SWASTHYA : THE VIRTUAL DIETICIAN

SOFTWARE ENGINEERING PROJECT REPORT

[Submitted in partial fulfillment]

As a part of the curriculum of

B.Sc (H) COMPUTER SCIENCE

Submitted by:

DIYA GARG (18044570007)

CHAKSHITA GUPTA (18044570015)

MISHIKA RAWAT (18044570032)

B.Sc. (H) COMPUTER SCIENCE

IV SEMESTER

Mata Sundri College for Women, University of Delhi

Mata Sundri Lane, New Delhi-110002

ACKNOWLEDGMENT

"…the beauty of the destination is half veiled and the fragrance of success is half dull, until the traces of those enlightening the path are left to fly with wind spreading word of thankfulness."

We hereby take the opportunity to express our deep gratitude to our mentor, **Ms. Ashema Hasti** (Assistant Professor, Department of Computer Science, Mata Sundri College for Women) for her patient guidance, enthusiastic encouragement and useful critiques of this project work. We would also like to thank her advice and assistance in keeping our progress on schedule.

We are also thankful to all our friends, batchmates, and other people who have directly or indirectly helped us during the preparation of this project.

DIYA GARG (18044570007)
CHAKSHITA GUPTA (18044570015)
MISHIKA RAWAT (18044570032)

CERTIFICATE

This is to certify that the project report entitled "**SWASTHYA: THE VIRTUAL DIETECIAN**" has been submitted by Diya Garg, Chakshita Gupta and Mishika Rawat, students of B.Sc.(H) Computer Science of Mata Sundri College for Women, in partial fulfillment for the academic year 2020-21. The project has been carried out under the supervision and guidance of **Ms. Ashema Hasti** (Assistant Professor, Department of Computer Science, Mata Sundri College for Women) and that the project has not formed the basis for the award previously of any other degree, diploma, fellowship or any other similar title.

_____

Ms. Ashema Hasti
(Project Guide)

ABSTRACT

'SWASTHYA- THE VIRTUAL DIETICIAN' proposes an intelligent agent for setting up diet plans based on the inputs provided by the user. The system creates a meal plan in accordance with a person's lifestyle and health requirements. The online artificial dietician is a system having artificial intelligence about human diets.

Due to the modern lifestyle, carefree attitude and being materialistic, people are taking their health and diet otherwise. These days people tend to suffer from numerous health disorders and fitness problems majorly due to an unbalanced diet. In the present scenario, there is a trade-off between health, wealth and time. Many a time, they are ignorant about the right nutrient value for a healthy being.

Therefore, to facilitate them with a proper diet chart along with light exercises according to their lifestyle and cope up with their busy schedule, a need for software emerges that can provide diet consultancies to the people at their preferred time and mobile phones without having to visit a dietician.

The prime objective of the software is to list all the possible diet plans along with the nutrient value of the food items for the user in accordance with his/her lifestyle by taking their height, weight, working hours, and eating hours and practices as inputs.

TABLE OF CONTENTS

TABLE OF DIAGRAMS

LIST OF TABLES

1. Problem Statement

Due to the modern lifestyle, carefree attitude and being materialistic, people are taking their health and diet otherwise. Therefore, to facilitate them with a proper diet chart according to their lifestyle and cope up with their busy schedule, a need for an app emerges that can provide diet consultancies to the people at their preferred time and mobile phones without having to visit a dietician. The users can take advantage of the app by registering themselves, entering the basic details and signing in with a username and password.

The prime objective of the app is to list all the possible diet plans along with the nutrient value of the food items for the user in accordance with his/her lifestyle by taking their height, weight, working hours, and eating hours and practices as inputs.

The app is beneficial for the young generation who live away from their homes and cannot have a proper diet maintained. This app provides them with alternatives to manage the balance. The another yet distinguishable aim of our Swasthya App is to provide solutions on how to gain more with minimum affordable eateries, a basic plan that suggests a diet that can fulfill the essential needs of the body and not only it replenishes the loss but also helps to gain energy. A person needs a dietician not only when he is malnutrition or is unable to get the best. The ever increasing problem today is obesity. Youth is stressed about how to lose weight healthily without starving or spending lumps on money on a gym membership. The Swasthya App comes to aid by providing a slow and steady yet robust plan that provides a diet with which you can loose/gain extra calories without any fret and lead a stress-free life because yes! What we eat is what we feel.

The diet plan not only covers the nutritional aspect but also provides light exercises that can help one to keep their body in shape and discipline. What makes this Swasthya App better over existing artificial dieticians is the added functionality that the registered users can have direct conversations through messages with a certified dietician(s) to get the best of our services. Moreover, it will keep a track of past inputs and data of the user and put forth the diet plan after considering the user's history.

2. Software Lifecycle Model

Swasthya follows **Incremental Process Model**.

We have used the incremental model as it combines elements of linear and parallel process flows. It generates working software quickly and early during the software lifecycle. This model is more flexible and less costly to change scope and requirements. It is easier to test and debug during a smaller iteration. In this model, the customers can respond to each built. Also, functionality can be refined and expanded in the later stages in the later software releases. The user can visualize the software before the completion of the entire project in order to evaluate and provide feedback. We are using this model as requirements are completely understood, however, small changes can be incorporated.

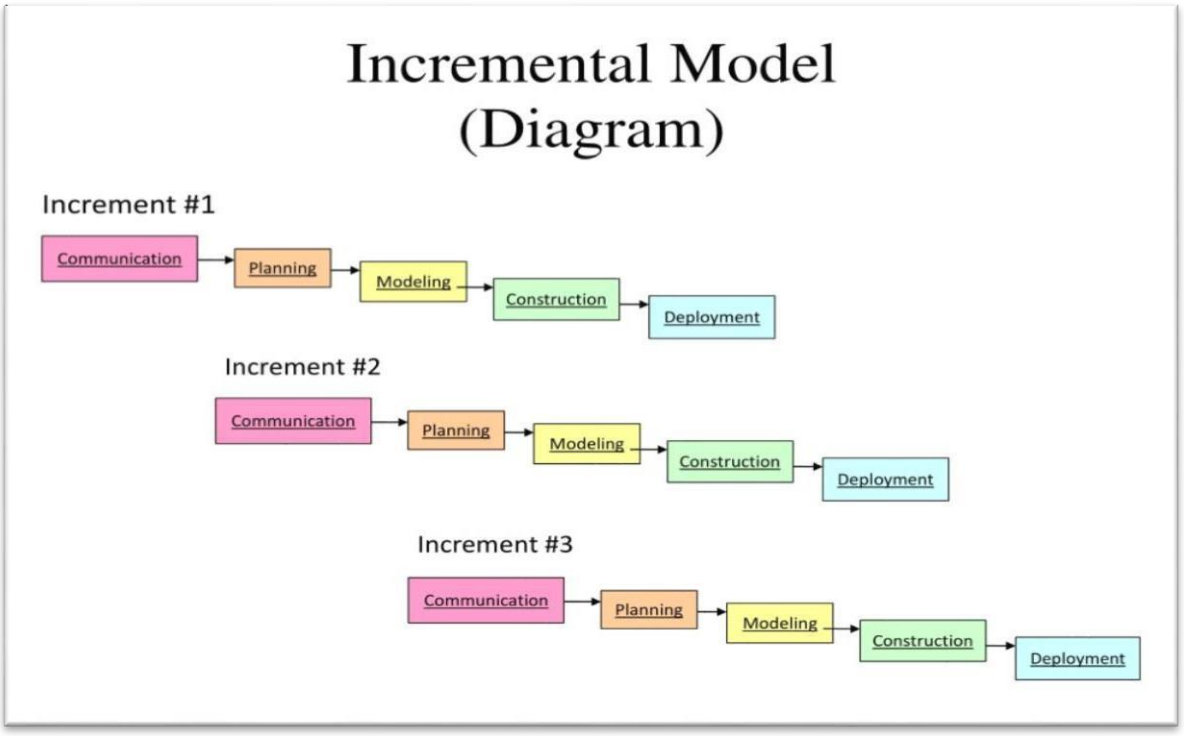


Fig 2.1 Incremental Model

3. REQUIREMENTS ANALYSIS

3.1 DATA FLOW DIAGRAMS

3.1.1 CONTEXT DIAGRAM

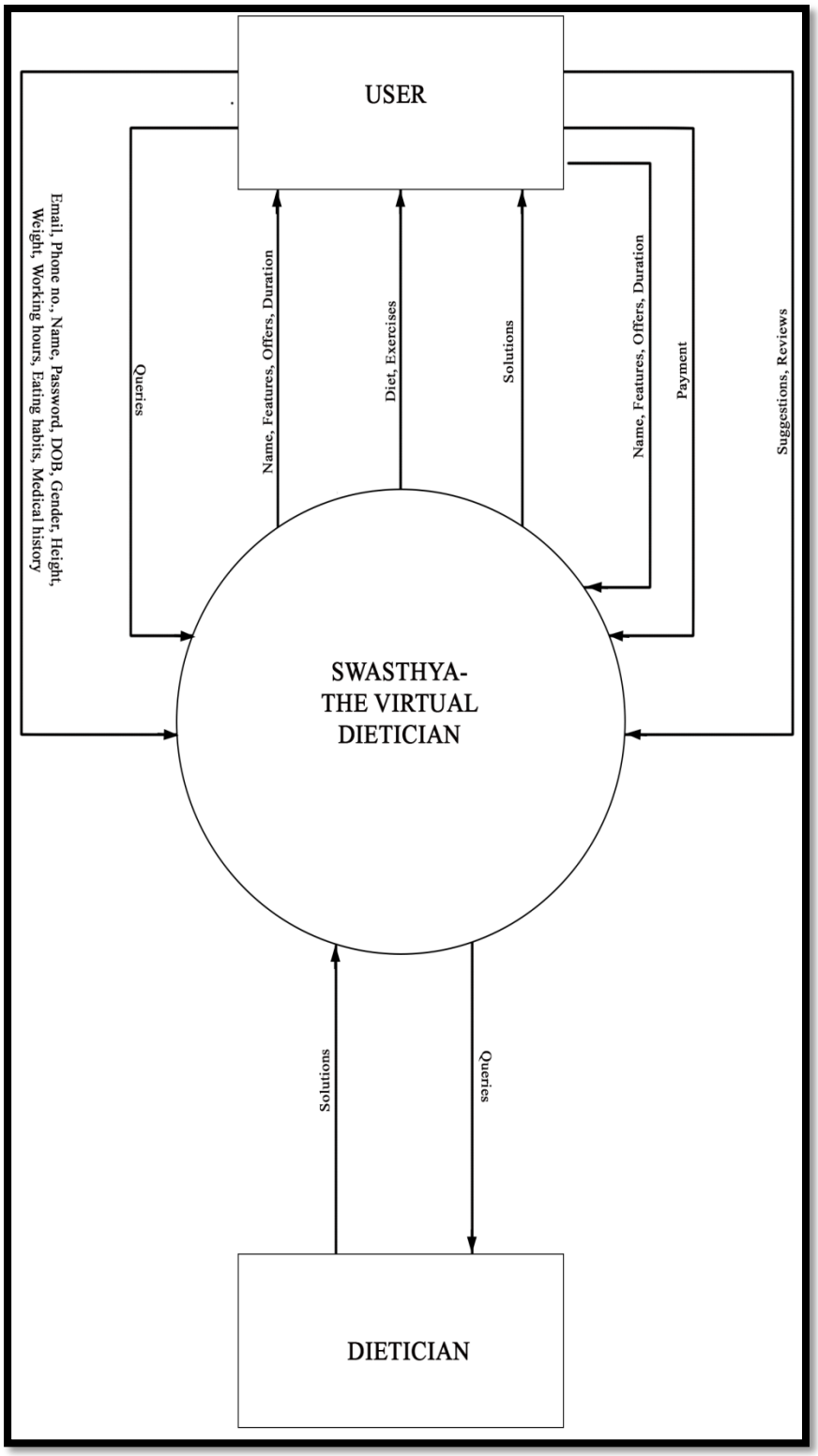


Fig 3.1 Context Diagram

3.1.2 LEVEL-1 DFD

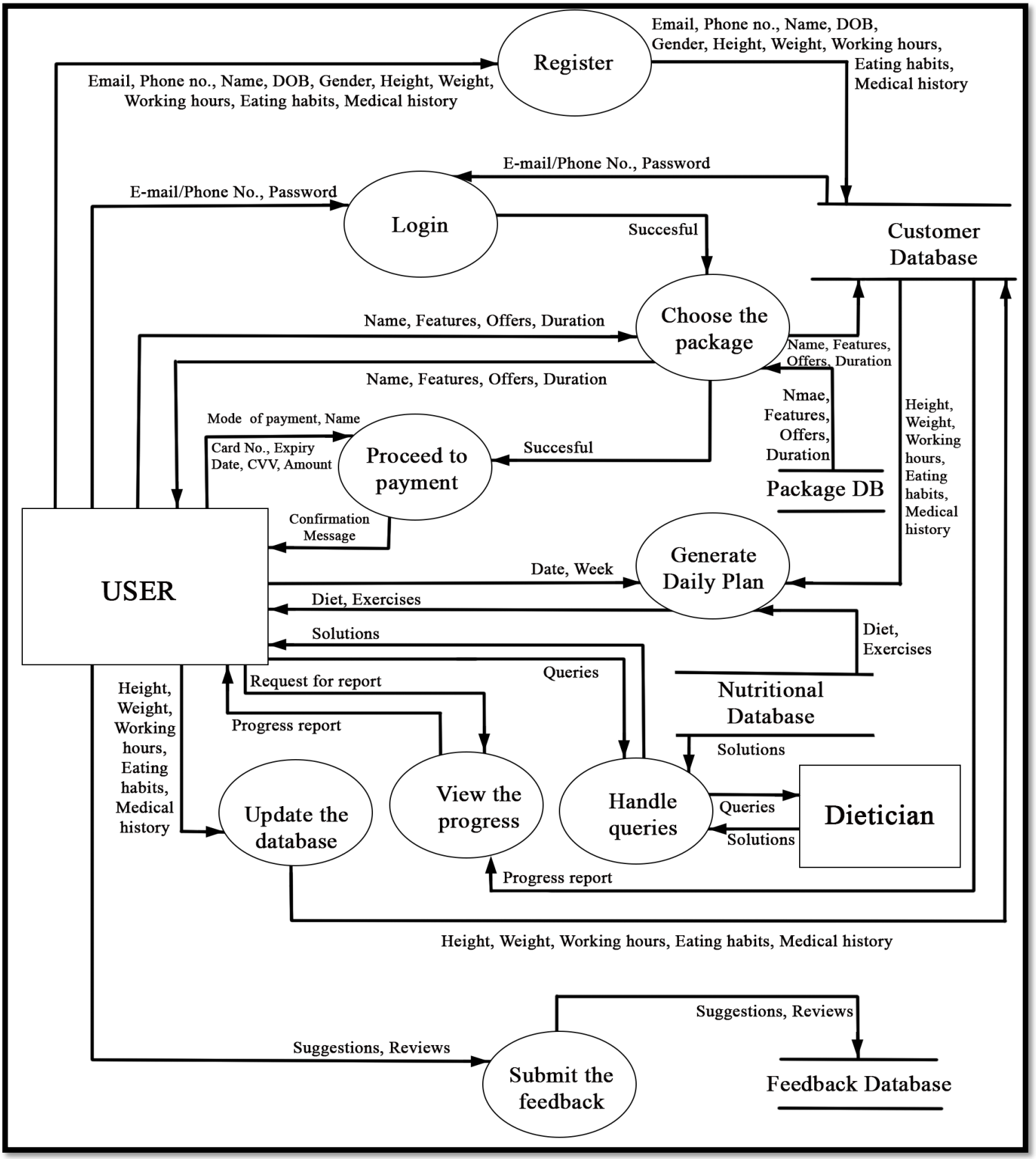


Fig 3.2 Level-1DFD

3.1.3 LEVEL-2 DFDs

REGISTRATION

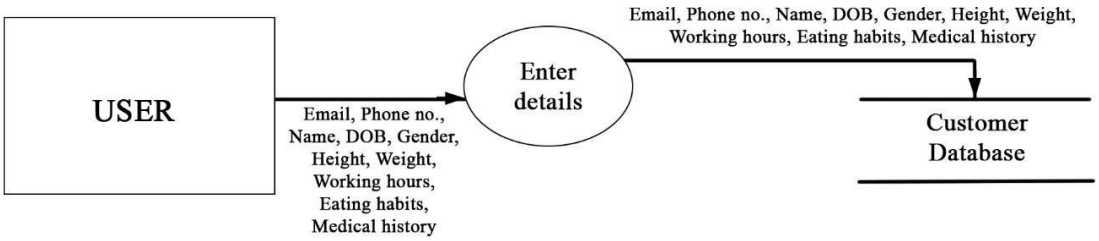


Fig 3.3 Level-2 DFD (REGISTRATION)

LOGIN

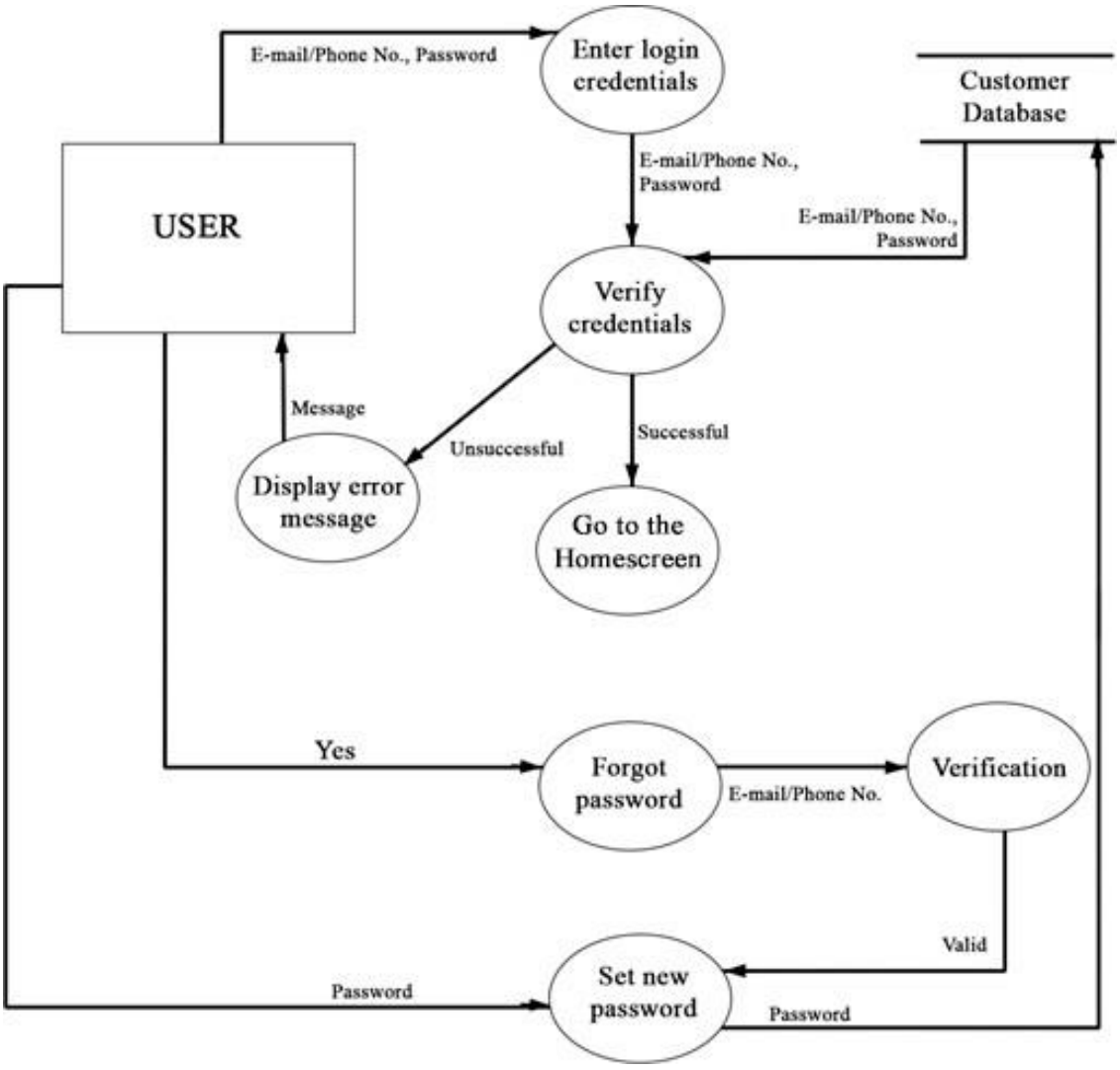


Fig 3.4 Level-2 DFD (LOGIN)

CHOOSE THE PACKAGE

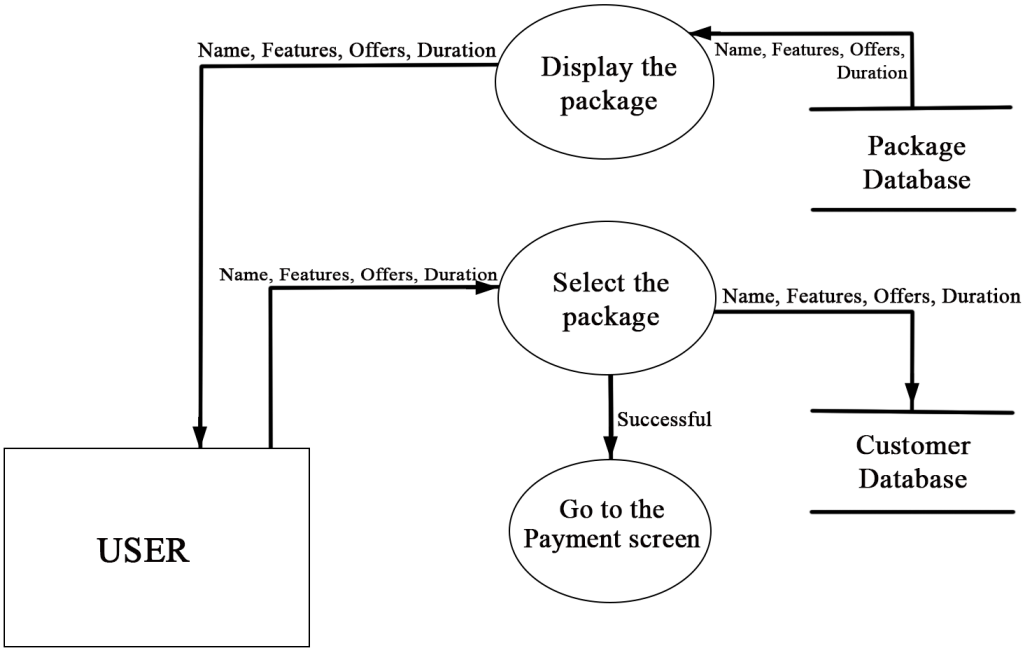


Fig 3.5 Level-2 DFD (PACKAGE)

PAYMENT

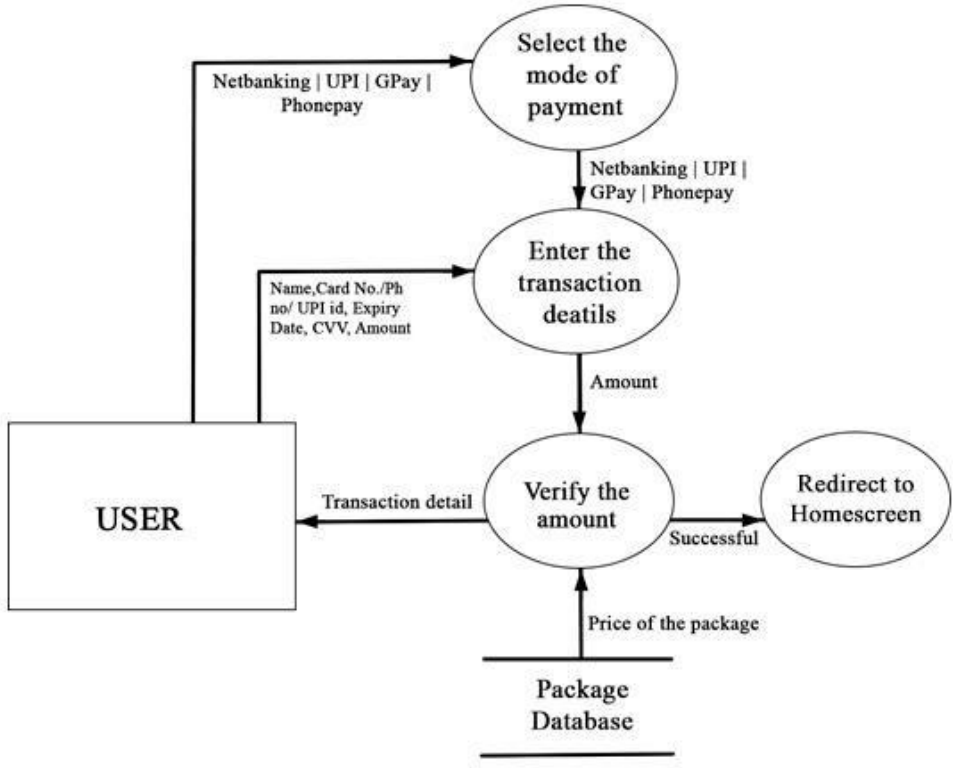


Fig 3.6 Level-2 DFD (PAYMENT)

GENERATION OF DAILY PLAN

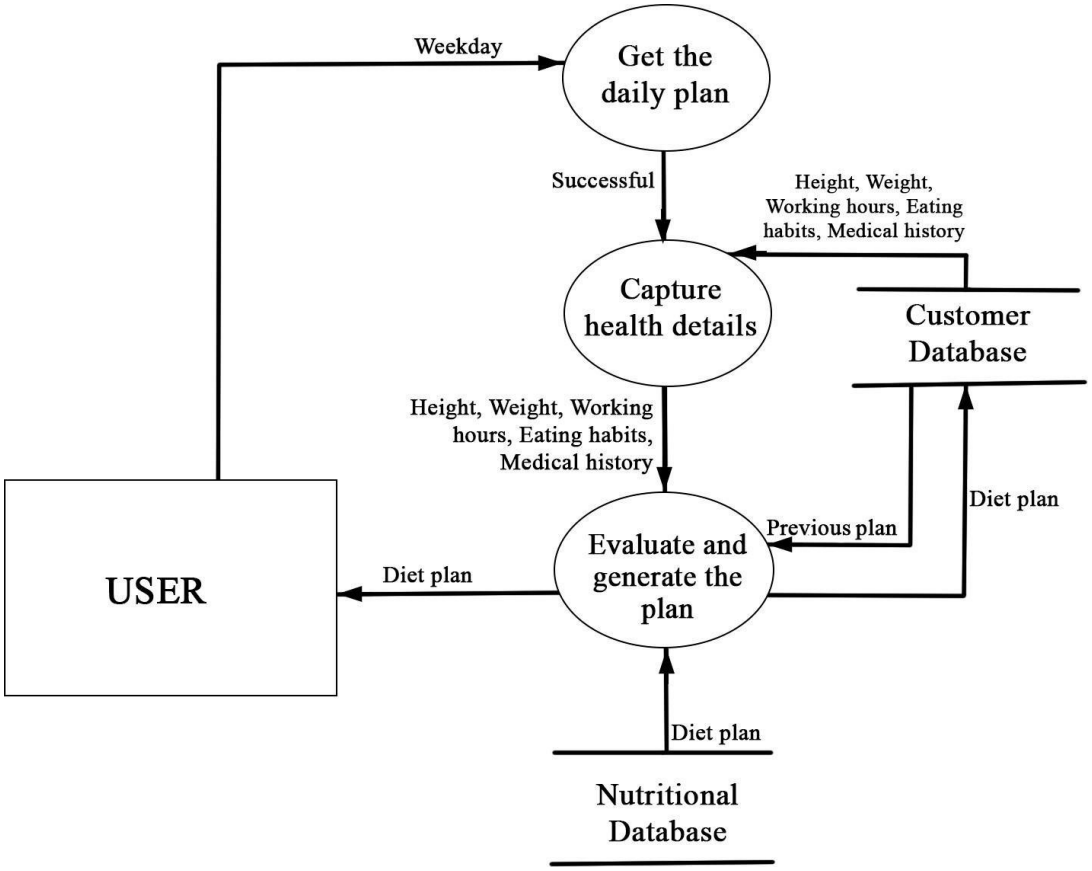


Fig 3.7 Level-2 DFD (GENERATION OF DAILY PLAN)

UPDATE THE DETAILS

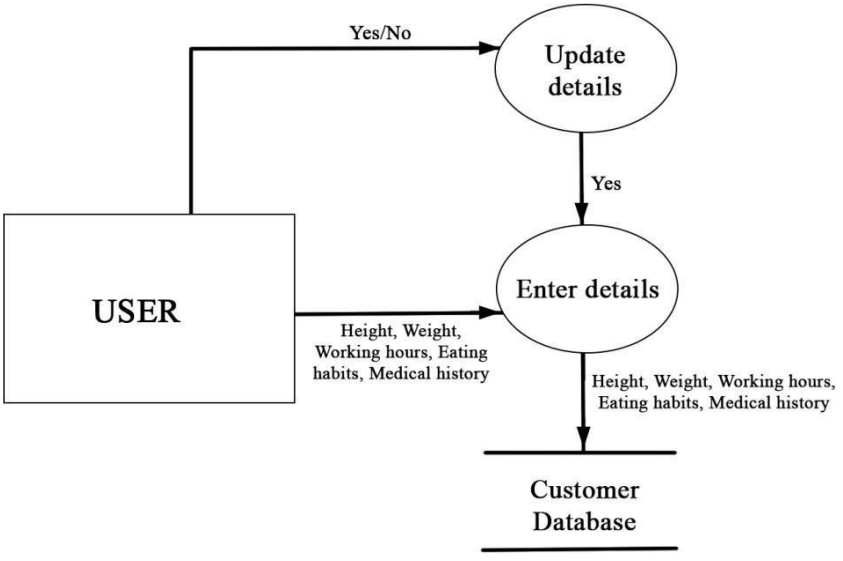


Fig 3.8 Level-2 DFD (UPDATE THE DETAILS)

MANAGE PROGRESS REPORT

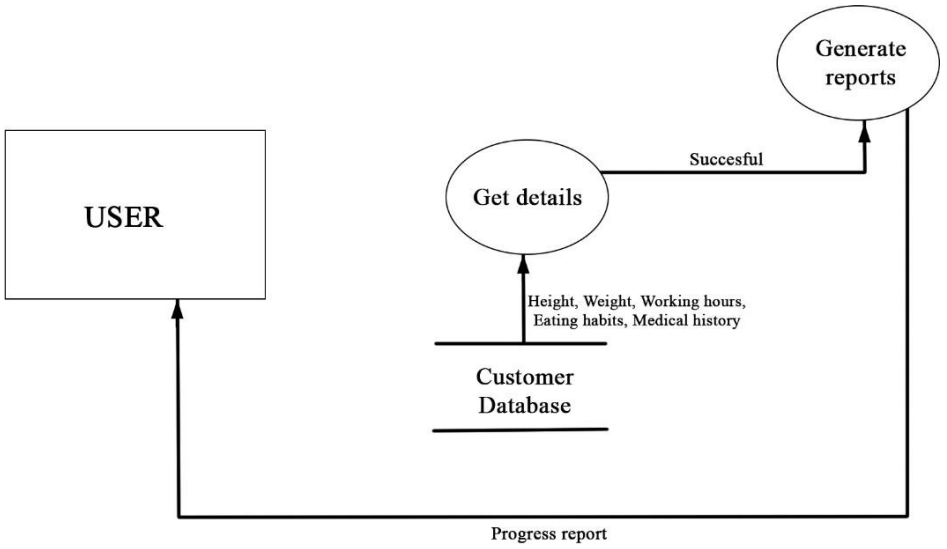


Fig 3.9 Level-2 DFD (REPORT)

QUERIES

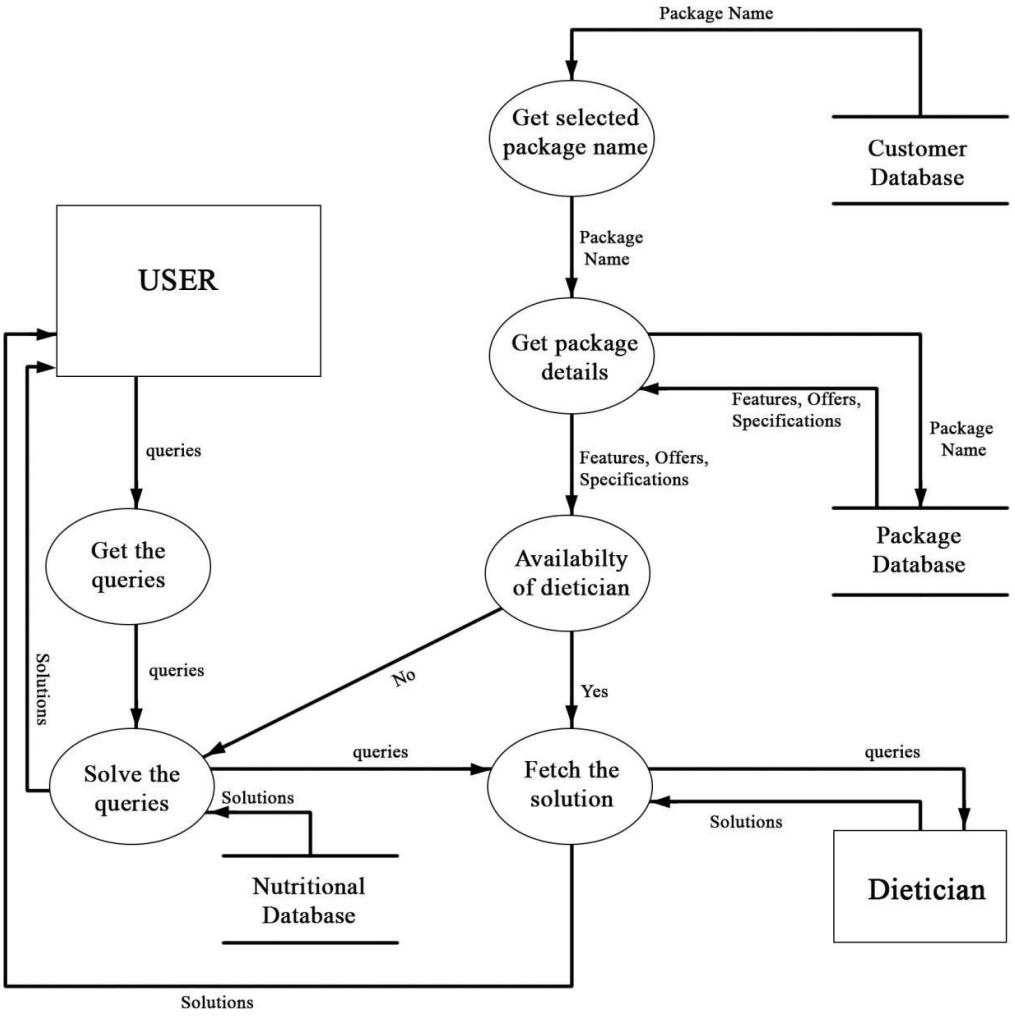


Fig 3.10 Level-2 DFD (QUERIES)

FEEDBACK

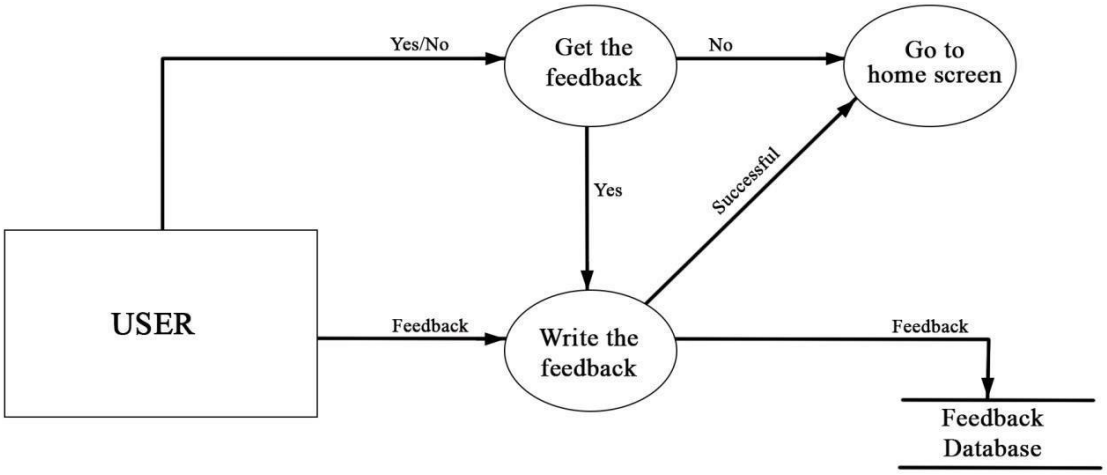


Fig 3.11 Level-2 DFD (FEEDBACK)

3.2 DATA DICTIONARY

Legal character: [a-z|A-Z]
Digit: [0-9]
Special character: [@|$|#|+|-|.]

1	Name	{Legal character}*
2	Email id	{Legal characters+Digit+special characters}*
3	Username	{Legal characters+Digit+special characters}*
4	Password	{Legal characters+Digit+special characters}*
5	Contact	{Digit+ Digit+ Digit+ Digit+ Digit+ Digit+ Digit+ Digit+ Digit+ Digit}
6	Date Of Birth	{Digit+ Digit+ Digit}*
7	Gender	{Legal character}*
8	Height & Weight	{Digit+ Digit}*
9	Working Hours	{Digit+ Digit}*
10	Medical History	{Legal character}*
11	Eating Habits	{Legal character}*
12	Queries	{Legal character}*
13	Payment	{Digit+ Digit+ Digit+ Digit}
14	Offers	{Legal characters+Digit+special characters}*
15	Feedback	{Legal character}*

Table 3.1: Data Dictionary

4. SYSTEM REQUIREMENTS SPECIFICATIONS

4.1 INTRODUCTION

The pace with which the world is growing it is often seen that we tend to ignore our health and later suffer the grave consequences. Wise men say that health is wealth and to emphasize this important prospect of life we are developing software that allows the user to keep a track of their diet and provide them with the best solutions that adjust with their day to day life.

4.1.1 Purpose

The major objective of this software is to provide the customer best service which includes diet plans, feedbacks and many other functionalities that aim towards the satisfaction of the consumer. This software is going to boost up the confidence of the user and make them more physically and mentally fit. Registered users can avail many other functions as well such as personal trainer and light exercises.

4.1.2 Scope

Swasthya has been developed to run on any platform and environment. This project also provides security with the use of Login-id and password, so that any unauthorized users cannot use the account. Only authorized users can access the software. The proposed system's scope is limited to functionalities as mentioned below :

1. **REGISTRATION**
   -The new user would need to create an account by entering these details- name, email-id and phone number, username and password.

2. **LOGIN**
   -The registered users can log in to the software directly by entering their username and password.
   -They can also use the feature of Forgot Password in case they forget their password.

3. **CHOOSE PACKAGE**
   -The user can select any preferred package which will later decide a few other special features provided to premium users.

4. **PAYMENT**

   -This will allow the user to make either online payment through credit/debit cards or wallets or through UPI method.

   -Cashbacks or vouchers or some gift hampers will be provided to the premium users.

5. **GENERATION OF DAILY PLAN**

   -The user can request the diet plan and/or any changes needed in the current plan.

6. **UPDATE THE DETAILS**

   -User data will be updated in the customer database so that the progress will be more appropriate in future references.

7. **MANAGE PROGRESS REPORT**

   -With this feature users will be able to monitor all the progress they have made and decide upon the effectiveness of the current plan.

8. **QUERIES**

   -For any queries such as a health disorder or abnormal effect on the body due to following plan, nutritional values or any change in plan, the user can use this option.

9. **FEEDBACK**

   -With the feedback provided by multiple users, app can improvise upon its functionality for the betterment of users.

4.1.3 Definitions, Acronyms and Abbreviations

DFD	Data Flow Diagram
IEEE	Institute of Electrical and Electronics Engineers
ERD	Entity Relationship Diagram
DOB	Date of Birth

Table 4.1: List of Acronyms

4.1.4 Overview

The basic idea is to build an app where customers can utilize the online dietician facilities. The system also acts as a useful resource for people with less privileges and those who want to invest in their health without going to a dietician or hiring a personal nutritionist.

4.2. PROJECT DESCRIPTION

Swasthya is an online application having artificial intelligence about human nutritional needs and diets. The user registers them on the app by providing the essential details like name, email, phone number, weight, height, medical history and eating habits. After registering, the user gets a choice to select the package and further can proceed to the payment process. Based on the personal and medical details, the user gets well- constructed diet plans along with light exercises that must be followed in the routine for better health. An e-mail or message notification is sent to the customer as soon as the payment is processed.

4.2.1 Product Perspective

4.2.1.1 System interfaces

A credit card processing system - The system will access the credit card processing system via its web services.

4.2.1.2 System specifications

- **Hardware requirements**
  The user must have a smartphone for the installation of the app.
- **Software requirements**
  The software developing team must have adequate knowledge of PHP Hypertext Preprocessor server side scripting language.

4.2.1.3 Communication interface

- The payment gateway & credit card processing system.
- In-built messaging and Email facility.

4.2.2 Product Functions

The online dietician Swasthya App is a mobile application. The system provides nine functionalities:

- Register
- Login
- Choose the package
- Payment
- Generate the daily plan
- Update the details
- Manage progress report
- Handle queries
- Submit the feedback

4.2.3 User Characteristics

User can use the app for adopting a healthy lifestyle. The user will select the package and give his/her preferences regarding the food items which will be taken in consideration for generating the daily diet plan. The progress report can be requested after following the diet plans for few days and updating the information. The Swasthya App will do self comparison and provide the user the progress report. The feedback can also be taken from the users for further improvements in the app.

4.2.4 General Constraints

- The interface will be in English only.
- The system works for single server.

4.2.5 Assumptions and Dependencies

The product requires back-end database server MySQL for storing the information entered by the user. Databases are also required for the packages and the feedback that user will provide. Following are the assumptions for our project:

- User must be having a smart phone for the installation of the app.
- User must have basic knowledge of English.

4.3 SPECIFIC REQUIREMENTS

4.3.1 External Interface Requirements

4.3.1.1 System Interface

A credit card system processing system – The system will access the credit card processing system via its web services.

4.3.1.2 User Interfaces

User of the system will have access to graphical user interface. There is no command line user interface.

4.3.1.3 Hardware Interfaces

The user must have a smartphone for the installation of the app.

4.3.1.4 Software Interfaces

The app will be built using PHP Hypertext Preprocessor language.

4.3.1.5 Communication Interfaces

-The payment gateway & credit card processing system.

- In-built messaging and Email facility.

4.3.2 Functional Requirements

- **FR1- Register**

  Input data: Name, address, phone number, email id, date of birth, gender, height, working hours, eating habits, medical history, username and password.

  Processing step: The user who is new to the app must register.

  Output Data: The user has registered in the app and has valid credentials.

- **FR2 – Login**

  Input data: E-mail id or phone number and password.

  Processing step: User can now login in the app.

  Output Data: The user can use the application and get to home screen.

- **FR3 –Choose package**

  Input data: Name, offers, features, duration of the package.

  Processing step: The user will select the desired package.

  Output Data: Proceed to payment gateway.

- **FR4 – Generate the daily plan**

  Input data: Day of the week.

  Processing step: Suitable diet and exercise plan will be made for user.

  Output Data: Diet and exercise plan will be provided to the user.

- **FR5 –Manage progress report**

  Input data: Height, weight, working hours, eating habits, medical history

  Processing step: Gathering information from database and generating report

  Output Data: Progress report.

- **FR6 – Query**

  Input data: User query.

  Processing step: The user can ask for changes in plan.

  Output Data: Updated plan (may or may not be available).

- **FR7 – Payment**

  Input data: Credit card number, month, year, CVV, amount.

  Processing step: The payment has to be made for the selected package.

  Output data: Status of package and confirmation message.

- **FR8 – Update the details**

  Input data: Height, weight, working hours, eating habits, medical history.

  Processing step: The details will be updated in the database.

  Output data: Updated details of the user

- **FR9 – Feedback**

  Input data: Reviews/suggestions.

  Processing step: Storing in the corresponding database.

  Output data: Feedback submitted.

4.3.3 Performance Requirements

The performance requirements are as follows:

- System login shall take less than 5 seconds.
- Only one user can login from same device.
- Shall return results within 10 seconds.
- Diet plans shall be processed within 120 seconds.
- App will be working 24 hours a day and 7 days a week.

4.3.4 Logical Database Requirements

- The system must store all the user account information as well as the diet chart records.
- All the data shall be stored in text-based flat files.
- For each user account, the login ID, name, password, age, email address shall be stored in one file.
- Each attribute shall be delimited by a semicolon, and all the entries shall be sorted alphabetically by the login id. Each entry shall also be delimited by a semicolon and sorted alphabetically by the Email id.

4.3.5 Design Constraints

- **Software Language Use**d- The languages that can be used for coding Swasthya :The Virtual Dietician are C, C++, Java and HTML.
- **Database Design**- In our database design, we give names to data flows, processes and data stores. Although the names are descriptive of data, they do not give details. Our interest is to build some details of the contents of data flow, processes and data stores. A data dictionary is a structured repository of data about data. It is a set of rigorous definitions of all DFD data elements and structures.

4.3.5.1 Standard Compliance

Report format: All the reports produced for this project are in compliance with the standard templates in accordance with the standard guidelines and policy.

Naming Conventions: All the documents are named using the standard naming conventions.

4.3.5.2 Hardware Limitations

Although you can run Android Studio in 2 GB Ram but it is highly recommend to use atleast 4 GB RAM, and if you use 8 GB RAM, then it will be a great experience. Now for processing power, Intel core i3 clocked at nearly 2 GHz is enough to handle most normal android application, but for writing big apps, a better processor like i5 or i7 is needed.

4.3.5.3 Reliability and Fault Tolerance

Probability of a piece of software operating without failure while in a specified environment over a set duration of time will be 85%.

Backup components shall be provided that automatically take the place of failed components, ensuring no loss of service.

4.3.5.4 Security Requirements

- Sensitive data isn't distributed among third party mediators.
- No sensitive data in backups.
- Memory is cleared and sensitive data is not stored for long.
- Sensitive data is not stored outside the app's storage system.
- Passwords are not exposed through the interface.
- Users are educated about the risks and prevention methods.

4.3.6 Software system attributes

4.3.6.1 Reliability

The average time of failure shall be 7 days. If the app crashes then a backup should be given in 7 days.

4.3.6.2 Availability

The dietician Swasthya App shall be available to users 24 hours a day, 7 days a week. If the bug appears then it should be handled within 12 hours.

4.3.6.3 Security

Users will be able to access only their personal information and not that of other users. Medical conditions and payment methods will be handled through a secure server to ensure the protection of user's credit card and personal information.

4.3.6.4 Maintainability

Any updates or detect fixes shall be made on server-side computers only without any patches required by the user.

4.3.6.4 Portability

User must have a smartphone that supports android 7 and later versions or ios 13 and later versions.

5. PROJECT PLANNING

5.1 PROJECT SCHEDULING

Work Tasks	Planned Start	Actual Start	Planned Complete	Actual Complete	Assigned Person(s)	Effort Allocated
Problem Statement	Jan.w1, d1	Jan.w1, d3	Jan.w2, d3	Jan.w2, d5	Diya, Chakshita, Mishika	3 person per week
Software Lifecycle Model	Jan.w1, d2	Jan.w1, d4	Jan.w1, d3	Jan.w1, d5	Mishika	1 person per week
Project Scheduling	Jan.w3, d3	Jan.w3, d3	Jan.w3, d4	Jan.w3, d4	Mishika, Chakshita	2 person per week
Timeline Chart	Jan.w3, d3	Jan.w3, d3	Jan.w3, d4	Jan.w3, d4	Diya	1 person per week
Software Requirement Specification	Jan.w3, d4	Jan.w3, d5	Feb.w1, d1	Feb.w1, d5	Diya, Chakshita, Mishika	3 person per week
Context Level Diagram	Feb.w1, d2	Feb.w1, d2	Feb.w1, d5	Feb.w1, d5	Diya, Chakshita	2 person per week
Data Flow Diagram 1	Feb.w2,d1	Feb.w2, d1	Feb.w2, d4	Feb.w2, d5	Diya, Chakshita, Mishika	3 person per week
Data Flow Diagram 2	Feb.w2, d3	Feb.w2, d5	Feb.w4, d3	Feb.w4, d3	Mishika, Diya	2 person per week
Data Dictionary	Feb.w3, d3	Feb.w4, d3	Feb.w3, d4	Feb.w4, d5	Chakshita	1 person per week
Project Metrics	Mar.w1, d1	Mar.w1, d3	Mar.w2, d1	Mar.w2, d5	Chakshita	1 person per week
Effort Estimation (Cocomo II model)	Mar.w2, d1	Mar.w3, d1	Mar.w3, d1	Mar.w3, d5	Chakshita, Mishika	2 person per week
Risk Analysis	Mar.w2, d3	Mar.w2, d5	Mar.w3, d5	Mar.w3, d5	Diya	1 person per week
ER Diagram	Mar.w3, d2	Mar.w3, d5	Mar.w3, d4	Mar.w4, d1	Mishika	1 person per week
Data Design	Mar.w3, d2	Mar.w3, d5	Mar.w3, d4	Mar.w4, d1	Chakshita	1 person per week
System Design	Mar.w3, d4	Mar.w4, d1	Mar.w4, d3	Mar.w4, d5	Diya, Chakshita	2 person per week
Testing	Mar.w4, d5	Apr.w1, d1	Apr.w1.d4	Apr.w1.d5	Diya, Mishika	2 person per week
References	Mar.w2,d3	Mar.w3, d1	Mar.w3, d3	Mar.w3, d5	Diya	1 person per week
Annexure	Mar.w2,d3	Mar.w3, d1	Mar.w3, d3	Mar.w3, d5	Diya	1 person per week

Table 5.1: Project Scheduling

5.2 TIMELINE CHART

WORK TASKS	JANUARY				FEBRUARY				MARCH				APRIL			
	W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4
Problem Statement																
Software Lifecycle Model																
Project Scheduling																
Timeline Chart																
SRS																
Context Level Diagram																
Data Flow Diagram 1																
Data Flow Diagram 2																
Data Dictionary																
Project Metrics																
Effort Estimation (COCOMO)																
Risk Analysis																
ER Diagram																
Data Design																
Component Design																
Testing																

Table 5.2: Timeline Chart

5.3 EFFORT EASTIMATION & FP- BASED COMPUTING

- Function Point Metric is an example of Product metrics for Analysis Model.
- It is used as a means for measuring the functionality delivered by a system and also examines requirement/ analysis model for predicting size of resultant system .
- Using historical data, Function Point metric can be used to :-
  i. Estimate the effort or cost required to design, code or test the software.
  ii. To predict number of errors that will be encountered during testing.
  iii. Forecast number of components or number of projected source links in implemented system. Function points are derived using empirical relationship based on countable (direct) measures of software's information domain and quantitative assessment of software complexity.
- Software Information Domain Values consists of number of:-
  i. External Inputs (EI)
  ii. External Outputs (EO)
  iii. External Query (EQ)
  iv. Internal Logical Files (ILF)
  v. External Interface Files (EIF)

To compute function points (FP), the following relationship is used:

FP= Count total × [0.65 + 0.05 × ∑ (F_i)],

where Count total= Sum of all Function Point entries

Calculation of Value Adjustment Factors (VAF) is based on the responses of the following questions:

1	Does the system require reliable backup and recovery?	4
2	Are specialised data communications required to transfer the information to and from the application?	3
3	Are there distributed processing functions?	3
4	Is performance critical?	3
5	Will the system work in an existing heavily utilised operational environment?	3
6	Does the system require online data entry?	5
7	Does the online data entry require input transaction to be built over multiple screens or operations?	5
8	Are Internal Logical Files updated online?	5
9	Are input-output queries or files complex?	2
10	Is the internal processing complex?	4
11	Is the code designed to be reusable?	4
12	Are conversion and installation included in design?	1
13	Is the system designed for multiple installations in multiple organizations?	3
14	Is the application designed to facilitate changes and ease of use by the user?	4
COUNT TOTAL (Σ F_i)		**49**

Table 5.3: Value Adjustment Factors (VAF)

The count total is the sum of all FP entries obtained from the following table:

INFORMATION DOMAIN VALUES	**COUNT**	**WEIGHING FACTOR**			**COUNT* WEIGHING FACTOR (SIMPLE)**
		SIMPLE	**AVERAGE**	**COMPLEX**	
External Inputs(EI)	48	3	4	6	144
External Outputs(EO)	12	4	5	7	48
External Queries(EQ)	1	3	4	6	3
Internal Logical Files(ILF)	`10	7	10	15	70
External Interface Files(EIF)	1	5	7	10	5
TOTAL COUNT					270

Table 5.4: Weighting factor of information domain values

FUNCTION POINT (FP) = Total Count × [0.65 + (0.01× Σ (F_i))]

= 270 × [0.65 + (0.01× 49)]

=307.80

= 308

5.4 COST ESTIMATION : COCOMO II MODEL

Barry Boehm gave a hierarchy of software estimation models called COCOMO i.e. constructive cost model. The original COCOMO model was widely used in the industry and was later evolved into a comprehensive model. Estimation model is called COCOMO II model.
COCOMO II is a hierarchy of estimation models that consists of:

1. **Application composition model**- It is used during the prototyping of user interfaces, assessment of process during system and software interaction and evaluation of technology maturity.
2. **Early design stage model**- It is used once. The requirement has been stabilized and basic software architecture is established.
3. **Post-architecture stage model**- This model is used during the construction of software.

- **Application of Composition Model**

These model use sizing information for which 3 options are available which are object points, function points and lines of source code.

Object-point is an indirect software measure computed using counts of number of screens on the user interface, number of reports generated and number of reusable components and 3 GL Components required to build the application. Each object instance is classified into one of the three complexity levels: Simple, Medium or Difficult. Complexity is a function of number and source of client and server data tables required to generate screen or report and number of views or sections presented as part of the screen or report.

Complexity weighting for object types-

OBJECT TYPE	COMPLEXITY WEIGHT		
	Simple	Medium	Difficult
Screens	1	2	3
Reports	2	5	8
3GL Components			10

Table 5.5: Complexity weight for object type

Productivity rate for object points-

Developer's experience or capability	Very Low	Low	Normal	High	Very High
Environment maturity or capability	Very Low	Low	Normal	High	Very High
Value	4	7	13	25	50

Table 5.6: Productivity weight for object point

Object point count= Original no. of object instances*Weighing factor(Simple)

= (17*2)+(0*5)

= **34**

0% of the components are re-usable.

NOP = New Object Points (or Object Point Counts)

= (Object Points)*[(100-% of reuse)/100]

= 34*[(100-0)/100]

= **34**

The developer's experience and capability in a similar environment is low. PROD (Productivity Rate) = 7

Estimated Effort = NOP/PROD

= 2.4285714

= 2.43 PM

The total number of screens will be more in number while actually building the app. Here, only some of the sample screens are shown, and our effort calculated is 2.43 PM which is according to 17 screens only.

5.5 RISK ANALYSIS

- **SOFTWARE RISKS**

  It involves 2 aspects: uncertainty and loss. Uncertainty means risk may or may not happen. If risk becomes reality, then unwanted consequences or loss will occur.

- **PHASES INVOLVED IN RISK ANALYSIS AND MANAGEMENT**

  1. Risk Identification
  2. Risk Analysis
  3. Risk ranking and assessment
  4. Creating risk plan or RMMM plan

- **TYPES OF RISKS**

  According to general categorization there are 3 types of risks:

  1. Known Risk
  2. Predictable Risk
  3. Unpredictable Risk

  Another category of risk type:

  1. Project Risk
  2. Technical Risk
  3. Business Risk

  Project Risk: Identify potential problems that might occur in budget, schedule and staffing. It also includes project complexity, project size and degree of structural uncertainty.

  Technical Risk: Identify potential design problem, implementation problem, interface problems, verification problem and maintenance problem. They threaten the quality of the software produced.

  Business Risk: Threatens the viability of the software to be built and often jeopardize the project or the product. There are 5 types of business risks:

  a) Market risk
  b) Strategic risk
  c) Sales risk
  d) Management risk
  e) Budget risk

- **ASSESSING OVERALL PROJECT RISK**

1. Have top software and customer mangers formally committed to support the project? **YES**

2. Are end users enthusiastically committed to the project and the system / product to be built? **YES**

3. Are requirements fully understood by the software engineering team and its customers? **YES**

4. Have customers been involved fully in the definition of requirements? **YES**

5. Do end users have realistic expectations? **YES**

6. Is the project scope stable? **YES**

7. Does the software engineering team have the right mix of skills? **YES**

8. Are project requirements stable? **YES**

9. Does the project team have experience with the technology to be implemented? **YES**

10. Is the number of people on the project team adequate to do the job? **YES**

11. Do all customer / user constituencies agree on the importance of the project and on the requirements for the system / product to be built? **YES**

S.No	Risks	Category	Probability	Impact	RMMM Plan
1	Customer will change the requirements	Project Size	80%	Critical	Conduct multiple reviews so that the requests are well understood.
					Set a deadline for proposing changes after which changes proposed would be chargeable.
2	Size estimate may be signifficantly low.	Project Size	60%	Critical	Collect more historical data to get accurate estimates.
					Requirements should be complete and well understood.
3	Delivery deadline will be tightened.	Business	50%	Critical	Continuously trace the timeline chart.
					Hire experts to meet the deadline.
4	Staff inexperience	Staff size and Experience	30%	Critical	Appointment of experienced staff.
					Training of existing staff.
5	Lack of training on tools	Development Environment	80%	Marginal	Prefer tools that the staff is experienced with.
					Allocate the available tools to the team based on their skillset.

Table 5.7: Risk Management Table

6. DESIGN

6.1 ER DIAGRAM

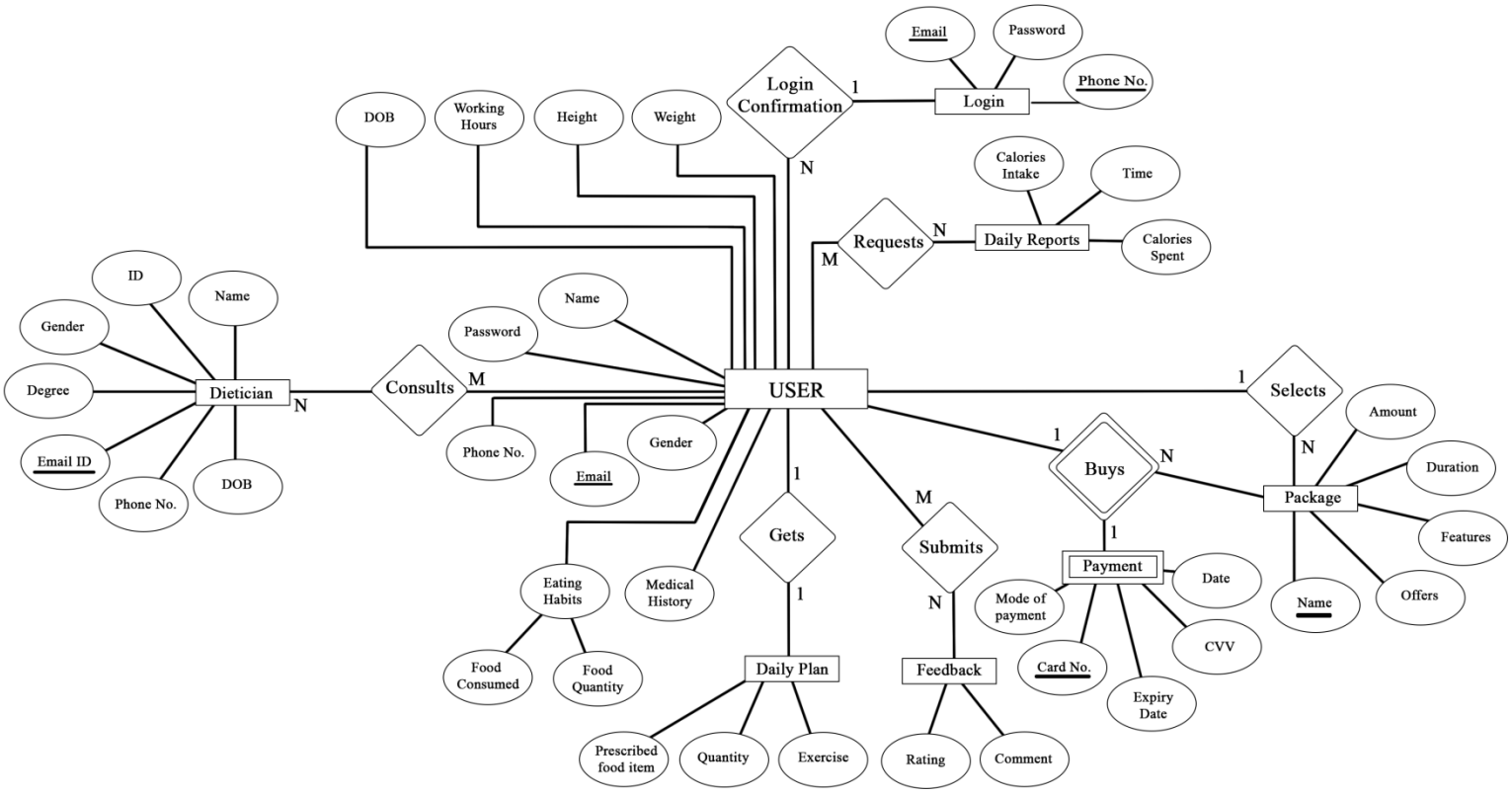


Fig 6.1 ER diagram

6.2. DATA DESIGN

User

Field Name	Type	Specifications	Constraint	Unique	Description
Name	Character	10 alphabetic characters	Not Null	Yes	User's name
Gender	Character	1 alphabetic character(M/F)	Not Null	No	User's gender
Password	Character	10 alpha-numeric characters	Not Null	Yes	Login password of user, must contain at least 8 characters including special ones
Phone No.	Integer	10 interger characters	Not Null	Yes	Login password of user, must contain at least 8 characters including special ones
Email	Varchar	20 alpha-numeric characters	Not Null, Primary Key	Yes	Gives the email id of the user
Medical History	String	50 alpha- numeric characters	Not Null	No	Gives the details of user's medical history
Height	Integer	2 integer characters	Not Null	No	User's height
Weight	Integer	2 integer characters	Not Null	No	User's weight
DOB	Date	MM/DD/YY' Format	Null	No	User's date of birth
Working Hours	Integer	2 integer characters	Not Null	No	The no. of hours the user works
Food consumed	String	50 alpha- numeric characters	Not Null	No	The food user likes to consume on general basis
Food consumed	String	50 alpha- numeric characters	Not Null	No	The quantity of food user consumes in a day

Table 6.1: Data Design Table for User

Dietician

Field Name	Type	Specifications	Constraint	Unique	Description
Name	Character	10 alphabetic characters	Not Null	Yes	Dietician's name
Gender	Character	1 alphabetic character(M/F)	Not Null	No	User's gender
Phone No.	Integer	10 interger characters	Not Null	Yes	User's contact no. must be of 10 digits
Email	Varchar	20 alpha-numeric characters	Not Null, Primary Key	Yes	Gives the email id of the user
Degree	String	50 alpha-numeric characters	Not Null	No	Gives user the qualifications and specialisations of the dietician
DOB	Date	MM/DD/YY' Format	Null	No	User's date of birth
Id	Character	6 alpha-numeric characters	Not Null	Yes	This provides an ID to the dietician

Table 6.2: Data Design Table for Dietician

Daily Plan

Field Name	Type	Specifications	Constraint	Unique	Description
Prescribes food item	String	50 alpha- numeric characters	Not Null	No	The dietician provides user with the food items that can be consumed by the him
Quantity	Integer	20 Integer characters	Not Null	No	The quantity of the prescribed food products that the user has to intake
Exercise	String	50 alpha- numeric characters	Not Null	No	The details of the light exercises that the user has to do

Table 6.3: Data Design Table for Daily Plan

Payment

Field Name	Type	Specifications	Constraint	Unique	Description
Mode of payment	Character	10 alphabetic characters	Not Null	Yes	Gives the mode of payment which the user prefers
Card Number	Integer	12 integer characters	Not Null, Primary Key	Yes	Card Number of the user
Expiry Date	Date	MM/DD/YY' Format	Not Null	Yes	Expiry date of the user's card
CVV	Integer	3 integer characters	Not Null	Yes	Gives the CVV code of the user's card
Date	Date	MM/DD/YY' Format	Not Null	No	The date at which the payment is being done

Table 6.4: Data Design Table for Payment

Login

Field Name	Type	Specifications	Constraint	Unique	Description
Email/Phone No.	Varchar/Integer	20 alpha-numeric characters/ 10 integer Characters	Not Null, Primary Key	Yes	Gives the email id of the user/ Login password of user, must contain at least 8 characters including special ones
Password	Character	10 alpha-numeric characters	Not Null	Yes	Login password of user, must contain at least 8 characters including special ones

Table 6.5: Data Design Table for Login

Package

Field Name	Type	Specifications	Constraint	Unique	Description
Name	Character	10 alphabetic characters	Not Null, Primary Key	Yes	Name of the package
Features	String	70 Alpha-numeric characters	Not Null	No	Details of the package
Amount	Integer	5 integer characters	Not Null	Yes	Cost of the package that user takes
Duration	Datetime	YYYY-MM-DD HH:MI:SS' Format	Not Null	Yes	Gives the duration of the package
Offers	String	30 Alpha-numeric characters	Not Null	No	Offers on the package that the user can avail

Table 6.6: Data Design Table for Package

Daily Report

Field Name	Type	Specifications	Constraint	Unique	Description
Calories Intake	String	50 alpha- numeric characters	Not Null	No	User provides the amount and details of calories he had taken in the day
Time	TimeStamp	YYYY-MM-DD HH:MI:SS	Not Null	No	The user provides the time of calorie intake
Calories Spent	String	50 alpha- numeric characters	Not Null	No	User provides the amount and details of calories he has spent by following the daily plan

Table 6.7: Data Design Table for Daily Report

Feedback

Field Name	Type	Specifications	Constraint	Unique	Description
Rating	Integer	1 integer character	Not Null	No	User can provide the app a rating on the scale of 5
Comments	String	50 alpha- numeric characters	Not Null	No	The user can write up about his/her experience with the app

Table 6.8: Data Design Table for Feedback

6.3. COMPONENT LEVEL DESIGN

Pseudocode for Select The Package module-

1. select_the_package() procedure begins
2. READ the package name, features, offers and duration from the package database
3. DISPLAY the package name, features, offers and duration
4. DO
5. GET the package name, features, offers and duration
6. STORE the package selected to the customer's database
7. PROCEED to payment screen //another module
8. WHILE select package is NULL
9. //End DO...WHILE
10. procedure ends

7. TESTING

We are performing **White Box Testing** for select the package module.

Pseudocode for select the package module is-

1. select_the_package() procedure begins
2. READ the package name, features, offers and duration from the package database
3. DISPLAY the package name, features, offers and duration
4. DO
5. GET the package name, features, offers and duration
6. STORE the package selected to the customer's database
7. PROCEED to payment screen //another module
8. WHILE select package is NULL
9. //End DO...WHILE
10. procedure ends

FLOWGRAPH

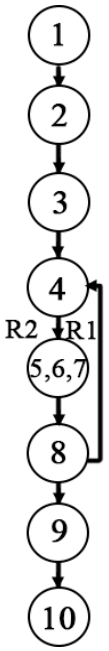


Fig 7.1: Flowgraph

CYCLOMATIC COMPLEXITY OF RESULTANT GRAPH

V (G) = Number of regions
= 2
V (G) = Edges-Nodes+2
= 8-8+2
=2
V (G) = Predicate nodes+1
= 1+1
= 2

LINEARLY INDEPENDENT PATHS FOR FLOW GRAPHS

Path 1: 1-2-3-4-5-6-7-8-9-10
Path 2: 1-2-3-4-5-6-7-8-4-5-6-7-8-9-10

TEST ID	INPUT VALUES	ACTUAL OUTPUT	EXPECTED OUTPUT
1	Package is selected	To be observed after execution	Display the selected package
2	Package is not selected	To be observed after execution	Show the packages to select until one is selected

Table 7.1: Test Cases Table

8. REFERENCES

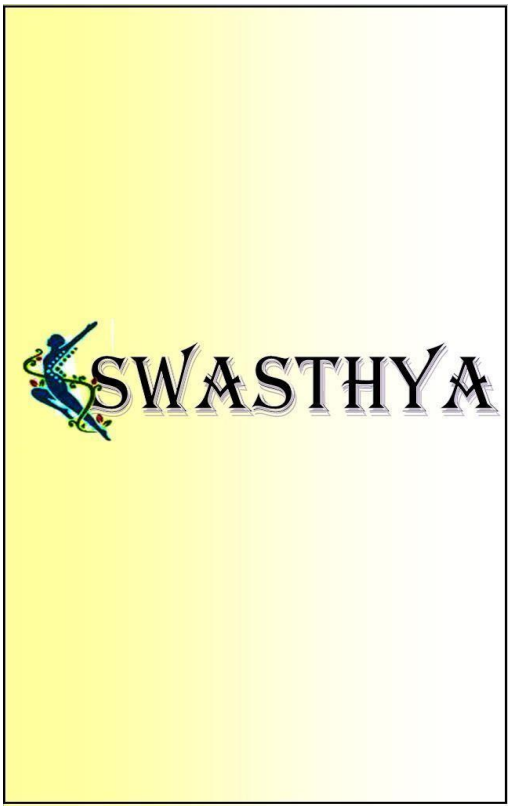
1. R.S. Pressman, Software Engineering: A Practitioner's Approach, McGraw-Hill, Ed 7,2010

2. P. Jalote, An Integrated Approach to Software Engineering, Narosa Publishing House, Ed 3, 2011

3. https://www.engpaper.com/cse/artificial-intelligence-dietician.html

4. https://www.smartics.eu/confluence/display/PDAC1/How+to+document+a+Software+Development+Project

9.ANNEXURES (SAMPLE SCREENS)



EI : 0

EO : 1

EQ : 0

ILF : 0

EIF : 0

Fig 9.1 Display Screen

EI : 1

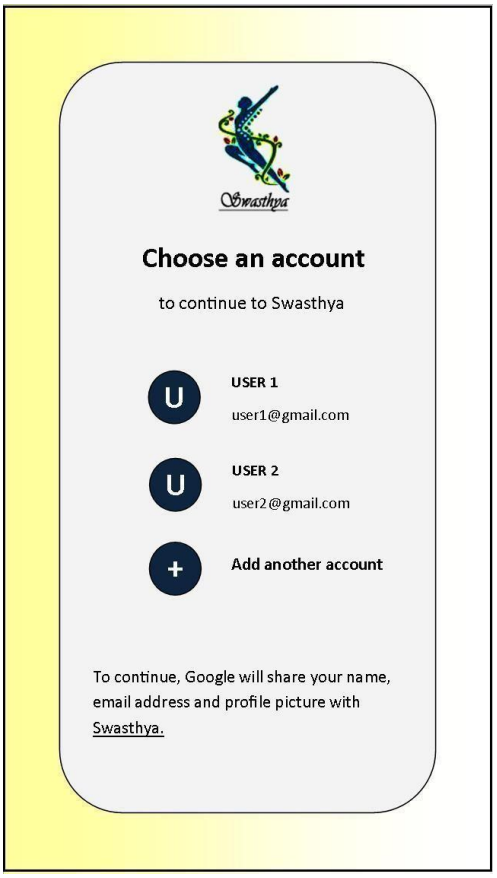
EO : 1

EQ : 0

ILF : 0

EIF : 0

Fig 9.2 Registration Screen



EI : 1

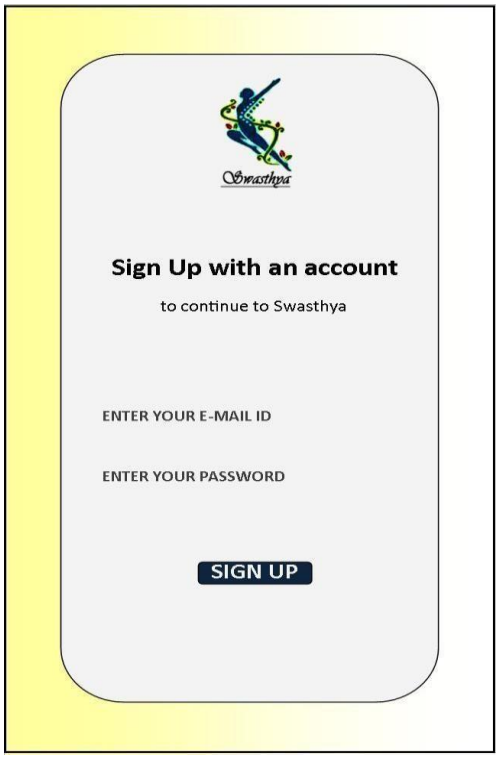
EO : 1

EQ : 0

ILF : 1

EIF : 0

Fig 9.3 Registration-1 Screen



EI : 3

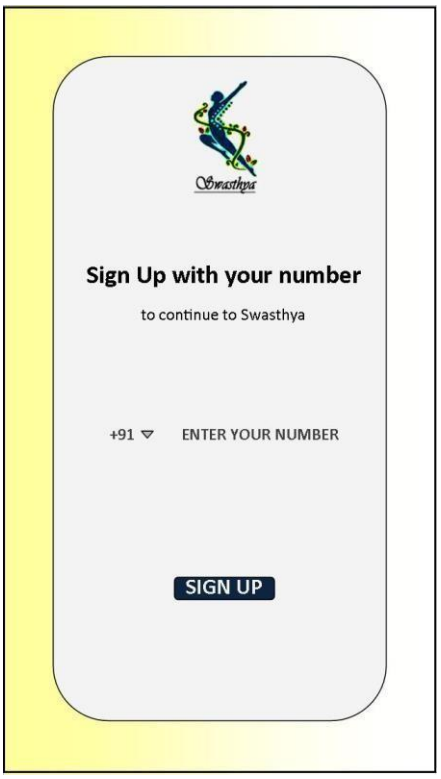
EO : 1

EQ : 0

ILF : 1

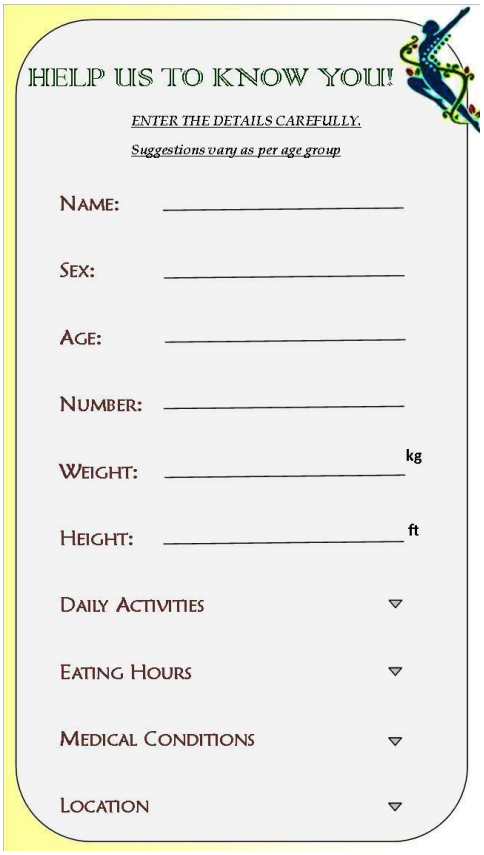
EIF : 0

Fig 9.4 Registration-2 Screen



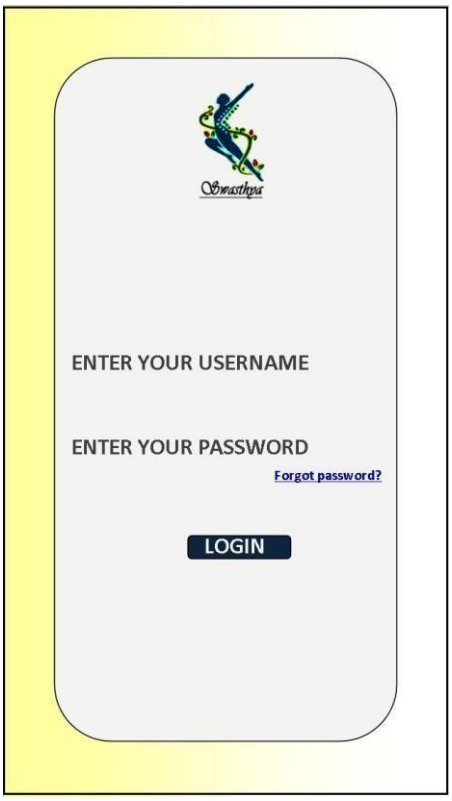
EI	: 3
EO	: 1
EQ	: 0
ILF	: 1
EIF	: 0

Fig 9.5 Registration-3 Screen



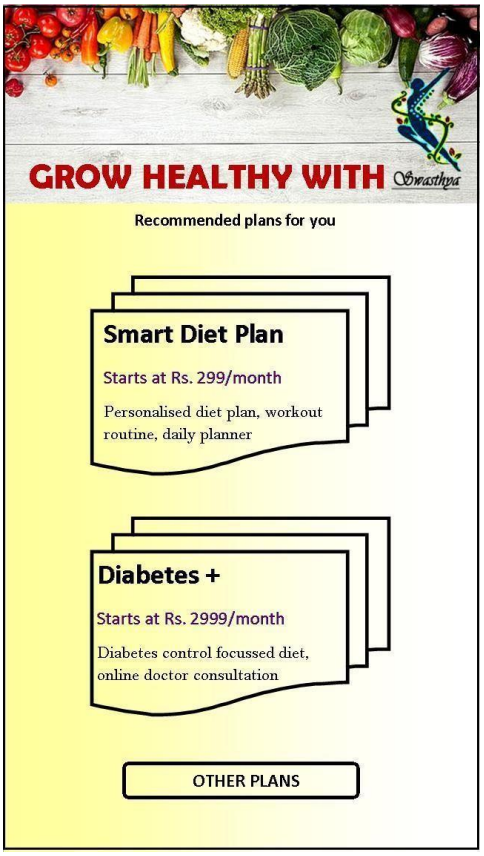
EI	: 10
EO	: 0
EQ	: 0
ILF	: 1
EIF	: 0

Fig 9.6 Upload Details



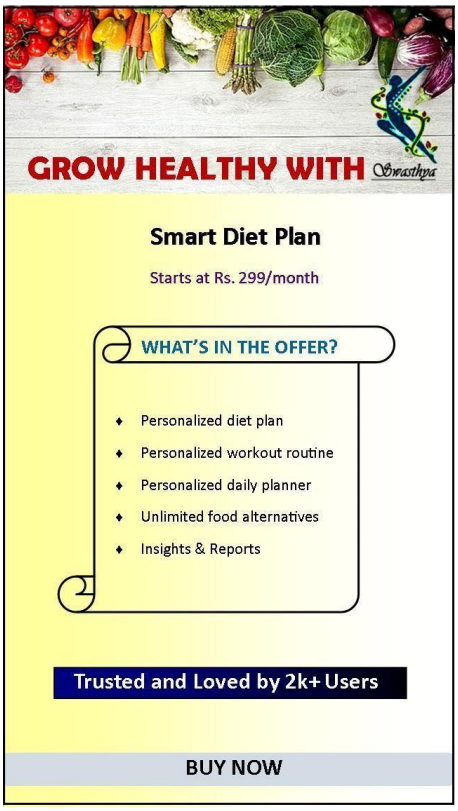
EI	: 3
EO	: 1
EQ	: 0
ILF	: 1
EIF	: 0

Fig 9.7 Login Screen



EI	: 1
EO	: 1
EQ	: 0
ILF	: 0
EIF	: 0

Fig 9.8 Packages-1 Screen



EI : 1

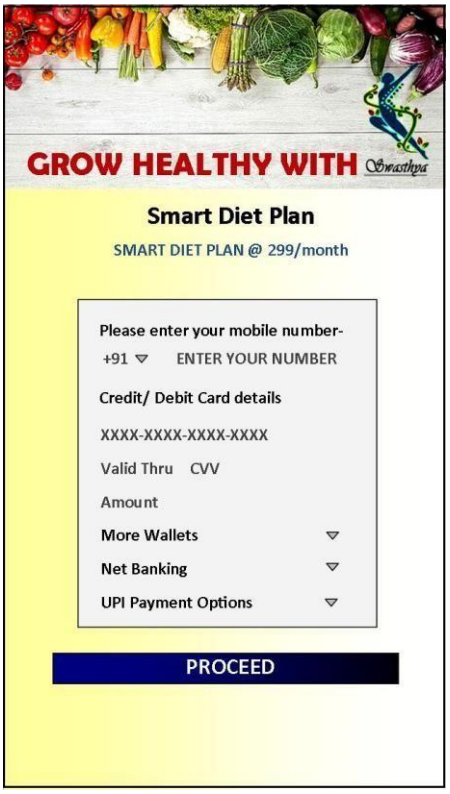
EO : 1

EQ : 0

ILF : 0

EIF : 0

Fig 9.9 Packages-2 Screen



EI : 6

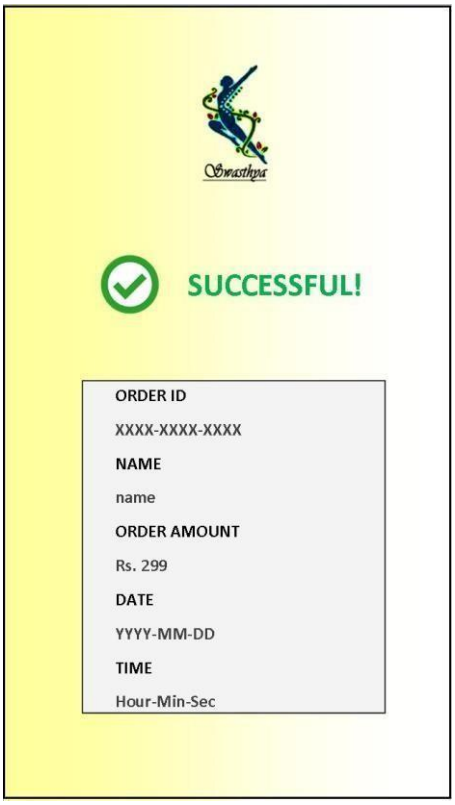
EO : 0

EQ : 0

ILF : 0

EIF : 0

Fig 9.10 Payment-1 Screen



EI	: 0
EO	: 1
EQ	: 0
ILF	: 1
EIF	: 0

Fig 9.11 Payment-2 Screen

EI	: 6
EO	: 1
EQ	: 1
ILF	: 1
EIF	: 1

Fig 9.12 Homescreen

EI : 2

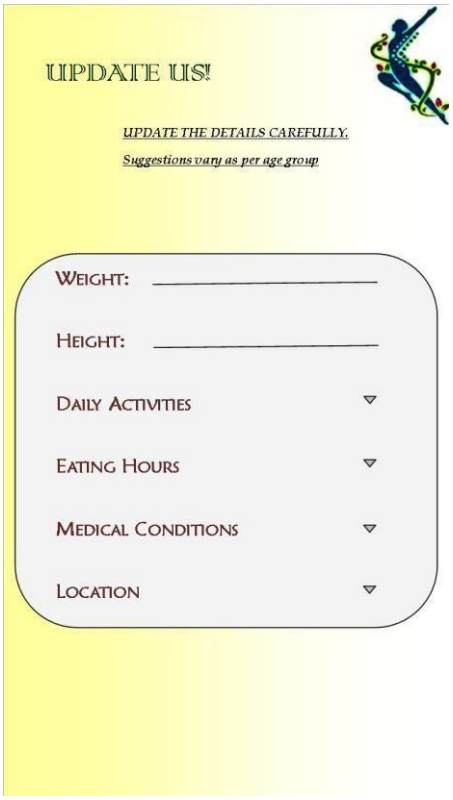
EO : 1

EQ : 0

ILF : 0

EIF : 0

Fig 9.13 Create Plan Screen



EI : 6

EO : 0

EQ : 0

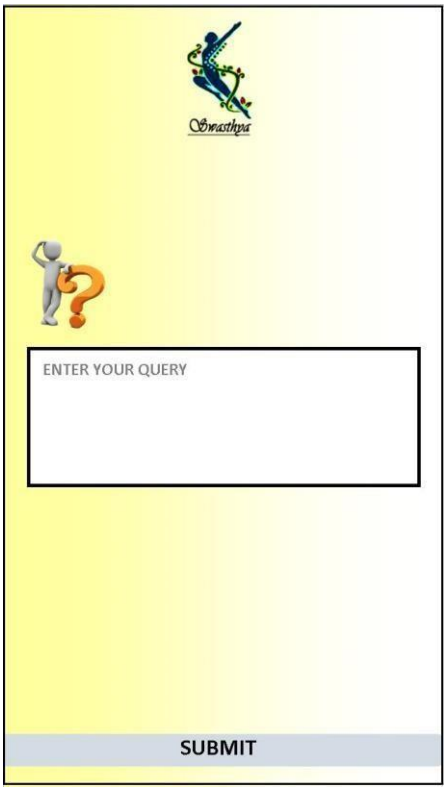
ILF : 1

EIF : 0

Fig 9.14 Update Details Screen

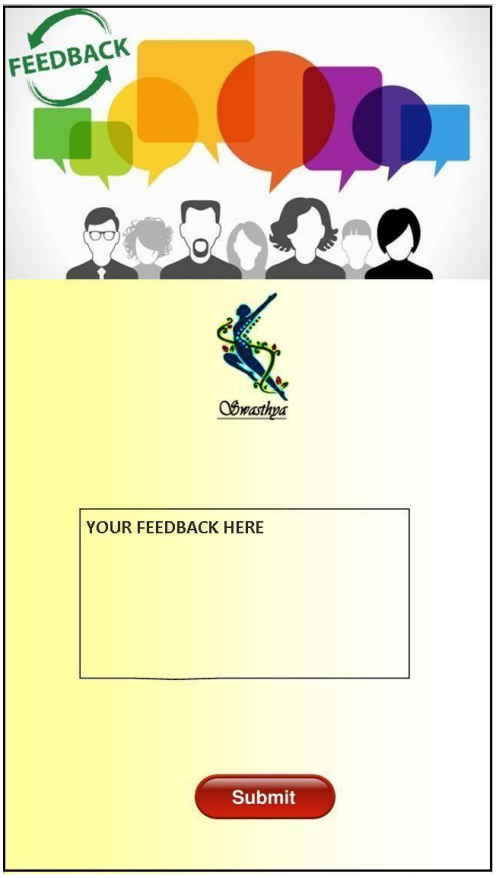
EI	: 1
EO	: 1
EQ	: 0
ILF	: 1
EIF	: 0

Fig 9.15 Report Screen



EI	: 2
EO	: 0
EQ	: 0
ILF	: 0
EIF	: 0

Fig 9.16 Queries Screen



EI	: 2
EO	: 0
EQ	: 0
ILF	: 1
EIF	: 0

Fig 9.17 Feedback Screen