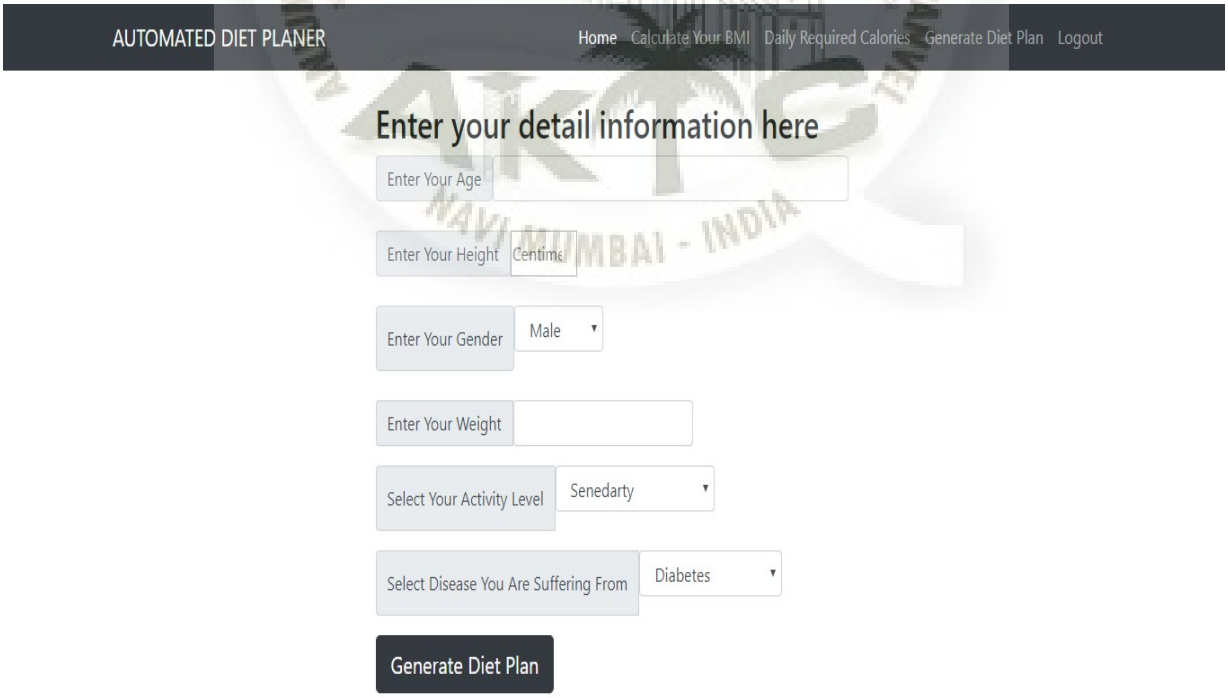

Figure 8.6: User Inputs

Service By KRRC (Central Library)

8.5 Admin view

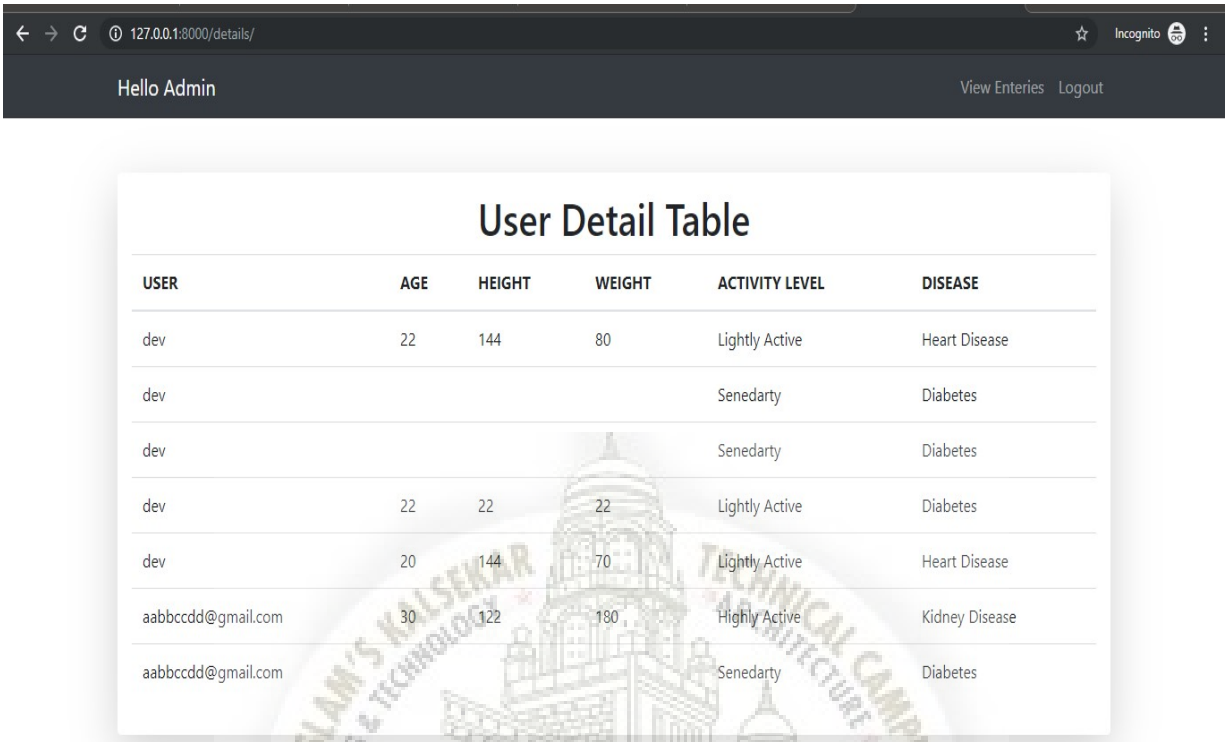

Figure 8.7: Admin View

Service By KRRC (Central Library)

8.6 Output

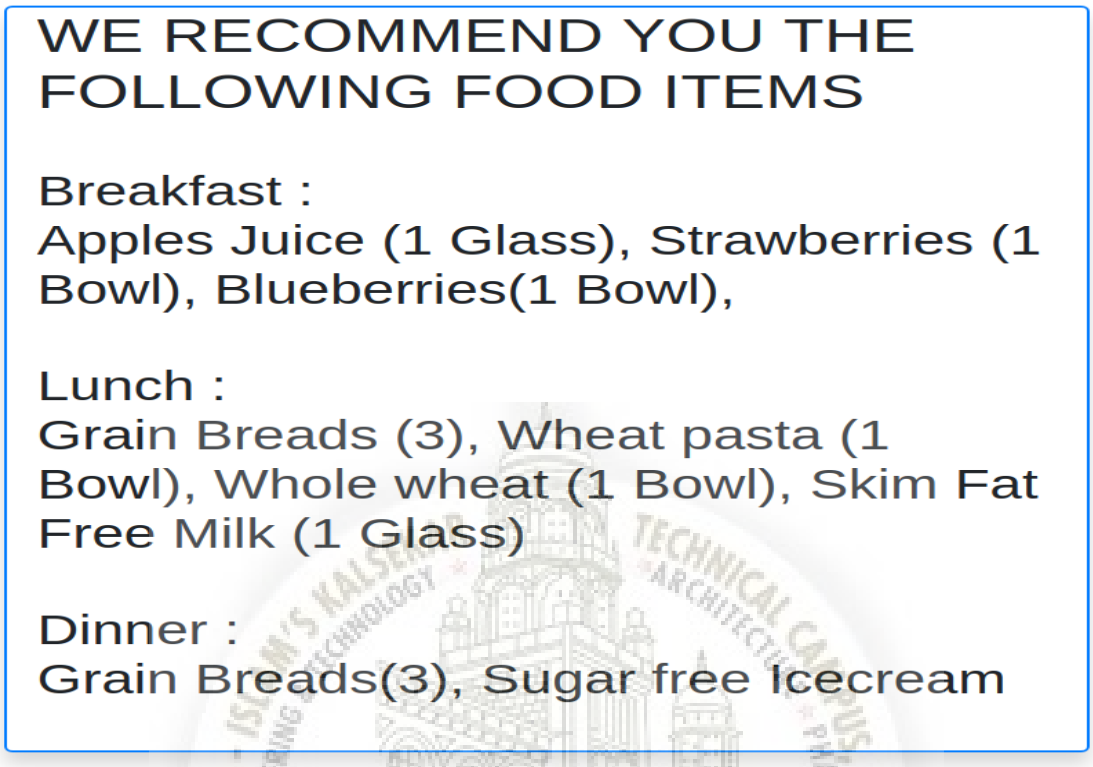

Figure 8.8: Output: generated Diet Plan

Service By KRRC (Central Library)

Chapter 9

Conclusion and Future Scope

9.1 Conclusion

A nutritious and wholesome balanced diet is a key to good health. A well-balanced diet includes eating the right amount of foods from the five main food groups. Most people will have three main meals a day. No single food contains all nutrients the body needs so it is important to eat a wide variety. The right amount of different nutrients can increase life expectancy by keeping the heart and body healthy, and preventing many long-term illnesses. Body weight can be kept to an acceptable level through healthy eating, leading to a fitter and more active lifestyle.This system aims to provide user with proper food item based on certain inputs provided by users.

9.2 Future Scope

- It can be extended to mobile applications which tracks user's activity
- More algorithms can also be used for better accuracy.
- We can also generate the recipes of different food items using scraper

Service By KRRC (Central Library)

References

[1] Kale, Ashvini, and Nisha Auti. "Automated menu planning algorithm for children: food recommendation by dietary management system using id3 for indian food database." Procedia Computer Science 50 (2015): 197-202.

[2] Shimada, Yasuyuki, et al. "A dietary menu-generating system to promote healthy life." 2006 SICE-ICASE International Joint Conference. IEEE, 2006.

[3] Aravind, J., and J. Dhalia Sweetlin. "Nutrient facts analysis using supervised learning approaches." 2017 Conference on Information and Communication Technology (CICT). IEEE, 2017

[4] Ponmani, S., and P. Vidhu Priya. "Classification Algorithms in Data Mining–A Survey." International Journal of Advanced Research in Computer Engineering Technology (IJARCET) Volume 6 (2017).

Service By KRRC (Central Library)

Achievements

1. Paper Presentation

   (a) *Diet Recommendation Using Machine Learning Algorithm*; Sabiya Malbari Abdul Rashid,Ifat Salim Alekar, Shaikh Mariya Irfan,Alamuri Ratnamala institute of engineering and technology(ShahpurThane dist),March 2019

2. Conferences

   (a) *Diet Recommendation Using Machine Learning Algorithm*;Sabiya Malbari Abdul Rashid,Ifat Salim Alekar, Shaikh Mariya Irfan, International conference on recent trends in Engineering,Technology and Management , April 2019(Venue:Alamuri Ratnamala institute of engineering and technology(Shahpur Thane dist)

Service By KRRC (Central Library)

Alamuri Ratnamala
Institute of Engineering and Technology
Accredited by NAAC with 'B+' Grade
ISO 9001 : 2015 CERTIFIED

ARMIET
Alamuri Ratnamala
Institute of Engineering and Technology
(Run by Koti Vidya Charitable Trust)

NATIONAL LEVEL EVENT

TECHFIESTA-2019

14th, 15th & 16th March 2019

Certificate of Excellence

This is to certify that, ~~Mr~~ / Ms Malbari Sabiya Abdul Rashid
has been awarded this certificate of appreciation for ~~Securing~~ / ~~Co-Ordinating~~ / Participating
in Paper Presentation on ~~Position in~~ Automated Diet using ML
held on 15-03-2019 organized by Alamuri Ratnamala Institute of Engineering and Technology,
A. S. Rao Nagar, Shahapur, Thane Dist, Maharashtra.

Principal | Rakesh President | Event Co-ordinator

Figure 9.1: Participation certificate

Alamuri Ratnamala
Institute of Engineering and Technology
Accredited by NAAC with 'B+' Grade
ISO 9001 : 2015 CERTIFIED

ARMIET
Alamuri Ratnamala
Institute of Engineering and Technology
(Run by Koti Vidya Charitable Trust)

NATIONAL LEVEL EVENT

TECH-FIESTA-2019

14th, 15th & 16th March 2019

Certificate of Excellence

This is to certify that, ~~Mr~~ / Ms Ifat Salim Alekar
has been awarded this certificate of appreciation for ~~Securing~~ / ~~Co-Ordinating~~ / Participating
in Paper Presentation on ~~Position in~~ Automated diet using ML
held on 15-03-2019 organized by Alamuri Ratnamala Institute of Engineering and Technology,
A. S. Rao Nagar, Shahapur, Thane Dist, Maharashtra.

Principal
President
Event Co-ordinator

Figure 9.2: Participation certificate

Service By KRRC (Central Library)

Figure 9.3: Participation certificate