

FEMME - THE PERIOD TRACKING APP

SOFTWARE ENGINEERING PROJECT REPORT
(Submitted in partial fulfillment)

As a part of the curriculum of
B.Sc. (H) COMPUTER SCIENCE



Submitted By:

**Parmeet Kaur(20044570023)
Vidhi Sharma(20044570020)
Tanya Rawat (20044570034)**

***B.Sc.(H)COMPUTER SCIENCE
IV SEMESTER***

Mata Sundri College for Women, University of Delhi
Mata Sundri Lane, New Delhi-110002

ACKNOWLEDGEMENT

It gives us immense pleasure to present you “**FEMME:The Period tracking app**”. We were lucky to get enormous support from extremely talented people, who deserve our great gratitude.

Firstly, we would like to thank our teacher and guide, **Ms. Ashema Hasti** who gave her valuable suggestions and ideas whenever we needed them. Also, She encouraged us to work on this project tirelessly by giving us numerous consultations.

We are also grateful to our parents for their constant support, guidance and providing us the necessary resources for the project.

Lastly, we would like to thank our classmates for their valuable suggestions for the betterment of this project and everybody who has helped us directly or indirectly in completion of this project.

We are immensely grateful to everyone involved in this project and without their inspiration and valuable suggestion it would not have been possible to develop the project within the prescribed time.

Parmeet Kaur(20044570023)

Vidhi Sharma(20044570020)

Tanya Rawat (20044570034)

CERTIFICATE

This is to certify that the project report entitled, "**FEMME- THE PERIOD TRACKING APP**", has been submitted by Parmeet kaur ,Vidhi Sharma and Tanya Rawat, students of B.Sc(H)Computer Science of Mata Sundri College for Women, in partial fulfillment for the academic year 2022-23. 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, University of Delhi)** 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

Over the past few years the interest in period tracking apps increased, which represent a sub-genre of quantified self apps in women health. They are available in a variety of complexity levels ranging from simple menstruation diaries up to applications with complex fertility calculation algorithms.

As we all know ladies share a love-hate relationship with periods. We hate the painful cramps, unbearable bloating and unpredictable mood swings. Nowadays girls are getting aware about the underlying reasons for irregular periods. One key way to understand your health status better with help of your menstrual cycle is to track it. While the traditional method to do so using a calendar or your own estimate isn't bad, using a period tracking app is a more efficient way. The users can take advantage of the app by registering themselves, entering the basic details and signing in.

The prime objective of our **FEMME-THE PERIOD TRACKING APP** is to not only remind us about the dates of the cycle and regularities but it also pays attention to these functions-The tracking of menstrual cycle-associated factors such as mood swings, symptoms, intake of medication and contraceptives, diet, vaginal discharge during period, food cravings, exercise and give us regular tips according to our situation. A menstrual calendar where period and ovulation dates as well as days on which additional data have been entered by users are highlighted in specific ways and also predict the dates of the coming month. Reminder about menstrual cycle, pills, general (like visiting doctor, health checkups).

For women, cycle length prediction can be useful to have an idea of when their period might arrive and the possible onset of Premenstrual syndrome (PMS). The value of having an accurate record of their menstrual cycles may be very important for some women, for their fertility, pregnancy and during the perimenopause. Your menstrual cycle is a direct indicator of your overall health, and periods are your body's way of telling you that things are working as they should. Having an extremely unpredictable or heavy period, or skipping a period, can indicate an existing underlying condition. By tracking and logging various details of your cycle, you will be able to recall things that you might otherwise forget when speaking with your healthcare provider. When you use this app, you'll always have the dates of your last period at your fingertips.

TABLE OF CONTENTS

1. PROBLEM STATEMENT.....	8
2. SOFTWARE LIFECYCLE MODEL.....	9
3. REQUIREMENT ANALYSIS.....	10-37
3.1 Data Flow Diagram.....	11-18
3.1.1 Context Diagram.....	11
3.1.2 Level-1 DFD.....	12
3.1.3 Level-2 DFD.....	13-18
3.2 Data Dictionary.....	19
3.3 Use Case.....	20-27
3.3.1 Use Case Diagram.....	20
3.3.2 Use Case Description.....	21-27
3.4 Sequence Diagram.....	28-37
4. SYSTEM REQUIREMENTS SPECIFICATIONS.....	38-44
4.1 Introduction.....	38-39
4.1.1 Purpose.....	38
4.1.2 Scope.....	38
4.1.3 Definitions, Acronyms and Abbreviations.....	39
4.1.4 Overview.....	39
4.2 Project Description.....	40-41
4.2.1 Product Perspective.....	40
4.2.2 Product Functions.....	40
4.2.3 User Characteristics.....	41
4.2.4 General Constraints.....	41
4.2.5 Assumptions and Dependencies.....	41
4.3 Specific Requirements.....	42-44
4.3.1 External Interfaces.....	42
4.3.1.1 System Interface.....	42
4.3.1.2 User Interface.....	42
4.3.1.3 Hardware Interface.....	42
4.3.1.4 Software Interface.....	42
4.3.1.5 Communication Interface.....	42
4.3.2 Functional Requirements.....	42
4.3.3 Performance Requirements.....	43
4.3.4 Logical Database Requirements.....	43
4.3.5 Design Constraints.....	43
4.3.5.1 Standard Compliance.....	43
4.3.5.2 Hardware Limitations.....	43
4.3.5.3 Reliability & Faulty Tolerance.....	43
4.3.5.4 Security Requirements.....	43
4.3.6 Software System Attributes.....	44
5. PROJECT PLANNING.....	45-55
5.1 Project Scheduling.....	45-46
5.2 Timeline Chart.....	47
5.3 Effort Estimation: FP-Based Computing.....	48-50
5.4 Cost Estimation: COCOMO-II Model.....	51-52
5.5 Risk Analysis.....	53-55
6. DESIGN.....	56-60
6.1 Data Design.....	56-57
6.2 Component Level Design.....	58-60
7. TESTING.....	61-62
8. REFERENCES.....	63
9. ANNEXURES.....	64-74

TABLE OF DIAGRAMS

<u>FIG NO.</u>	<u>DESCRIPTION</u>	<u>PAGE NO.</u>
2.1	Incremental Model	9
3.1	Context level diagram	10
3.2	Level 1 DFD	11
3.3	Level 2 Registration	12
3.4	Level 2 Login	13
3.5	Level 2 Feed & edit details	14
3.6	Level 2 check cycle & symptoms	15
3.7	Level 2 Generate Health charts	16
3.8	Level 2 Recommend Exercises and diet	16
3.9	Level 2 Notify & set Reminders	17
3.10	Level 2 Access to links	17
3.11	Level 2 Share &View stories	18
3.12	Level 2 feedback & rating	18
3.13	Use case diagram	20
3.14	Sequence dig. Registration	29
3.15	Sequence dig. Login	30
3.16	Sequence dig. Feed & edit details	31
3.17	Sequence Dig. check cycle & symptoms	32
3.18	Sequence dig. Generate Health charts	33
3.19	Sequence dig. Recommend Exercises and diet	34
3.20	Sequence dig. Notify & set Reminders	35
3.21	Sequence dig. Access to links	36
3.22	Sequence dig. Share &View stories	37
3.23	Sequence dig. Feedback & rating	38
7.1	Control Flow graph	62

LIST OF TABLES

<u>Table NO.</u>	<u>DESCRIPTION</u>	<u>PAGE NO.</u>
3.1	Data Dictionary	19
5.1	Project Scheduling	47
5.2	Timeline Chart	48
5.3	Value adjustment factor	49
5.4	Weighting Factor IDV	50
5.5	Complex weight for Obj. type	51
5.6	Productivity Rate for Obj. points	51
5.7	Risk management table	55
6.1	Data design-USER	56
6.2	Data design-ADMIN	57
6.3	Data design-FEEDBACK& RATING	57

1. PROBLEM STATEMENT

As we all know ladies share a love-hate relationship with periods. We hate the painful cramps, unbearable bloating and unpredictable mood swings. Nowadays girls are getting aware about the underlying reasons for irregular periods. One key way to understand your health status better with help of your menstrual cycle is to track it. While the traditional method to do so using a calendar or your own estimate isn't bad, using a period tracking app is a more efficient way. 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 our **FEMME-THE PERIOD TRACKING APP** is to not only remind us about the dates of the cycle and regularities but it also pays attention to these functions-

1. The tracking of menstrual cycle-associated factors such as mood swings, symptoms, intake of medication and contraceptives, diet, vaginal discharge during period, food cravings, exercise and give us regular tips according to our situation.
2. A menstrual calendar where period and ovulation dates as well as days on which additional data have been entered by users are highlighted in specific ways and also predict the dates of the coming month.
3. Reminder about menstrual cycle, pills, general (like visiting doctor, health checkups).
4. Access to online forums providing medical information via links and pop-up windows.
5. A feedback option in which the user can give reviews about the app. Moreover, read other women's stories and can share own experiences.

For women, cycle length prediction can be useful to have an idea of when their period might arrive and the possible onset of Premenstrual syndrome (PMS). The value of having an accurate record of their menstrual cycles may be very important for some women, for their fertility, pregnancy and during the perimenopause. Your menstrual cycle is a direct indicator of your overall health, and periods are your body's way of telling you that things are working as they should. Having an extremely unpredictable or heavy period, or skipping a period, can indicate an existing underlying condition. By tracking and logging various details of your cycle, you will be able to recall things that you might otherwise forget when speaking with your healthcare provider. When you use this app, you'll always have the dates of your last period at your fingertips.

2. SOFTWARE LIFECYCLE MODEL

Model best suited for this FEMME is the **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 build. 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.

Advantages of Incremental Model

- o Errors are easy to be recognized.
- o Easier to test and debug.
- o More flexible.
- o Simple to manage risk because it is handled during its iteration.
- o The Client gets important functionality early.

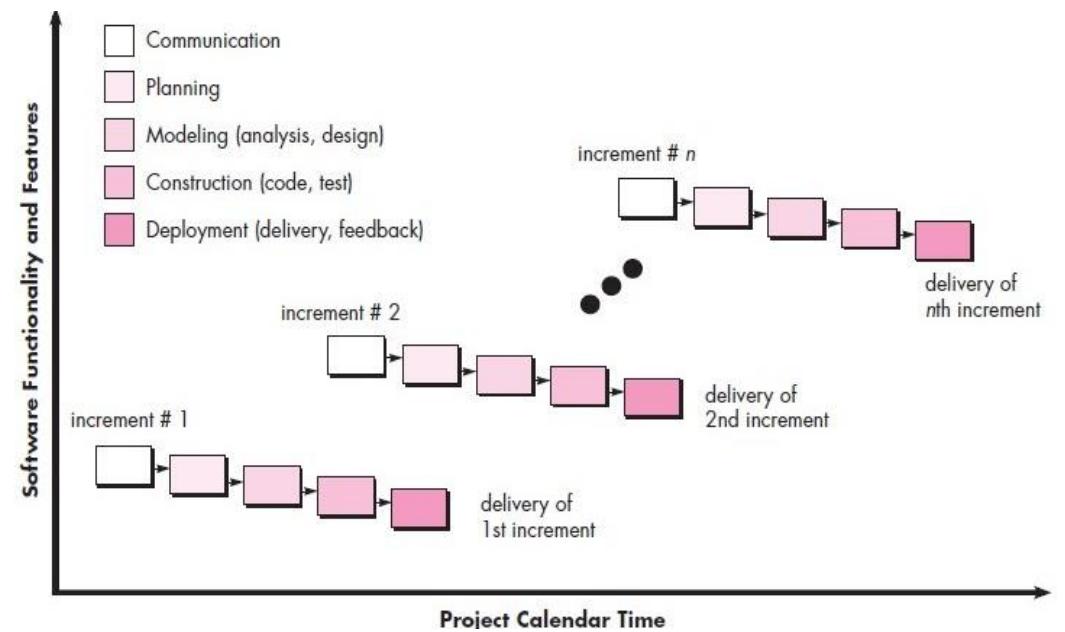


Fig 2.1 Incremental Model

3. REQUIREMENT ANALYSIS

3.1 DATA FLOW DIAGRAMS

3.1.1 CONTEXT LEVEL DIAGRAM

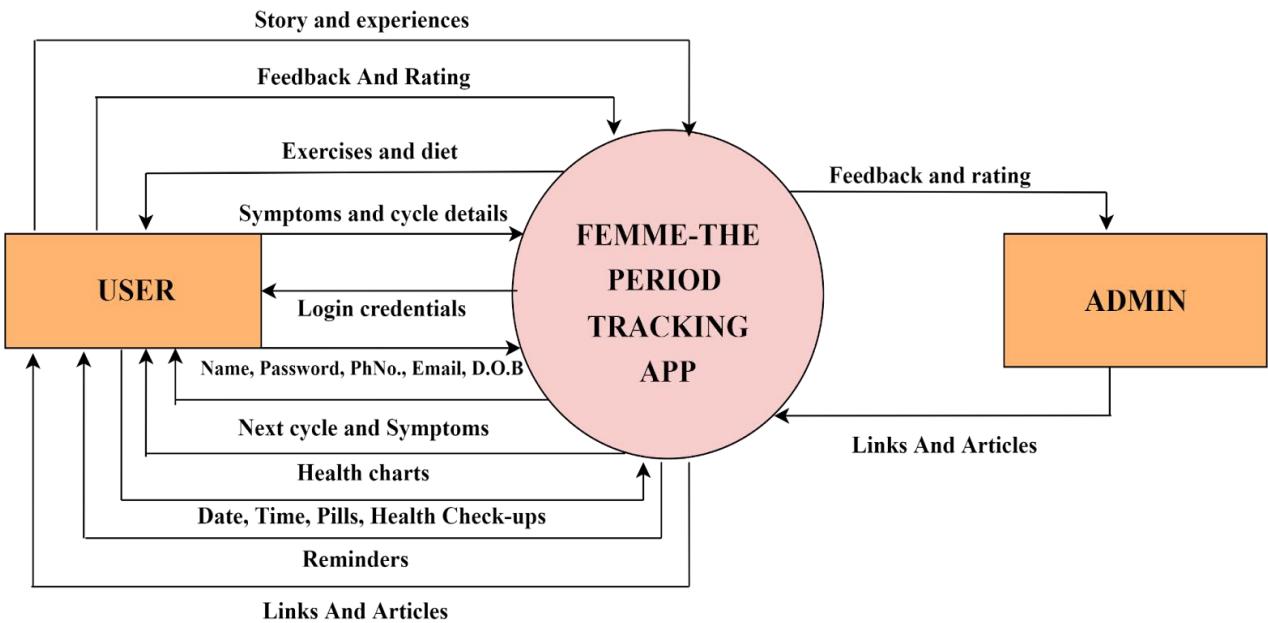


Fig 3.1 Context Level Diagram

3.1.2 LEVEL 1 DFD

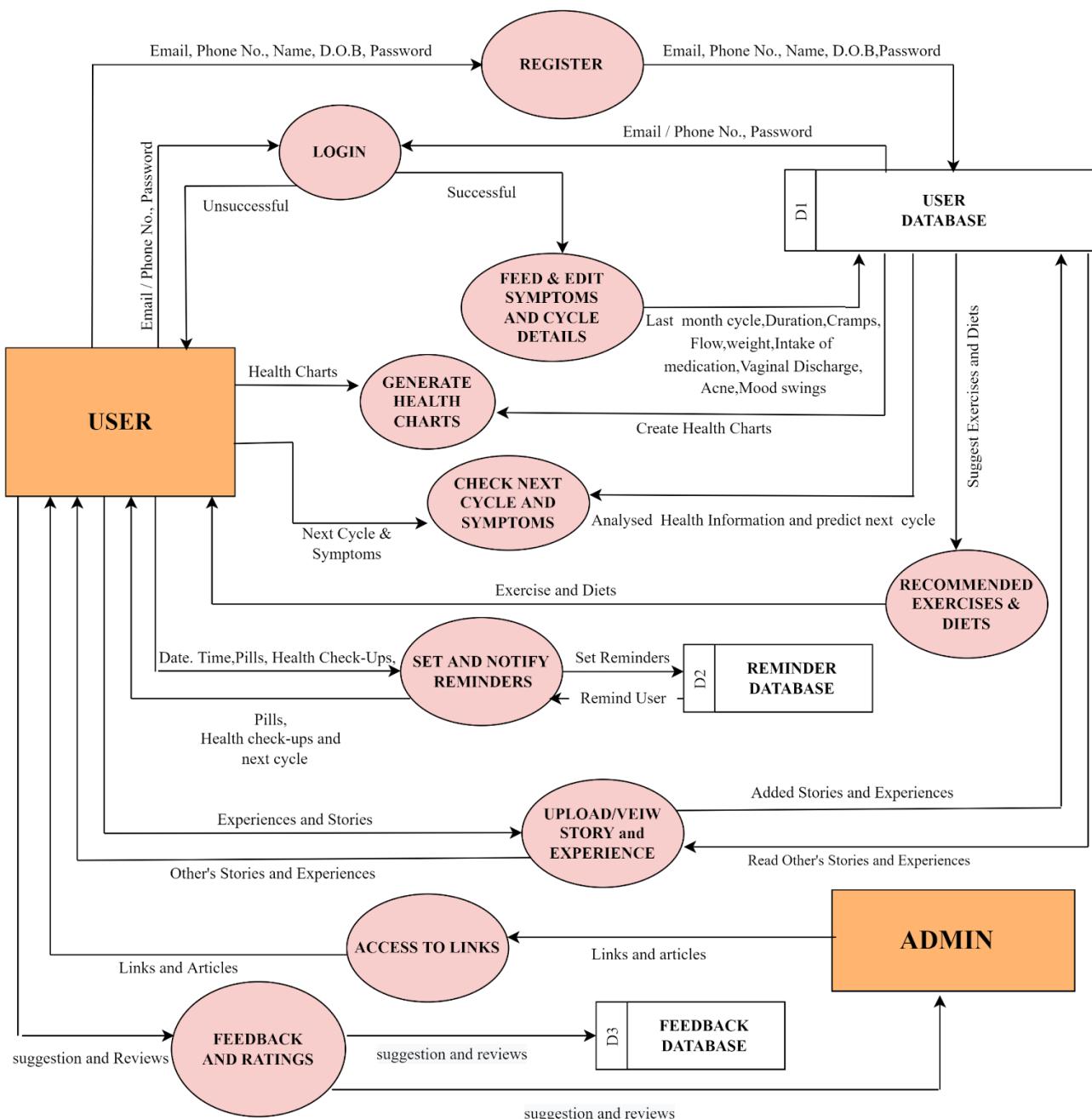


Fig 3.2 LEVEL 1 DFD

3.1.3 LEVEL 2 DFD

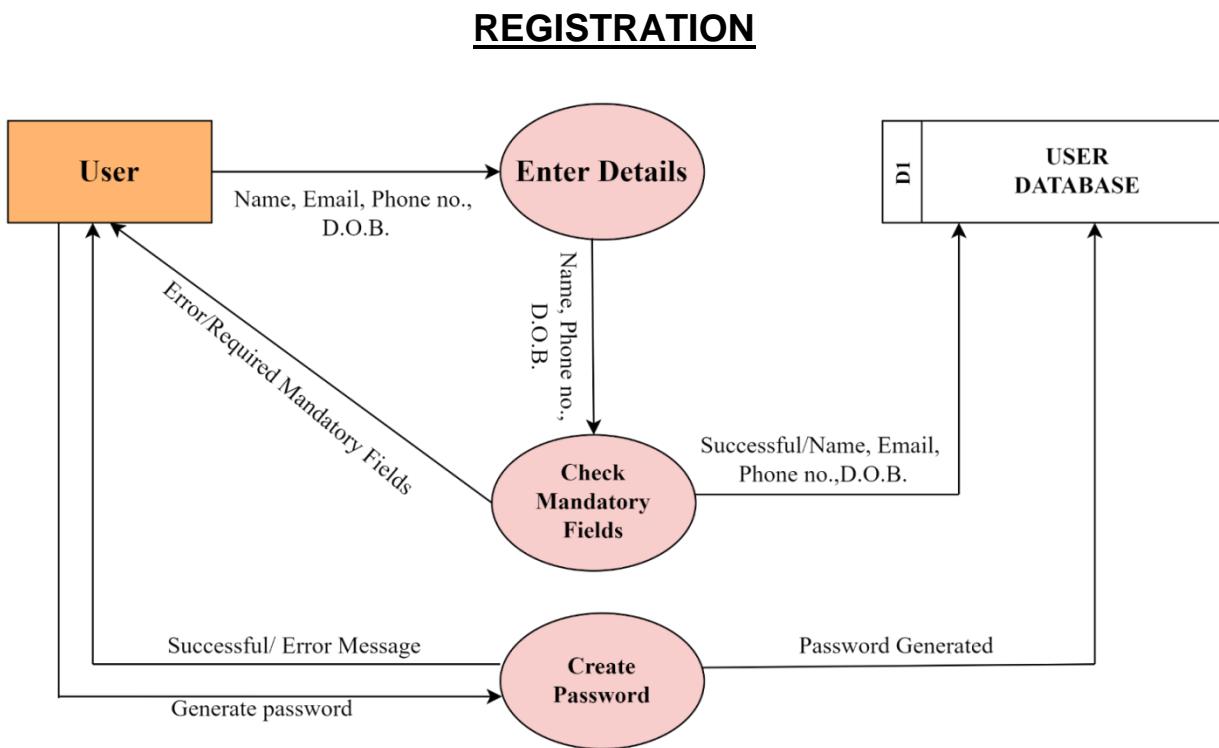


Fig 3.3 LEVEL 2 DFD (REGISTRATION)

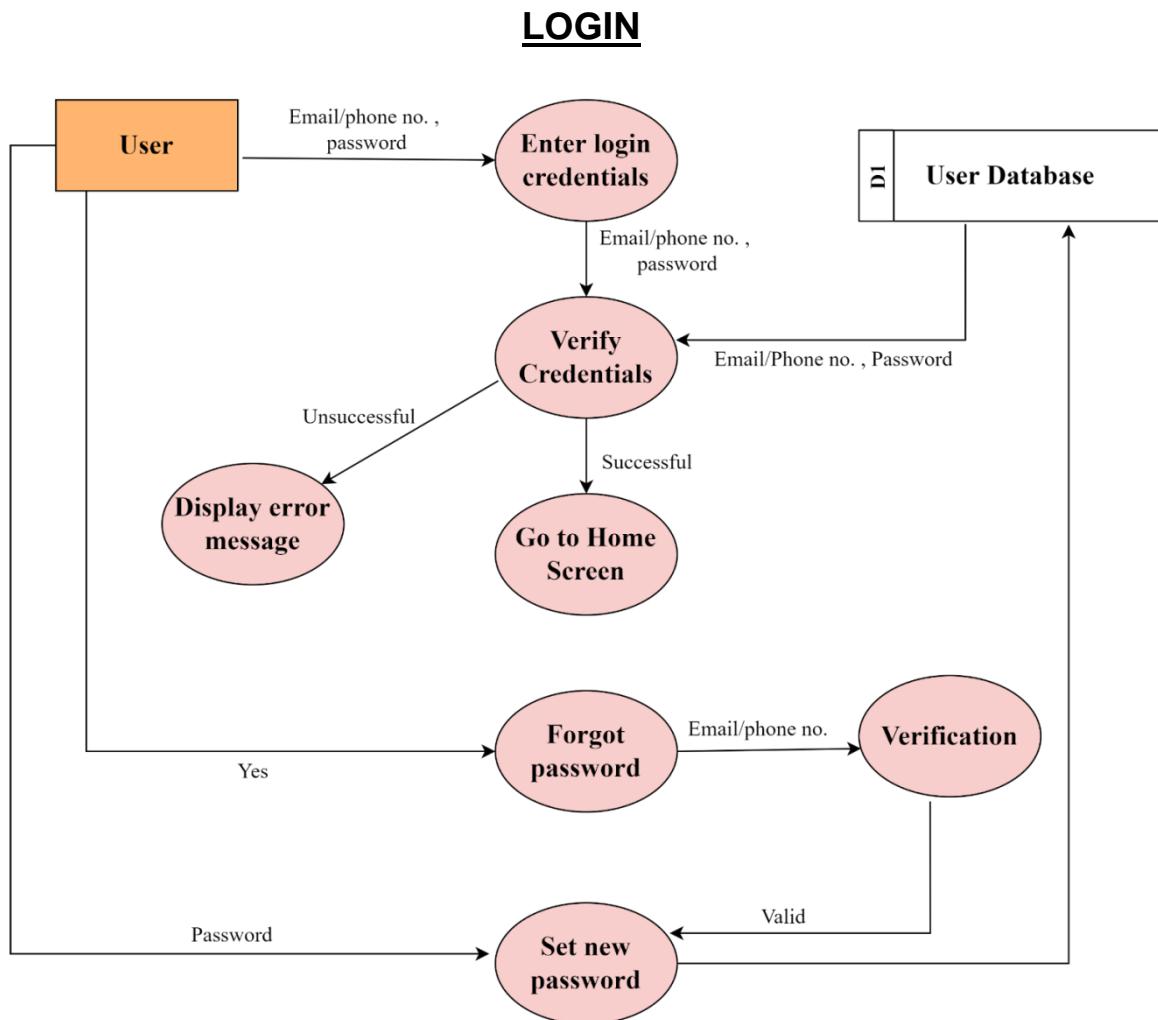


Fig 3.4 LEVEL 2 DFD (LOGIN)

FEED & EDIT SYMPTOMS AND CYCLE DETAILS

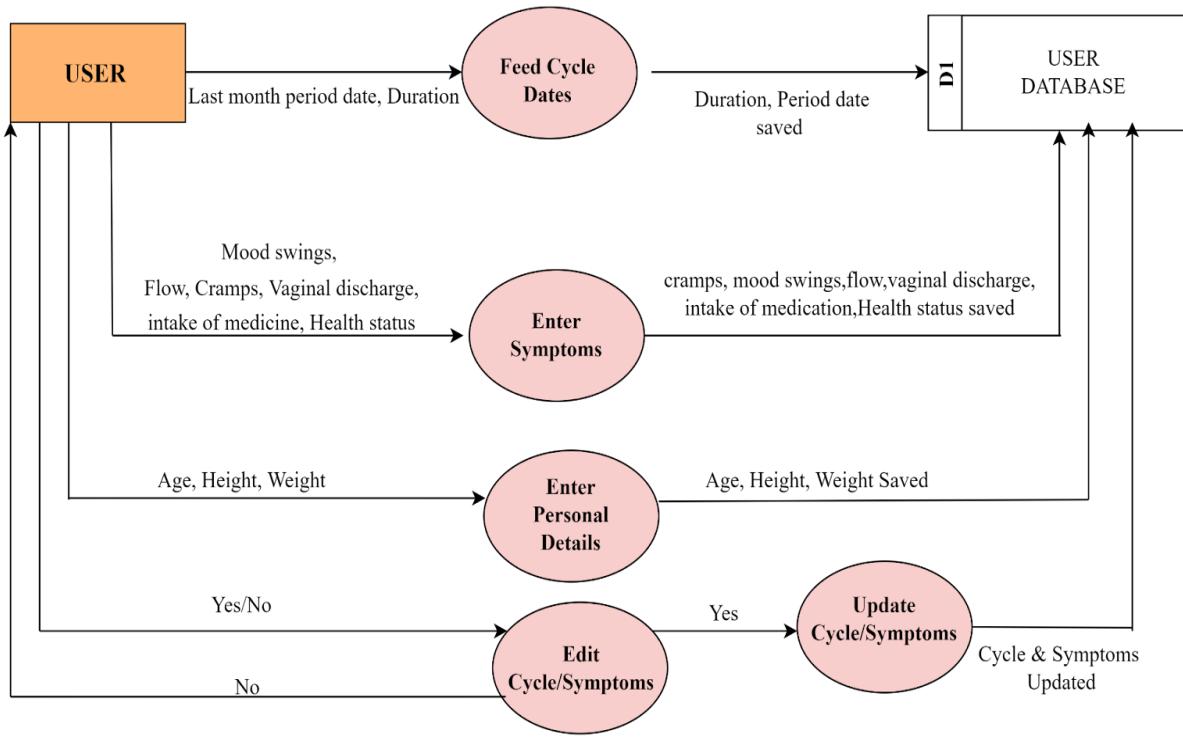


Fig 3.5 LEVEL 2 DFD (FEED & EDIT SYMPTOMS AND CYCLE DETAILS

CHECK NEXT CYCLE AND SYMPTOMS

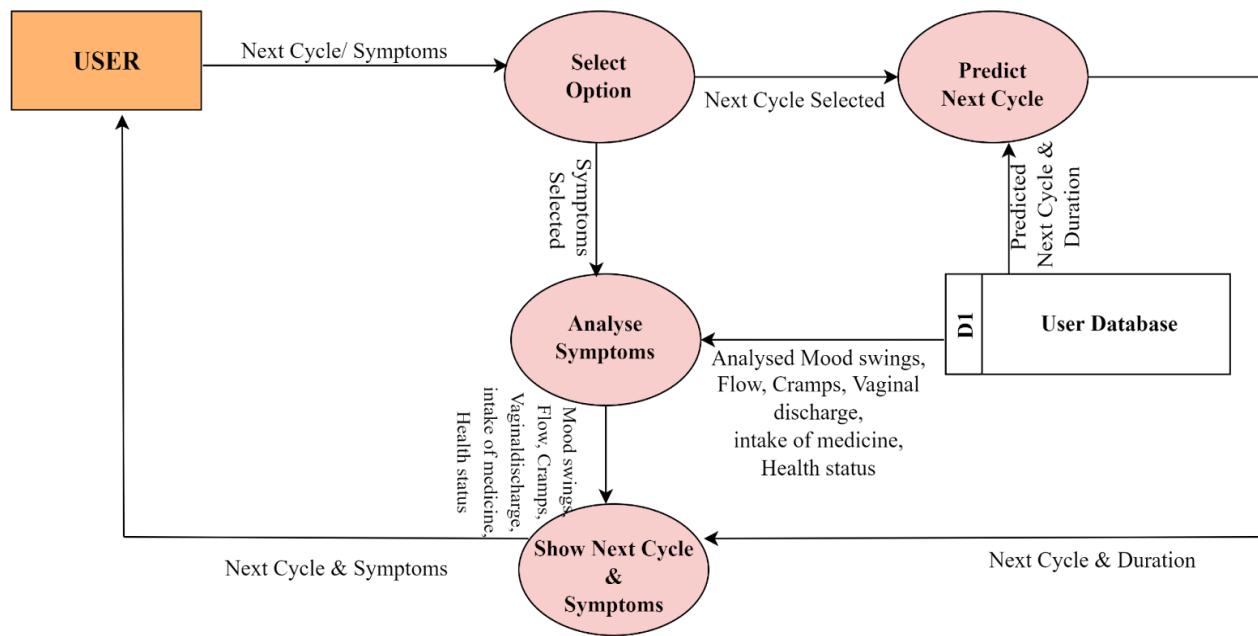


Fig 3.6 LEVEL 2 DFD (CHECK NEXT CYCLE AND SYMPTOMS)

GENERATE HEALTH CHARTS

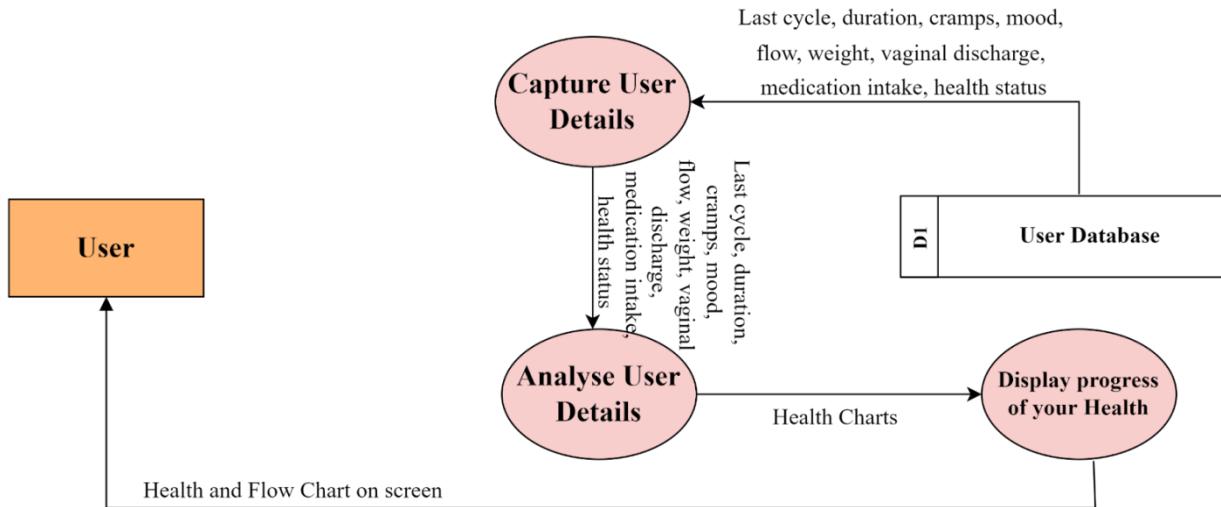


Fig 3.7 LEVEL 2 DFD (GENERATE HEALTH CHARTS)

RECOMMEND EXERCISE AND DIET

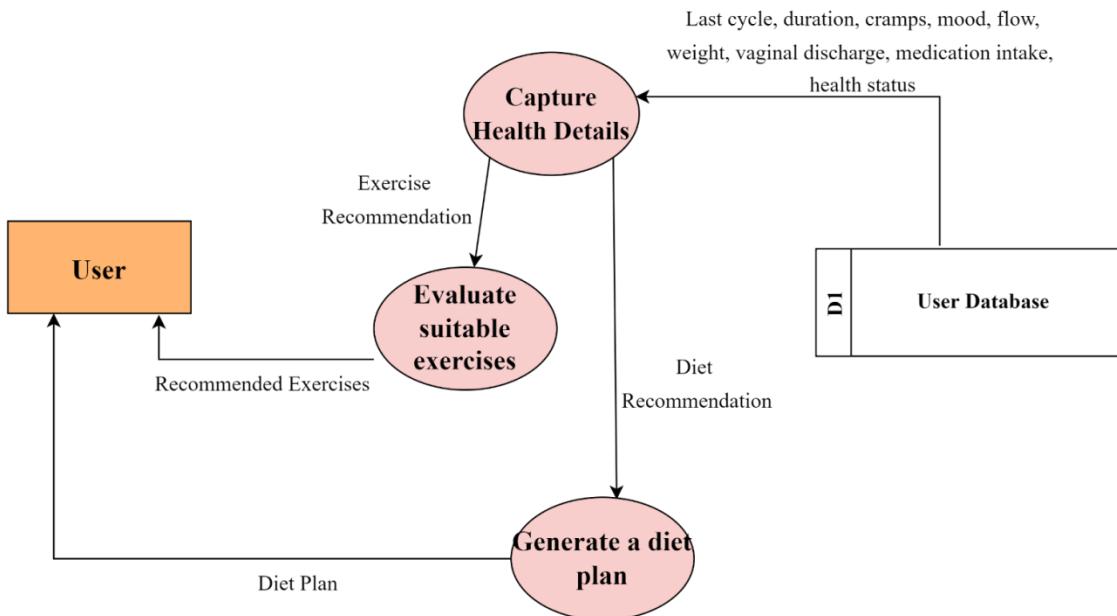


Fig 3.8 LEVEL 2 DFD (RECOMMEND EXERCISE AND DIET)

NOTIFY AND SET REMINDERS

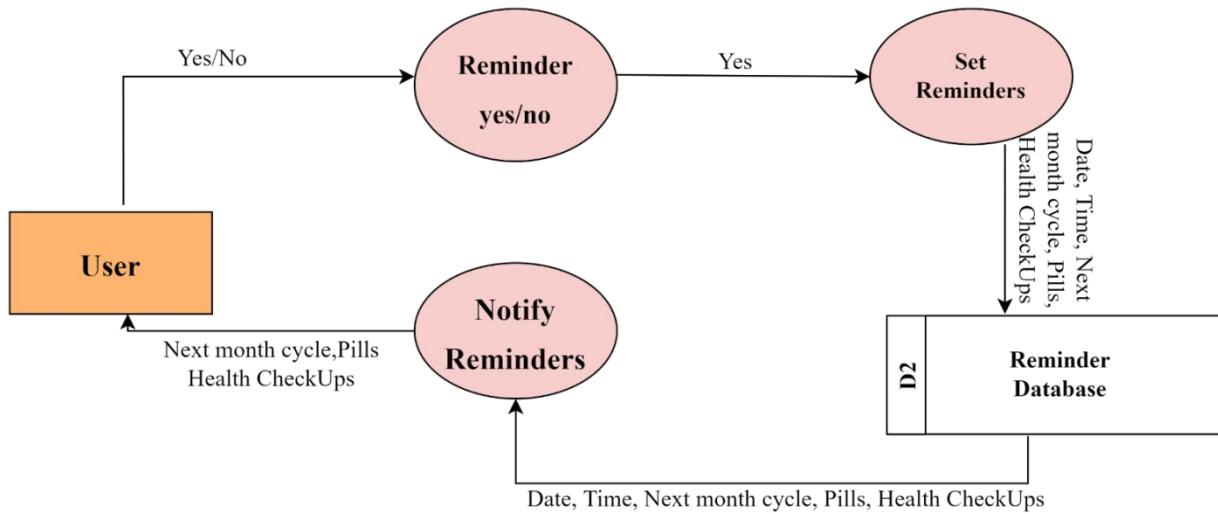


Fig 3.9 LEVEL 2 (NOTIFY AND SET REMINDERS)

ACCESS TO LINKS

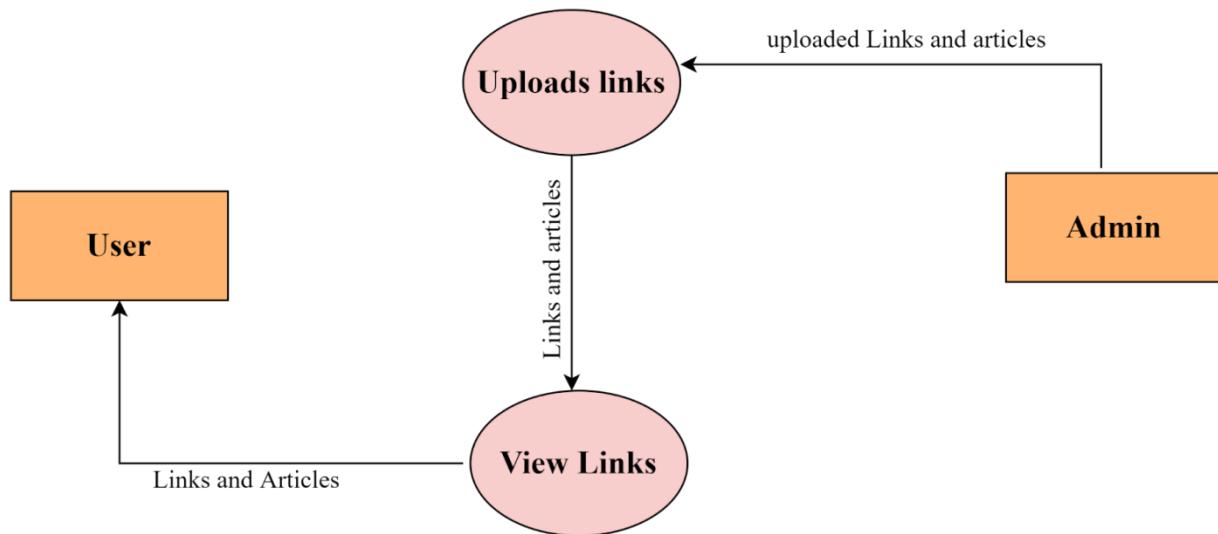


Fig 3.10 LEVEL 2 DFD (ACCESS TO LINKS)

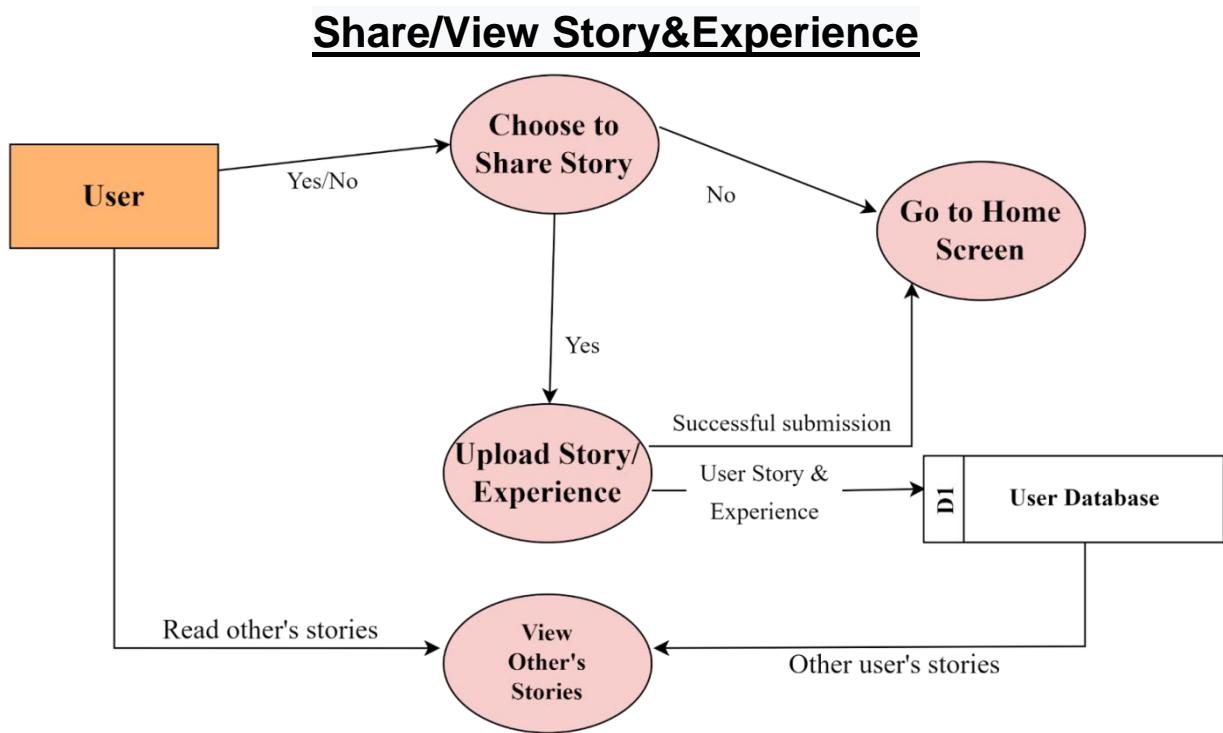


Fig 3.11 LEVEL 2 DFD (SHARE/VIEW STORY AND EXPERIENCE)

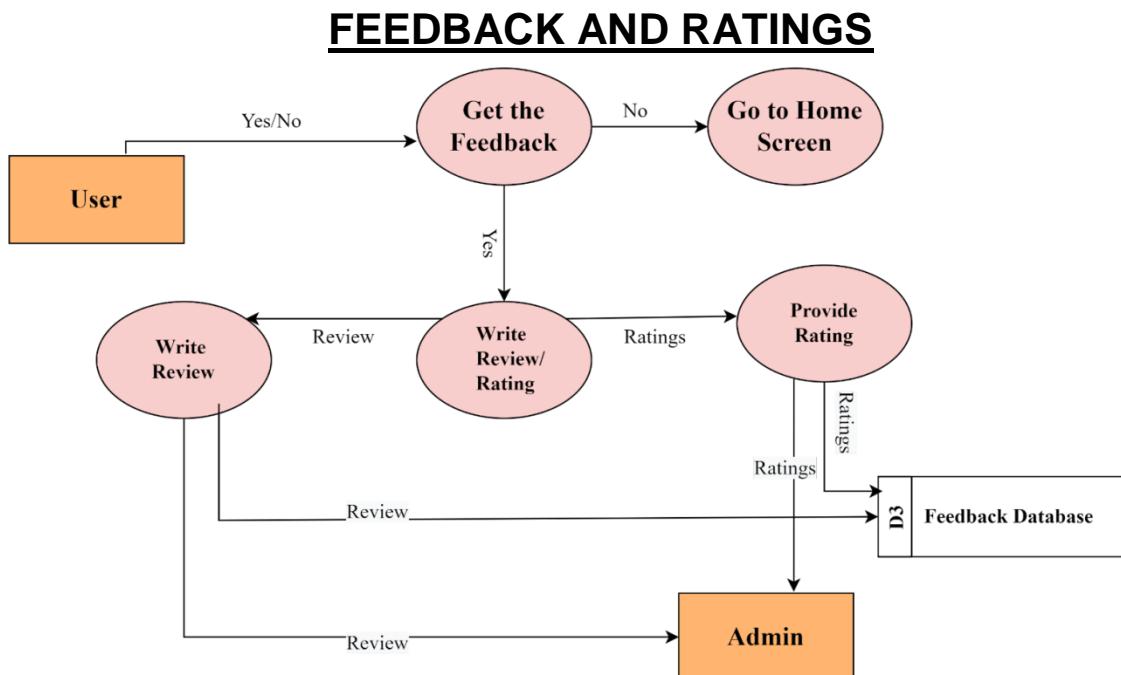


Fig 3.12 LEVEL 2 DFD (FEEDBACK AND RATINGS)

3.2 DATA DICTIONARY

Legal Character: [a-z | A-Z]

Digit: [0-9]

Special Character: [@ | \$ | # | + | - | ,]

1	Name	{Legal Character}*
2	Email	{Legal Character + Digit + Special Character}*
3	Password	{Legal Character + Digit + Special Character}*
4	Phone no.	{Digit +Digit +Digit +Digit +Digit +Digit +Digit +Digit }*
5	D.O.B.	{Digit +Digit +Digit }*
6	Height	{Digit +Digit +Digit +Legal Character+ Legal Character}*
7	Weight	{Digit +Digit +Legal Character+ Legal Character}*
8	Reminders	{Digit +Legal Character}*
9	Symptoms	{Legal Character}*
10	Last Month Period	{Digit +Digit +Digit }*
11	Story/Experience	{Legal Character}*
12	Feedback	{Legal Character}*
13	Ratings	{Digit }*

Table 3.1 DATA DICTIONARY

3.3 USE CASES

3.3.1 USE CASE DIAGRAM

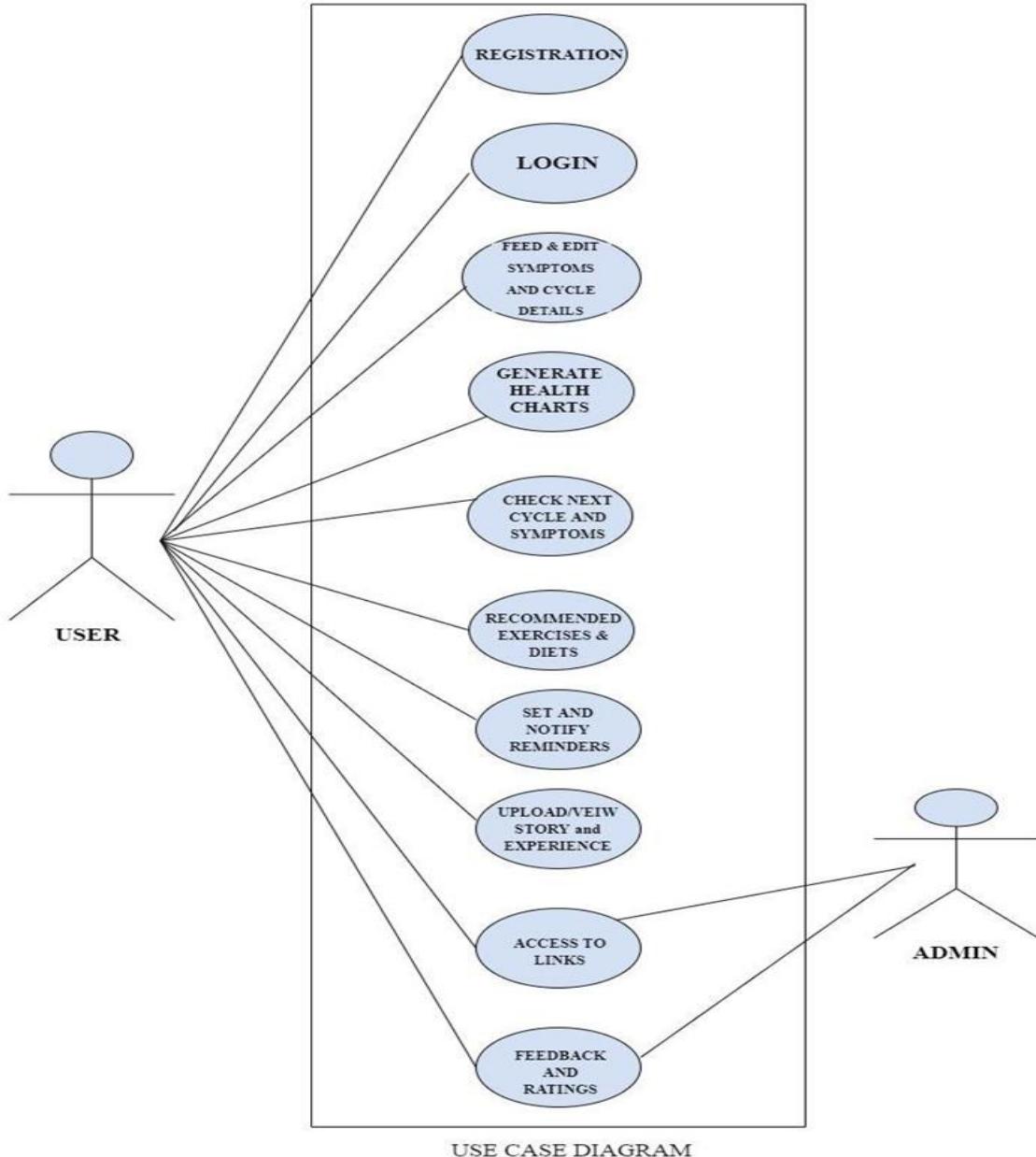


Fig 3.13 USE CASE DIAGRAM

3.3.2 Use Case Description

Actors involved:

1. User
2. Admin
3. System

Use cases:

1. Registration
2. Login
3. Feed and edit details
4. Generate health charts
5. Check next cycle and symptoms
6. Set and notify reminders
7. Recommended exercise and diet
8. Update/view story and experience
9. Access to links
10. Feedback and Rating

Use Case Description for FEMME:

1. Registration

1.	Description: The actors can create a new account and avail the benefits of the app.
2.	Actors: The actors involved in this use case are: <ul style="list-style-type: none">• User
3.	Flow of events: <ul style="list-style-type: none">• Basic Flow:<ul style="list-style-type: none">○ The system asks for email Id, Name, Date of Birth, Phone number.○ The system asks the actor to create a password.○ The system registers the actor in the database.○ The system generates login credentials and displays them.○ The system validates the actor as the user.○• Alternate Flow:<ul style="list-style-type: none">○ If in the basic flow, the actor enters invalid information, the system displays an error message asking the actor to re-enter the information.
4.	Special Requirements:

	<ul style="list-style-type: none"> • None
5.	Pre-condition: None
6.	Post-condition: If the registration process is successful, the user can log in, if not then the use case remains unchanged.

2. Login

1.	Description: The actors can log in by entering the login credentials.
2.	Actors: The actors involved in this use case are: <ul style="list-style-type: none"> • User • Admin
3.	Flow of events: <ul style="list-style-type: none"> • Basic Flow: <ul style="list-style-type: none"> ○ The system asks for an email Id/phone no., password. ○ The system then verifies the details and allows the actor to log in. • Alternate Flow: <ul style="list-style-type: none"> ○ If in the basic flow, the actor does not enter any mandatory details or enters invalid information, the system displays an error message asking the actor to re-enter the information. ○ If the actor is not validated and asks for a change in the password then the system verifies the email ID and asks the actor to set a new password.
4.	Special Requirements: <ul style="list-style-type: none"> • The email Id and phone number should be unique. • The password must be strong. • All the fields are mandatory.
5.	Pre-condition: The actors must have an already existing account in the system. In case of no prior account, the actor must first register themselves and then log into the system.
6.	Post-condition: If the login is successful, the actor is logged into the system. If not, then the use case remains unchanged.

3. Feed and edit details

1.	Description: The actors are asked to enter some details regarding their health and period cycle. The user can delete/update the details.
-----------	---

2.	<p>Actors: The actors involved in this use case are:</p> <ul style="list-style-type: none"> • User
3.	<p>Flow of events:</p> <ul style="list-style-type: none"> • Basic Flow: <ul style="list-style-type: none"> ○ The system asks for last month's cycle, period duration details, other symptoms and personal details. ○ The user enters the details asked. ○ Users can update details. The system retrieves and displays the details. The user can make the desired changes in the details entered. The system then updates the details. • Alternate Flow: <ul style="list-style-type: none"> ○ If in the basic flow, the actor does not enter any mandatory details, the system displays an error message asking the actor to re-enter the information. ○ If in the process of update, the user decides not to update the details then the update is cancelled.
4.	<p>Special Requirements:</p> <ul style="list-style-type: none"> • None
5.	<p>Pre-condition: The actors must be logged into the system.</p>
6.	<p>Post-condition: If the details are submitted successfully, the system creates some health charts and reminders for the user.</p>

4. Check next cycle and symptoms

1.	<p>Description: The user can check the details of their next cycle and other symptoms.</p>
2.	<p>Actors: The actors involved in this use case are:</p> <ul style="list-style-type: none"> • User
3.	<p>Flow of events:</p> <ul style="list-style-type: none"> • Basic Flow: <ul style="list-style-type: none"> ○ The user logs into the system and clicks on the check next cycle option. ○ The system retrieves the information and displays the information for the selected month. ○ The system displays the analyzed user information and the predicted next period cycle of the user and the symptoms. • Alternate Flow: <ul style="list-style-type: none"> ○ None
4.	<p>Special Requirements:</p> <ul style="list-style-type: none"> • The user must have entered the details for the prediction and analyses.

5.	Pre-condition: The actors must have been logged into the system.
6.	Post-condition: None

5 . Generate Health charts

1.	Description: The purpose of this use case is that the users can see the health charts generated for them by the system.
2.	Actors: The actors involved in this use case are: <ul style="list-style-type: none">• User
3.	Flow of events: <ul style="list-style-type: none">• Basic Flow:<ul style="list-style-type: none">○ The user logins in the system and clicks on the health chart option.○ The system retrieves the user details and analyses them to create a health chart for the user.○ Then the health chart generated for the specific user and displays it over the screen.• Alternate Flow:<ul style="list-style-type: none">○ None
4.	Special Requirements: <ul style="list-style-type: none">• The user must have entered the details for the prediction and analyses.
5.	Pre-condition: The actors must have been logged into the system.
6.	Post-condition: None

6. Recommended Exercise and Diet

1.	Description: The purpose of this use case is to recommend the right exercise and diet to the user of the app.
2.	Actors: The actors involved in this use case are: <ul style="list-style-type: none">• User
3.	Flow of events: <ul style="list-style-type: none">• Basic Flow:<ul style="list-style-type: none">○ The user logins in the system and clicks on the Exercise and Diet option.○ The system retrieves the health details and evaluates suitable exercises and

	<p>diet plan, then recommends the exercise and diet for the specific user and displays them over the screen.</p> <ul style="list-style-type: none"> • Alternate Flow: <ul style="list-style-type: none"> ○ None
4.	<p>Special Requirements:</p> <ul style="list-style-type: none"> • The user must have entered the cycle details, flow details, vaginal discharge details.
5.	<p>Pre-condition: The actors must have been logged into the system. The actors must have filled in the required information for the analysis.</p>
6.	<p>Post-condition: None</p>

7. Notify and set reminders

1.	<p>Description: The purpose of this use case is to show reminders to the user about the upcoming period cycle, health checkups, pills, etc.</p>
2.	<p>Actors: The actors involved in this use case are:</p> <ul style="list-style-type: none"> • User
3.	<p>Flow of events:</p> <ul style="list-style-type: none"> • Basic Flow: <ul style="list-style-type: none"> ○ The user can set reminders by clicking on the options set reminder. ○ The system retrieves the Reminders' information from the reminder database. ○ The system notifies the user of the upcoming cycle, health checkups, pills, etc. • Alternate Flow: <ul style="list-style-type: none"> ○ None
4.	<p>Special Requirements:</p> <ul style="list-style-type: none"> • The user must have entered the details of the last few cycles.
5.	<p>Pre-condition: The actors must have been logged into the system. The actors must have filled in the required information for the analysis and reminders.</p>
6.	<p>Post-condition: None</p>

8. Access to links

1.	<p>Description: The purpose of this use case is to display links to online resources and articles</p>
-----------	--

	to the user.
2.	Actors: The actors involved in this use case are: <ul style="list-style-type: none"> • User • Admin
3.	Flow of events: <ul style="list-style-type: none"> • Basic Flow: <ul style="list-style-type: none"> ○ The admin adds the links to online resources and articles for the user. ○ The system displays the links to resources to the user via pop-ups. ○ The user may read them. • Alternate Flow: <ul style="list-style-type: none"> ○ None
4.	Special Requirements: <ul style="list-style-type: none"> • None
5.	Pre-condition: The actors must have been logged into the system.
6.	Post-condition: None

9. Share/View Story and experience

1.	Description: It allows the users to share their own experiences and stories and also read the stories of other users.
2.	Actors: The actors involved in this use case are: <ul style="list-style-type: none"> • User
3.	Flow of events: <ul style="list-style-type: none"> • Basic Flow: <ul style="list-style-type: none"> ○ The user selects the option to share story. ○ The user writes their own story and experience that they want to share with the FEMME community. ○ The system saves the story entered by the user. ○ The user may also read the stories of other users by selecting the read story option. • Alternate Flow: <ul style="list-style-type: none"> ○ None
4.	Special Requirements: <ul style="list-style-type: none"> • None

5.	Pre-condition: The actors must have been logged into the system.
6.	Post-condition: None

10. Feedback and Rating

1.	Description: It allows the users to share their feedback and Ratings for the application.
2.	Actors: The actors involved in this use case are: <ul style="list-style-type: none">• User• Admin
3.	Flow of events: <ul style="list-style-type: none">• Basic Flow:<ul style="list-style-type: none">◦ The user may provide the feedback and rating for the app by selecting the feedback and rating option.◦ The system stores the feedback and rating in the feedback database.◦ The feedback is also sent to the admin of the app.• Alternate Flow:<ul style="list-style-type: none">◦ If in the basic flow, the user decides not to enter the feedback and ratings then the feedback and rating are cancelled.
4.	Special Requirements: <ul style="list-style-type: none">• None
5.	Pre-condition: The actors must have been logged into the system.
6.	Post-condition: If the feedback is successfully given, the feedback is sent to the admin for further improvements.

3.4 SEQUENCE DIAGRAMS

REGISTRATION

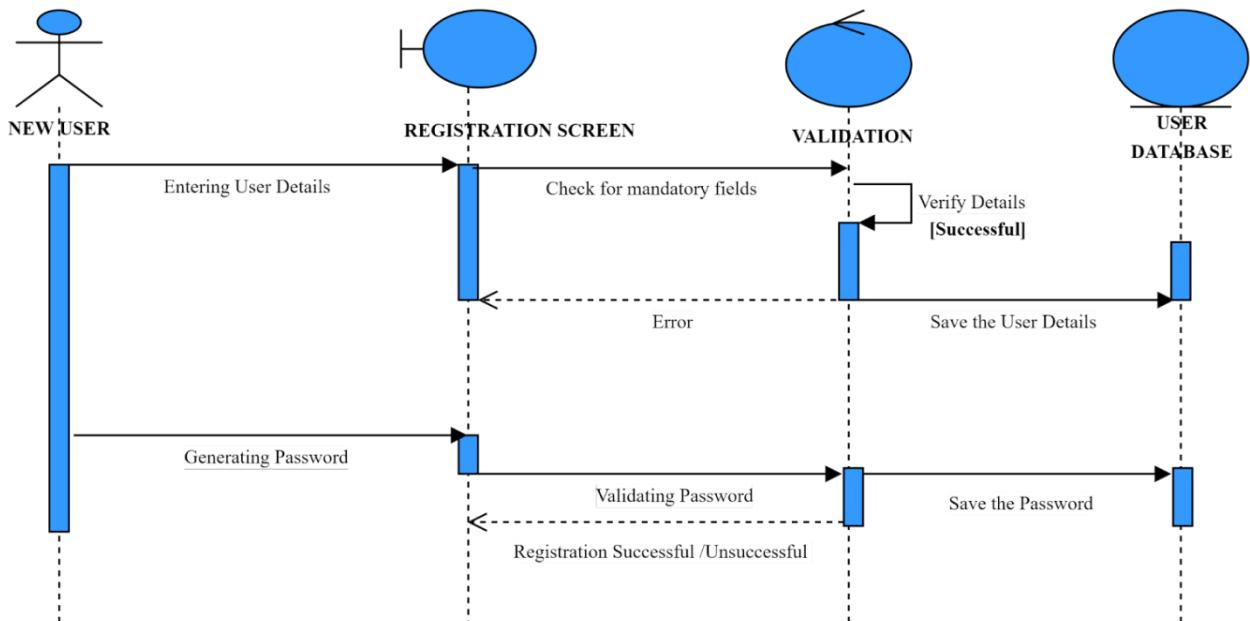


Fig 3.14 Sequence Diagram (Registration)

LOGIN

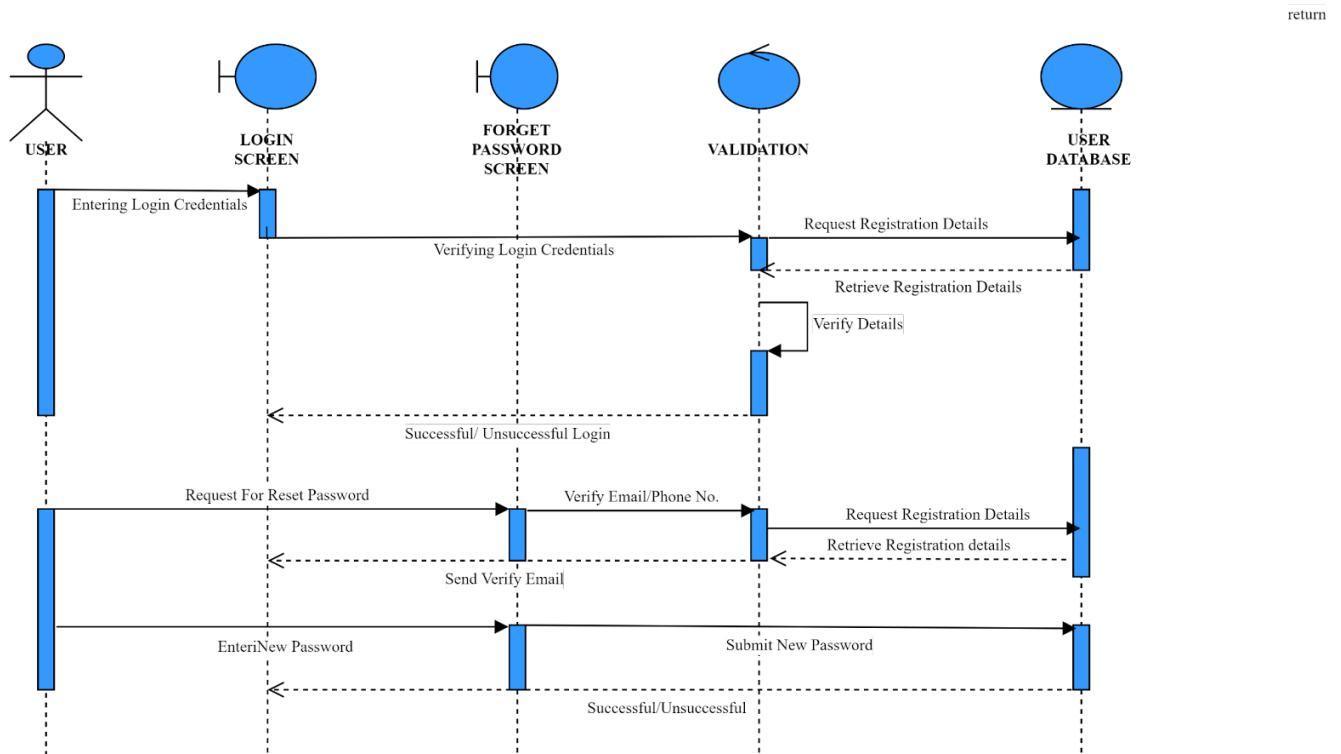


Fig 3.15 Sequence Diagram (Login)

FEED AND EDIT CYCLE & SYMPTOM DETAILS

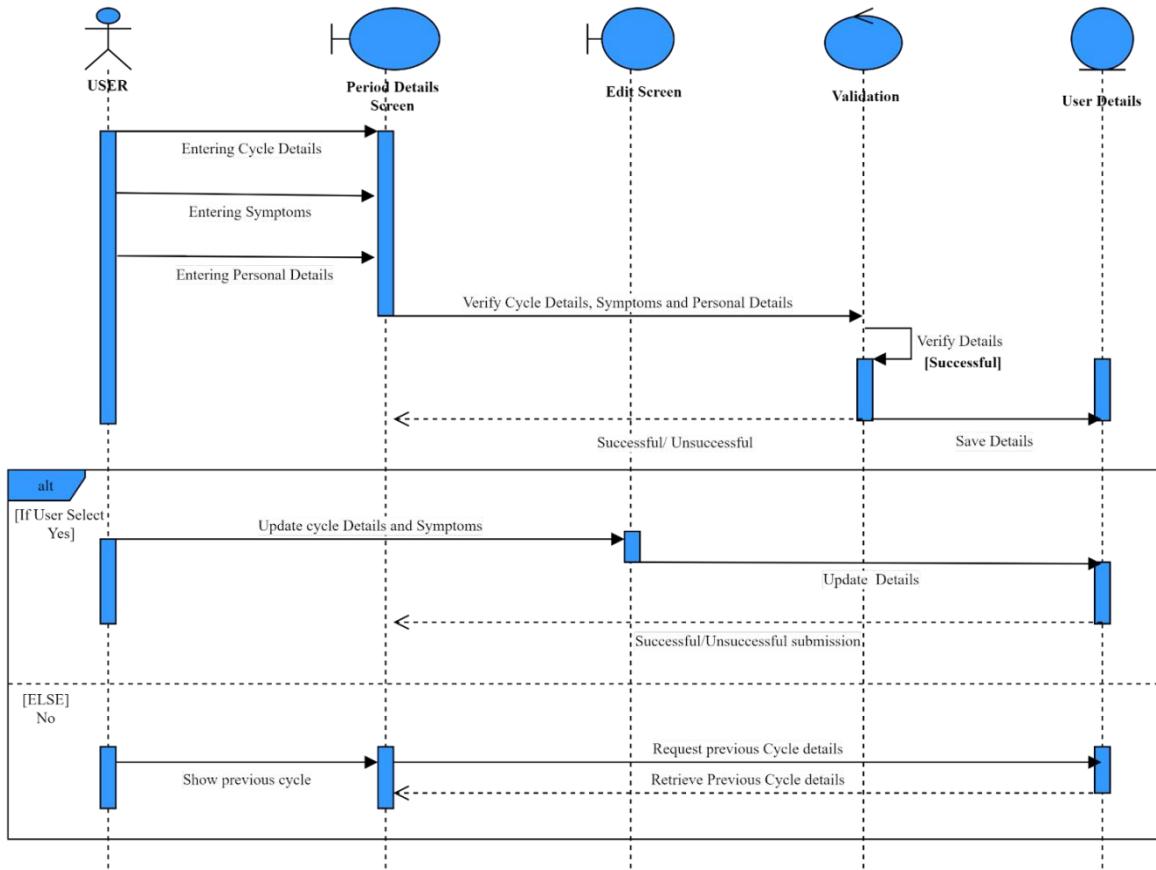


Fig 3.16 Sequence Diagram (Feed and Edit Symptoms & Cycle Details)

CHECK NEXT CYCLE AND SYMPTOMS

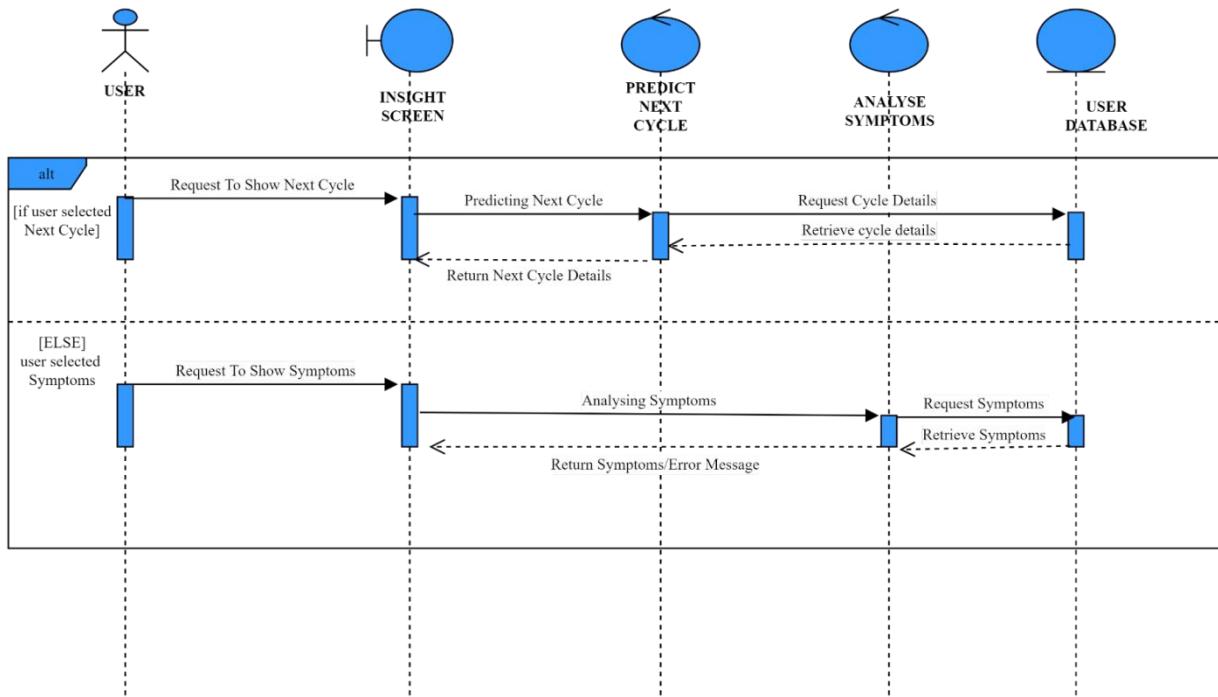


Fig 3.17 Sequence Diagram (Check Next Cycle and Symptoms)

GENERATE HEALTH CHARTS

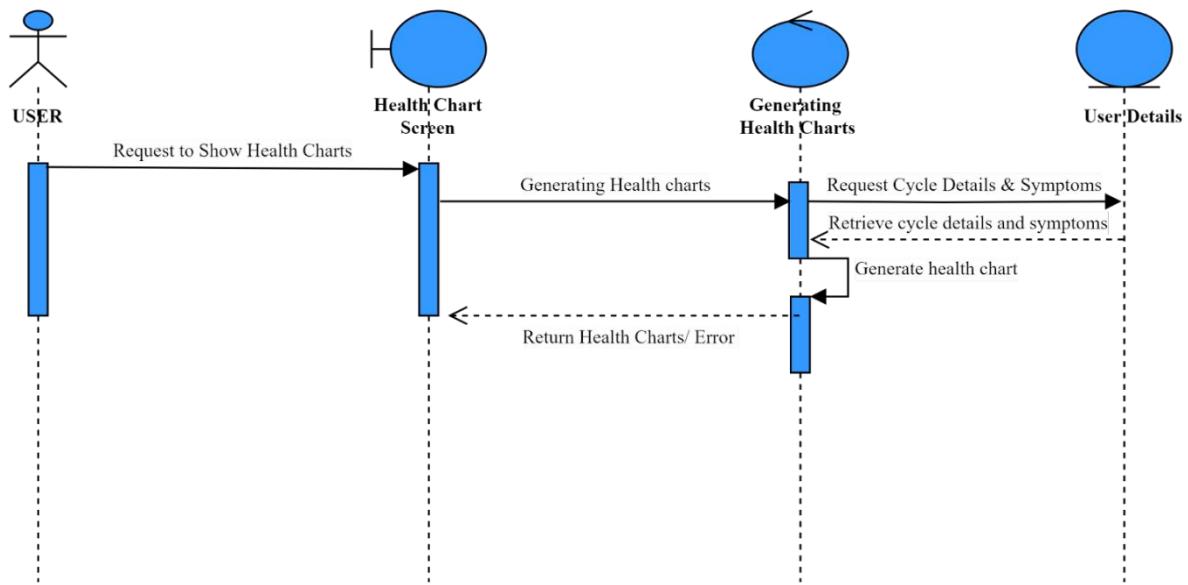


Fig 3.18 Sequence Diagram (Generate Health Charts)

RECOMMEND EXERCISE AND DIET

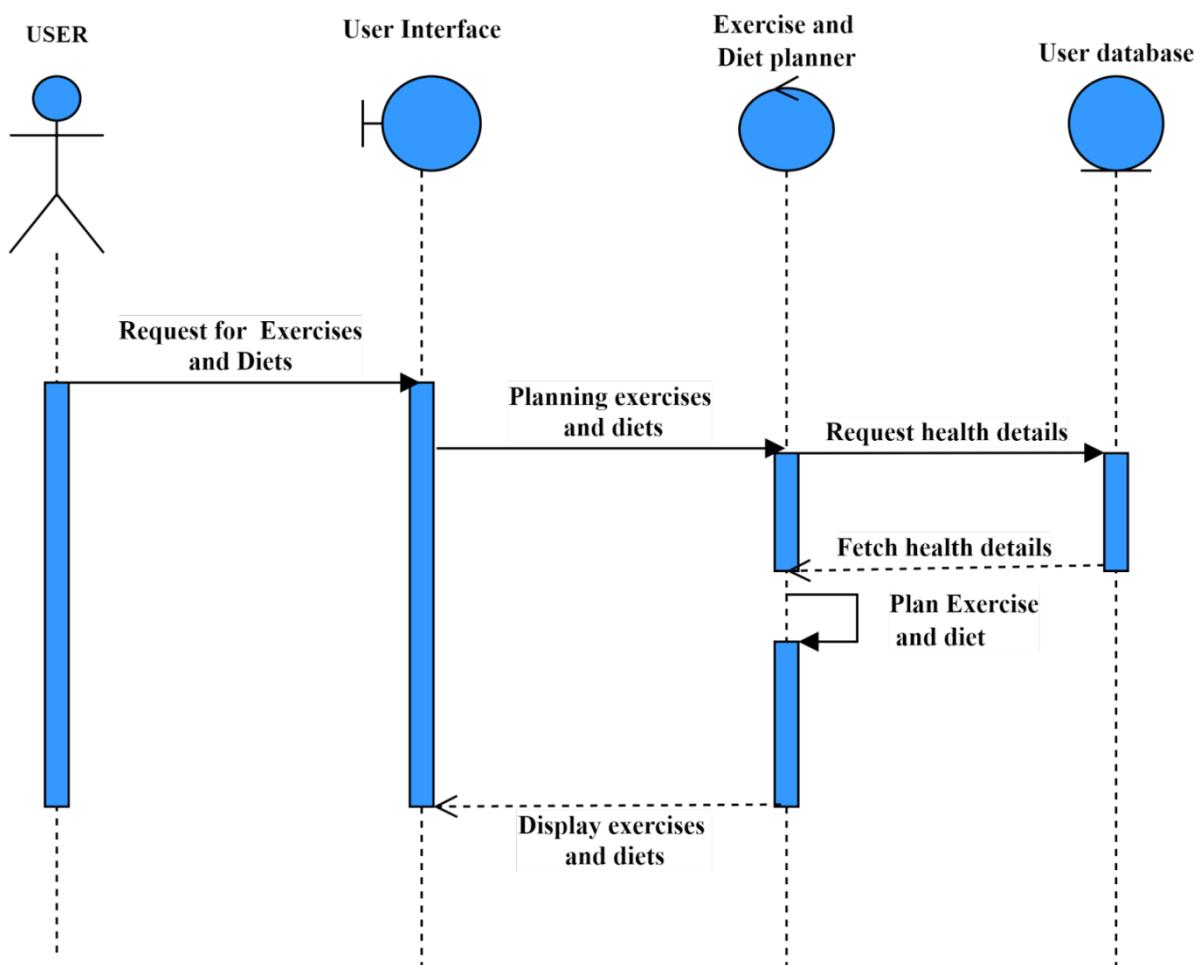


Fig 3.19 Sequence Diagram (Recommend Exercise and Diet)

NOTIFY AND SET REMINDERS

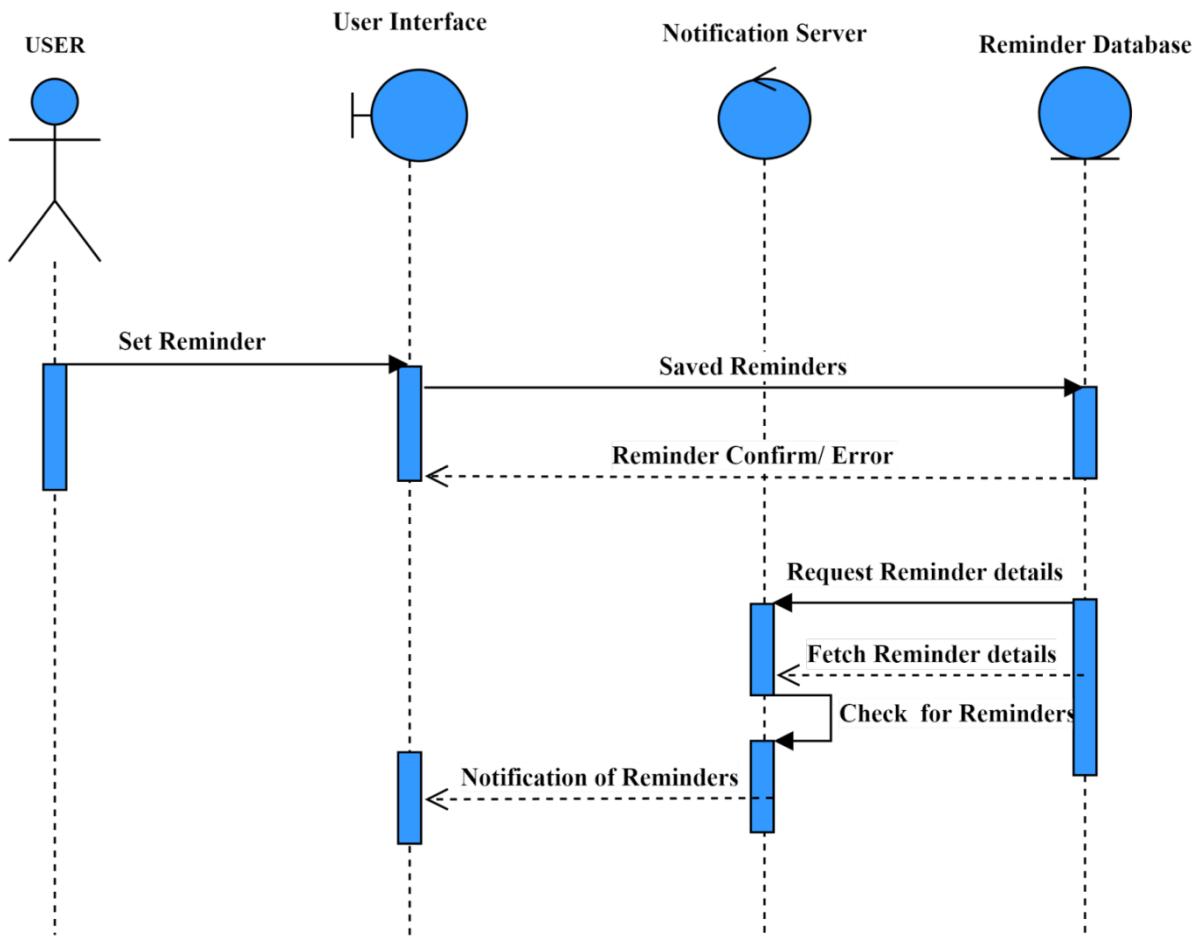


Fig 3.20 Sequence Diagram (Notify and Set Reminders)

ACCESS TO LINKS

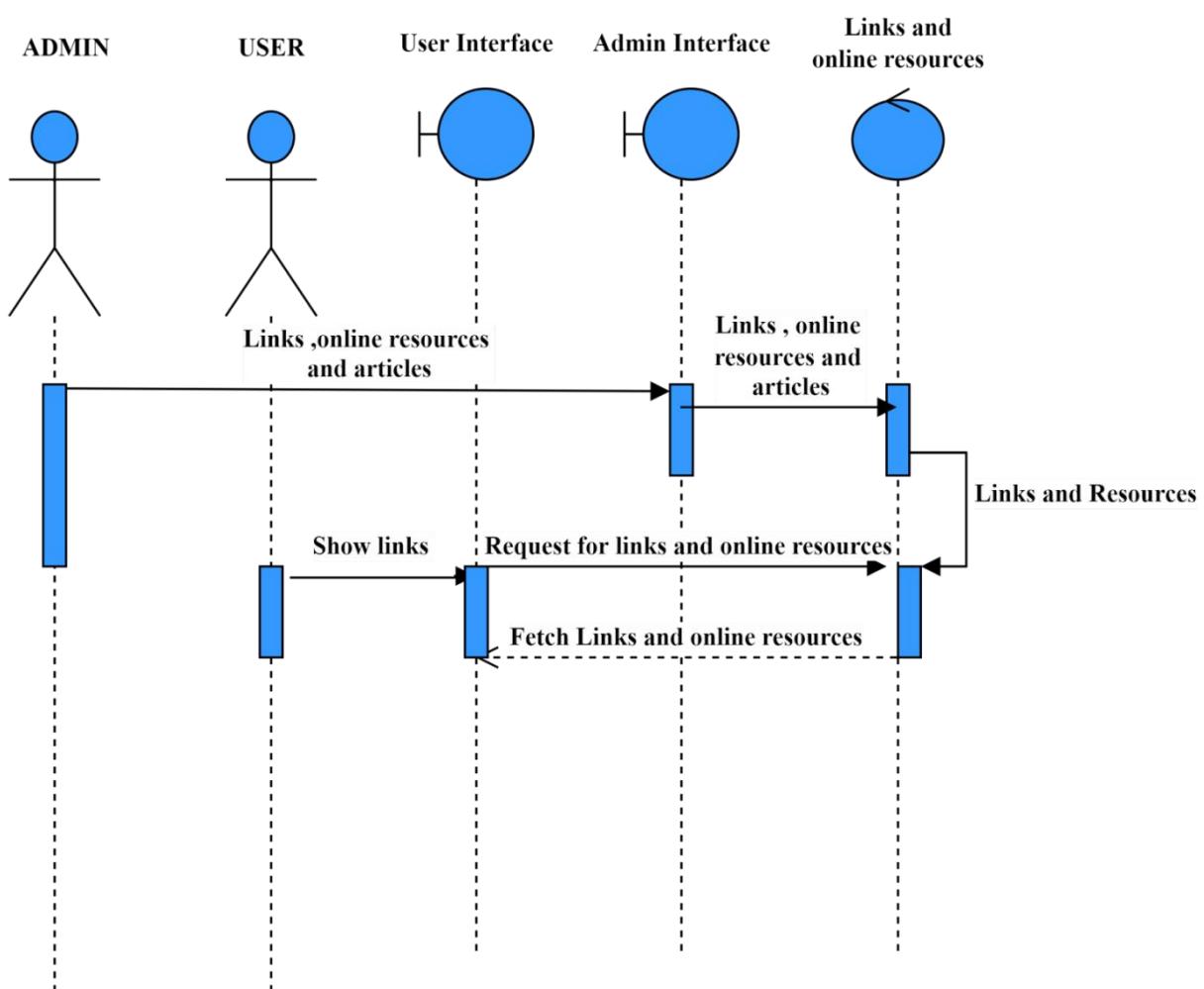


Fig 3.21 Sequence Diagram (Access to Links)

SHARE/VIEW STORY AND EXPERIENCE

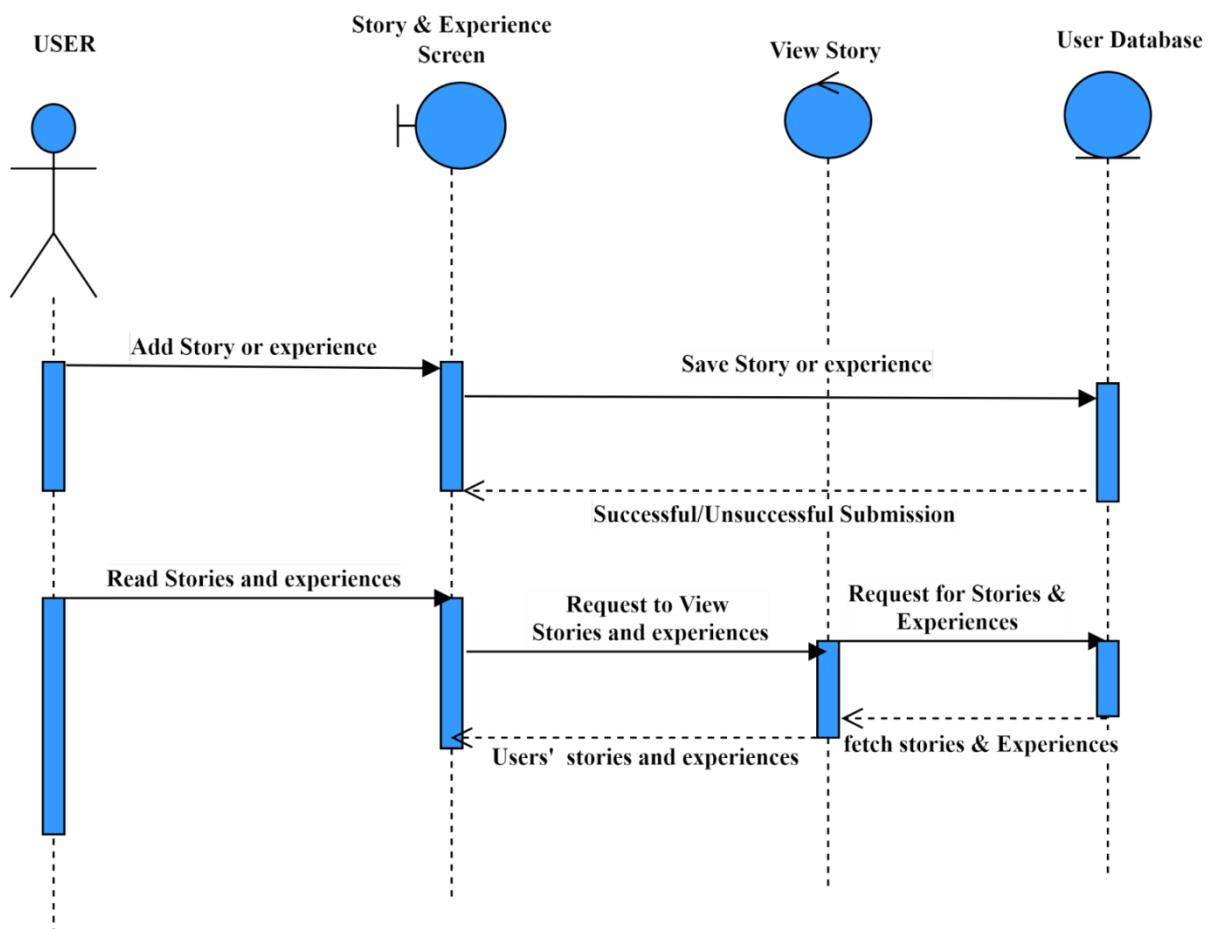


Fig 3.22 Sequence Diagram (Share/View Story and Experience)

FEEDBACK AND RATINGS

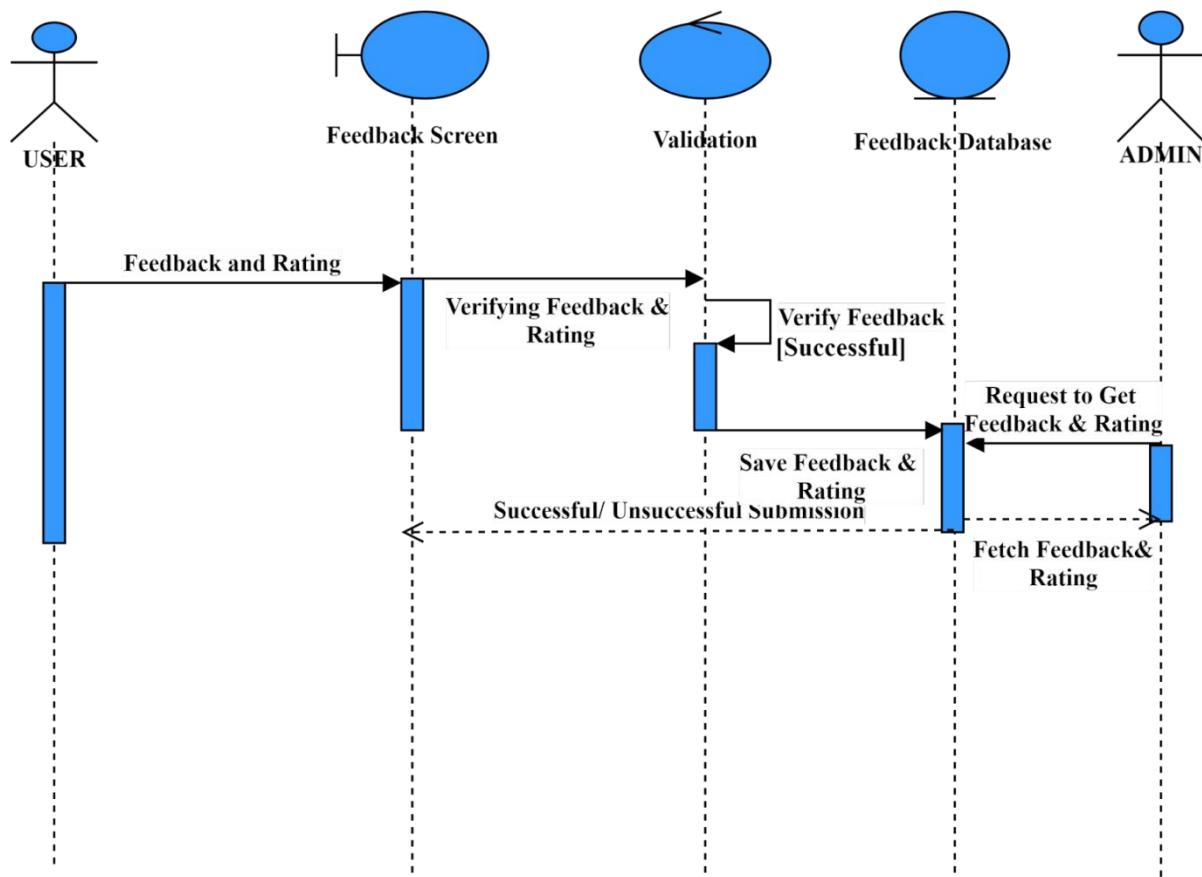


Fig 3.23 Sequence Diagram (Feedback and Ratings)

4. SOFTWARE REQUIREMENTS SPECIFICATIONS

4.1 INTRODUCTION

This document will cover almost all the intended features of the software. It aims to define all the software requirements exhaustively and accurately.

“FEMME: The period-tracking app” is a project built to help women track their period and health without using primitive methods such as a calendar.

The FEMM app is more than just a period tracker: it provides you with cutting edge science that helps you keep track of your health, understand what is going on with your body, flag potential issues and connect with a network of other users to share your experience and story. We’re a new revolution in women’s health!

4.1.1 Purpose

The purpose of this SRS is to fully document the specifications and requirements of the android application, FEMME; which is intended to provide easy data input to track your cycle or period, track physical and emotional symptoms to better diagnose potential health problems, daily reminders, view several charts on a screen to help you understand your patterns, track medications you are currently taking, various view styles let you see and interpret your cycles better: calendar view, chart view and the detailed chart view, which let you see your symptoms, understand your health and the interplay of hormones in your body through the application.

The audience for this SRS will be the women who want the application to keep track of their period and health.

4.1.2 Scope

The objective of the project is to create and implement an android application for FEMME.

Once the user registers she can do the following:

- Login: With the email Id and a unique password the user can log into the application.
- Feed and edit details: The user can feed the last cycle detail and other symptoms they have experienced till now, for the system to analyze these details. The user can also update and change these details.

- Check next cycle and symptoms: Users can check their upcoming cycle and symptoms they may face before and during the cycle.
- Generate Health charts: Users can also view the health charts created for them in the chart view section and the detailed chart view section.
- Exercise and diet recommendation: The application recommends exercise and diet to the users to keep them healthy and fit.
- Set and notify reminders: The users can set reminders and get notified about their upcoming cycle, health checkups, pills and other important details.
- Access to links: Admin will provide links to additional resources for the users.
- View/Update Story and experiences: Users can share their experience and their stories with other women in the FEMME community. They can also read different stories and relate to them.
- Feedback and Rating: With the feedback and ratings provided by multiple users, the app can improvise upon its functionality for the betterment of users.

Furthermore, the user, if logged in as ADMIN, shall be able to upload links to online resources to the user and get feedback from the user to improvise the application.

4.1.3 Definition, Acronyms and Abbreviations

User- Someone who is interacting with the software.

Admin- Person who interacts with the system.

4.1.4 Overview

The rest of this SRS document describes the various system requirements, interfaces, features and functionalities in detail. The other two sections in the SRS are:

1. Overall Description, which will describe major components of the system, interconnection and external interfaces.
2. Specific Requirements, which will describe the functions of actors, their role in the system and constraints.

4.2 THE OVERALL DESCRIPTION

This section describes the general factors that affect the software and its requirements. This section will show how the system interacts with other systems and introduce its basic functionality.

4.2.1 Project Perspective

“FEMME” is a period tracking android application that serves many different purposes. It takes the user information in the user database and analyses and evaluates it. It retrieves the information and displays it to the user in an organized manner. It has a reminder database to remind users about the upcoming cycles and other important details through notifications. It also has a feedback database that takes the user feedback for future considerations.

4.2.1.1 System interfaces

Operating system: Android

Database: SQL

4.2.1.2 Operations

- Admin can change/update the links to online resources and read the feedback provided by the users.
- Users can edit/update history, search and details entered. Users can view the detailed personalized charts made for them. They can also connect with the FEMME community to share their stories and read others’. They can submit feedback and ratings for the application.

4.2.1.3 Site adaptation Requirements

None

4.2.2 Project Functions

The project will be performing the following functions:

- Login
- Registration
- Feed/Edit symptoms and cycle details (last month cycle, flow, general symptoms, duration etc.)

- Check next cycle and symptoms (Prediction of upcoming months and symptoms based on the details)
- Generate health charts
- Recommended Exercises and diet: The software recommends the exercises and diet to the user.
- Set and notify Reminders (doctor's appointment, pills, next month cycle)
- Access to links: The user can access the links provided in the application by the admin.
- Upload/view story and experience: The user can share their story and experience and also read others' stories.
- Feedback
- Rating

4.2.3 User Characteristics

To use the application user must have :

- Educational level: The user must be comfortable with English.
- Experience: The user must be aware of the policies of our application.
- Technical Expertise: The user must be able to interact with the Graphical user interface.

4.2.4 Constraints

- Username and password will be used to identify the users.
- The graphic user interface is only in English.

4.2.5 Assumptions and Dependencies

- The customer is already a registered employee of the company authorized to access the software.
- Hardware never fails.

- All the editing work is done by the assigned user only.
- The software can access only the information already stored in the database.
- All the editing work is done by the assigned user only.

4.3 SPECIFIC R/EQUIREMENTS

This section consists of a detailed description of all inputs and outputs from the software system.

4.3.1 External Interface

The following screens will be provided to access the functionalities of the software. Log in – it allows two types of logins:

1. Admin: Admin can log in to share the links for online resources and read the user feedback.
2. User

User Screen

- Registration: The user needs to register to use the application. The user can log in after registration.
- Log in: The user can log in to use the features provided by the application.
- Feed Symptoms and Cycle detail: to feed the symptoms and cycle details.
- Check next cycle and symptoms: to check the predicted next cycle and symptoms the user may face.
- Generate health charts: The user can check the detailed personalized health charts.
- Recommended exercise and diet: The user can check and follow the recommended exercises and diets.
- Set/Notify Reminder: The user can set reminders and get notified.
- Access to links: The user can click on the links for online resources.
- Share/View Story and experience: The user can share their story and experience, also they can read stories of other users.
- Feedback and Rating: to provide user feedback and rating.

4.3.2 Functions

The functions of the software are enlisted as follows –

- Provides LOGIN to two types of actors: Admin and User.
- Lets the user enter their symptoms and cycle details.
- Saves records of the user's personal details, her account details for future use.
- Allows the user to check their upcoming cycle and symptoms.
- Generates detailed health charts for the user.
- Recommends exercises to the user.
- Sets reminder for the user and notifies her.
- Allow user to share their story, feedback and ratings.

4.3.3 Performance Requirements

- Easy installation on Android devices.
- The system should support 10,000 simultaneous users.
- Graphical user interface in English.
- Allow updating of records by the responsible user(s) within less than a second 80% times.
- Links to online resources should be processed within 10 seconds.
- Reminders should be set within 10 seconds.

4.3.4 Logical Database Requirements

- The system must store all the user details in one database.
- The user reminders are stored in a reminder database and the feedback and ratings are stored in the feedback database.
- The software should manage the database so that it supports all types of mediums and their storage. The data stored can be used as frequently as 20 times an hour. The database should be accessed 90% times in less than a second.

4.3.5 Design Constraints

- Username and password will be used to identify the users.
- The graphic user interface is only in English.

Standards 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.

4.3.6 Software System Attributes

There are a number of quality attributes of software that can serve as requirements. It is important that required attributes be specified so that their achievements can be objectively identified.

- ➔ Reliability: The system syncs frequently to the backup server to avoid the loss of information.
- ➔ Security:
 - ◆ All the information will be password protected.
 - ◆ Maximum 5 consecutive unsuccessful attempts will be allowed to enter the password. User accessibility is ensured in all ways.
- ➔ Availability: The system should always be available, 24X7. If the system crashes, the backup will be available again within 1 hour.
- ➔ Testability: The system should be able to be tested to confirm the performance and to ensure it performs as intended.
- ➔ Portability: The system should run on any Android device.
- ➔ Efficiency: Enough resources will be there to achieve the task efficiently.
- ➔ Flexibility: The system should be flexible enough to provide space to add new features and to handle them conveniently.

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	Jan.w1	Jan.w2	Jan.w2	Parmeet, Vidhi, Tanya	3 person per week
Software Lifecycle Model	Jan.w1	Jan.w1	Jan.w1	Jan.w1	Vidhi	1 person per week
Project Scheduling	Jan.w3	Jan.w3	Jan.w3	Jan.w3	Vidhi, Tanya	2 person per week
Context Level Diagram	Jan.w3	Jan.w3	Jan.w3	Jan.w3	Parmeet	1 person per week
Data Flow Diagram 1	Jan.w3	Jan.w3	Feb.w1	Feb.w1	Parmeet, Tanya, Vidhi	3 person per week
Data Flow Diagram 2	Feb.w1	Feb.w1	Feb.w1	Feb.w1	Parmeet, Tanya	2 person per week
Data Dictionary	Feb.w2	Feb.w2	Feb.w2	Feb.w2	Parmeet, Vidhi, Tanya	3 person per week
Use case	Feb.w2	Feb.w2	Feb.w4	Feb.w4	Vidhi, Parmeet	2 person per week
Software requirement specification	Feb.w3	Feb.w4	Feb.w3	Feb.w4	Tanya	1 person per week
PseudoCode and Code	Mar.w1	Mar.w1	Mar.w2	Mar.w2	Tanya	1 person per week
Effort & cost Estimation (FP model and Cocomo II model)	Mar.w2	Mar.w3	Mar.w3	Mar.w3	Vidhi, Tanya	2 person per week

Risk Analysis	Mar.w2	Mar.w2	Mar.w3	Mar.w3	Parmeet	1 person per week
Data Design	Mar.w3	Mar.w3	Mar.w3	Mar.w4	Tanya	1 person per week
Sequence Diagram	Mar.w3	Mar.w4	Mar.w4	Mar.w4	Parmeet, Tanya	2 person per week
Testing	Mar.w4	Apr.w1	Apr.w1	Apr.w1	Parmeet, Vidhi	2 person per week
References	Mar.w2	Mar.w3	Mar.w3	Mar.w3	Vidhi	1 person per week
Annexure	Mar.w2	Mar.w3	Mar.w3	Mar.w3	Parmeet	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 LIFE CYCLE	■																
PROJECT SCHEDULING		■	■														
USE CASE			■	■													
SRS					■	■	■	■	■								
CONTEXT LEVEL DIAGRAM					■	■											
DATA FLOW DIAGRAM 1					■	■											
DATA FLOW DIAGRAM 2					■	■	■	■	■								
DATA DICTIONARY							■	■	■	■	■	■	■	■	■	■	■
PROJECT METRICS								■	■	■	■	■					
EFFORT & COST ESTIMATION									■	■	■	■	■	■	■	■	■
RISK ANALYSIS									■	■	■	■	■	■	■	■	■
DATA DESIGN									■	■	■	■	■	■	■	■	■
COMPONENT DESIGN									■	■	■	■	■	■	■	■	■
TESTING									■	■	■	■	■	■	■	■	■

Table 5.2: TIMELINE CHART

5.3 EFFORT ESTIMATION & 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:

$$\text{FP} = \text{Count total} \times [0.65 + 0.05 \times \sum (F_i)]$$

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 (ΣF_i)		49

Table 5.3: VALUE ADJUSTMENT FACTOR(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)	26	3	4	6	78
External Outputs(EO)	24	4	5	7	96
External Queries(EQ)	2	3	4	6	6

Internal Logical Files(ILF)	14	7	10	15	98
External Interface Files(EIF)	1	5	7	10	5
TOTAL COUNT					283

Table 5.4: WEIGHTING FACTOR OF INFORMATION DOMAIN VALUE

$$\text{FUNCTION POINT (FP)} = \text{Total Count} \times [0.65 +$$

$$(0.01 \times \sum (F_i))] = 283 \times [0.65 + (0.01 \times 49)]$$

$$= 283 \times 1.14$$

$$= 322.62$$

$$= 323$$

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) = $(21*2)+(0*5)$

= 42

0% of the components are re-usable-

NOP = New Object Points (or Object Point Counts)

$$= (\text{Object Points}) * [(100\% \text{ of reuse}) / 100]$$

$$= 42 * [(100-0) / 100]$$

$$= \mathbf{42}$$

The developer's experience and capability in a similar environment is low. PROD (Productivity Rate) = 13

Estimated Effort = NOP/PROD = 42/13

$$= 3.2307692$$

$$= \mathbf{3.23 \text{ PM}}$$

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 **3.23 PM** which is according to **21 screens** only.

5.5 RISK ANALYSIS

- **Software Risks**

Risk always involves two characteristics: uncertainty—the risk may or may not happen and loss—if the risk becomes a reality, unwanted consequences or losses will occur.

- **Phases involved in Risk analysis and management**

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

- **Types of Risk**

Different categories of risks are:

1. Project risks
2. Technical risks
3. Business risks

Another general categorization of risks is:

1. Known risks
2. Predictable risks
3. Unpredictable risks

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 problems, implementation problems, interface problems, verification problems and maintenance problems. 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 managers 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	PROBAB-ILITY	IMPACT	RMMM PLAN
1	Customer will change the requirements.	Project Size	80%	Critical	<p>Conduct multiple reviews so that the requests are well understood.</p>
					<p>Set a deadline for proposing changes after which changes proposed would be chargeable.</p>

2	Size estimate may be significantly 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 & Experience	30%	Critical	Appointment of experienced staff. Training of existing staff.
5	Lack of training on tools.	Development Experience	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 Data Design

Field Name	Type	Specifications	Constraints	Unique	Description
User ID	Varchar	4 alphanumeric characters	NOT NULL, PRIMARY KEY	Yes	This provides ID to the user.
Email	Varchar	10 alphanumeric characters	NOT NULL	Yes	Email must contain special characters
Password	Varchar	10 alphanumeric characters	NOT NULL	Yes	Password must have atleast 8 characters
Phone no.	Integer	10 Integer characters	NOT NULL	Yes	Phone number must have 10 digits
Name	Character	15 alphabetic characters	NOT NULL	No	User's Name
Date of birth	Date	MM/DD/YY' format	NOT NULL	No	User's date of birth
Height	Integer	2 integer character	NOT NULL	No	User's height
Weight	Integer	2 integer character	NOT NULL	No	User's weight

Table 6.1 Data Design - User

Field Name	Type	Specifications	Constraint	Unique	Description
admin_id	alphanumeric	6 alphanumeric characters	Not Null, Primary key	Yes	This provides an id to an admin
Admin password	Alphanumeric	10 alphanumeric characters	Not Null	Yes	Admin's password must contain atleast 8 alphanumeric characters
Admin name	Character	10 alphanumeric characters	Not Null	Yes	Admin's name

Table 6.2 Data Design - Admin

Field Name	Type	Specifications	Constraint	Unique	Description
Feedback	String	100 alphabetic characters	Not Null	No	The user can write their feedbcak
Rating	Integer	5 integer characters	Not Null	No	The user can provide rating

Table 6.3 Data Design - Feedback and Rating

6.2 Component level design

Pseudocode for Feed and edit Cycle details and symptoms

1. Feed_edit_details() procedure begins
2. Choose an option to perform.
- 3.If (Option Selected is Feed details) then
4. Enter cycle details, symptoms, personal details
5. if(details verified)
6. display("successful submission")
7. Else display("unsuccessful submission")
- 8.EndIf
- 9.EndIf
- 10 If(option Selected is Edit details) then
- 11 Enter Edited details
- 12 if(successful updation)
- 13 display("successfully updated")
- 14 Else
- 15 display("Error message")
- 16 EndIf
- 17 EndIf
- 18 Feed_edit_details procedure ends

6.3 Code for SET AND NOTIFY REMINDERS

```
import datetime
def gettime():
    return datetime.datetime.now()

def setReminder(c):
    if c==1:
        print("SET REMINDER FOR")
        print("1.NextCycle")
        print("2.Pills")
        print("3.Health CheckUps")
        print("Your choice-")
        d=int(input())
        if d==1:
            time=input("SCHEDULE TIME: ")
```

```

date = input("SCHEDULE DATE: ")
with open("NextCycle.txt", "a") as Nc:
    Nc.write(str(gettime()) + ":" + "SCHEDULE ON TIME: " + time + " SCHEDULE
ON DATE: " + date + "\n")
    print("SUCCESSFULLY ENTERED")
elif d == 2:
    time = input("SCHEDULE TIME: ")
    value = input("PILLS DATA: ")
    with open("Pills.txt", "a") as pl:
        pl.write(str(gettime()) + ":" + "SCHEDULE ON TIME: " + time + " PILLS TO
TAKE: " + value + "\n")
    print("SUCCESSFULLY ENTERED")
elif d == 3:
    time = input("SCHEDULE TIME: ")
    date = input("SCHEDULE DATE: ")
    value = input("DOCTOR'S APPOINTMENT: ")
    with open("HealthCheckUps.txt", "a") as Hc:
        Hc.write(str(gettime()) + ":" + "SCHEDULE ON TIME: " + time + " SCHEDULE ON
DATE: " + date + " APPOINTMENT DATA: " + value + "\n")
    print("SUCCESSFULLY ENTERED")
else:
    print("WRONG INPUT")

def notifyReminder(c):
    if c== 2:
        print("NOTIFY REMINDER FOR")
        print("1.NextCycle")
        print("2.Pills")
        print("3.Health CheckUps")
        print("Your choice-")
        d = int(input())
        if d==1:
            with open("NextCycle.txt") as Nc:
                for i in Nc:
                    print(i, end="")
        elif d==2:
            with open("Pills.txt") as pl:
                for i in pl:
                    print(i, end="")
        elif d == 3:
            with open("HealthCheckUps.txt") as Hc:
                for i in Hc:
                    print(i, end="")

```

```
else:  
    print("WRONG INPUT")  
  
print(".....SET AND NOTIFY REMINDERS.....")  
print("Enter Your Choice!")  
print("1.SET REMINDER")  
print("2.NOTIFY REMINDER")  
print("Your choice-")  
a=int(input())  
if a==1:  
    setReminder(a)  
elif a==2:  
    notifyReminder(a)  
else:  
    print("INVALID INPUT")
```

7. TESTING

We will perform white box testing on the module Feed/Update cycle details and symptoms.

White Box Testing is a software testing technique in which the internal structure, design and coding of software are tested to verify the flow of input-output and to improve design, usability and security. In white-box testing, code is visible to testers so it is also called Clear box testing, Open box testing, transparent box testing, Code-based testing and Glass box testing.

Control Flowgraph

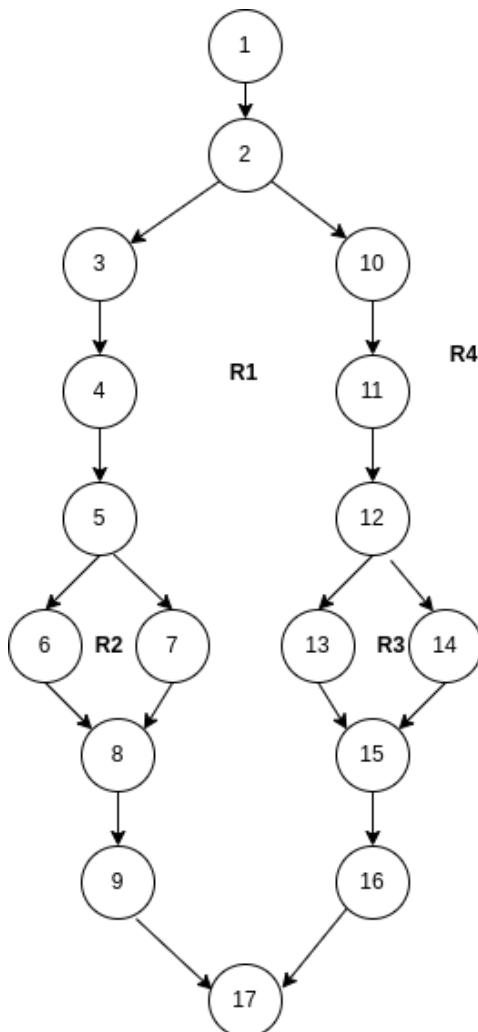


Fig7.1 control flow graph

Cyclomatic Complexity

Cyclomatic complexity is a software metric that provides a quantitative measure of the logical complexity of a program. Complexity is computed in one of three ways:

1. $V(G) = \text{Number of predicate nodes} + 1$
 $V(G) = 3+1= 3$
2. $V(G) = \text{Number of regions}$
 $V(G) = 4$
3. $V(G) = \text{Number of edges} - \text{number of vertices} + 2$
 $V(G) = 19 - 16 +2 = 4$

Basis Path Sets

Path 1 : 1, 2, 3, 4, 5, 6, 8, 9, 17

Path 2 : 1, 2, 3, 4, 5, 7, 8, 9, 16

Path 3 : 1, 2, 10, 11, 12, 13, 15, 16, 17

Path 4: 1, 2, 10, 11, 12, 14, 15, 16, 17

Test Cases

Test case 1 : The Entered details are successfully submitted.

Test case 2: The Entered details are not submitted successfully.

Test case 3: The Details are edited successfully.

Test case 4: The details are not successfully submitted.

8. REFERENCES

1. Pressman, R. S., & Maxim, B. R. (2015). Software Engineering: A Practitioner's Approach. 8th edition. McGraw-Hill.
2. Aggarwal, K. K., & Singh, Y. (2007). Software Engineering. 3rd edition. New Age International Publishers.
3. <https://www.engpaper.com/cse/index.html>
4. <https://www.figma.com/design/>
5. Maya-Period,Fertility, Ovulation android app.

9. Annexures



SCREEN 1:

EI: 0

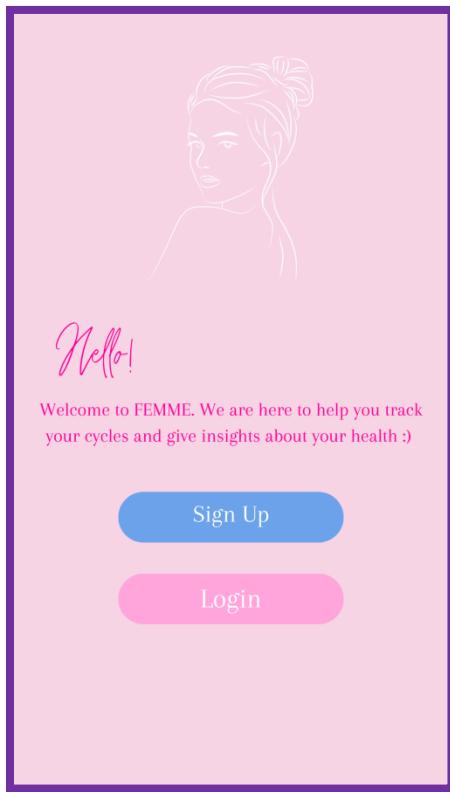
EO: 1

EQ: 0

ILF: 0

EIF: 0

9.1 DISPLAY SCREEN



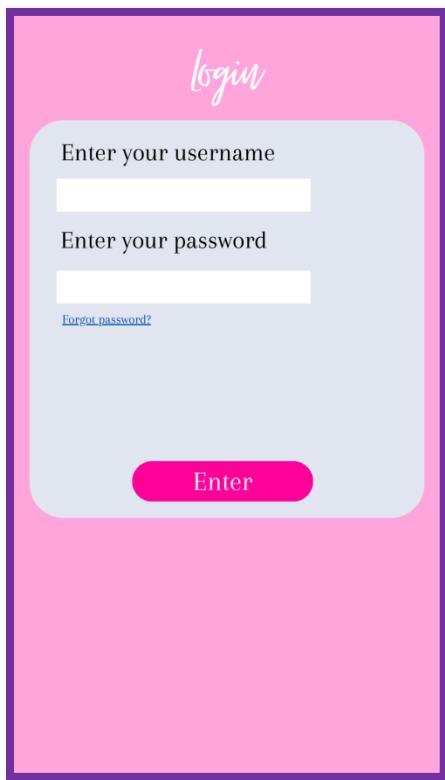
<u>SCREEN 2:</u>	
EI:	1
EO:	1
EQ:	0
ILF:	0
EIF:	0

9.2 REGISTRATION SCREEN



<u>SCREEN 3:</u>	
EI:	3
EO:	1
EQ:	0
ILF:	1
EIF:	0

9.3 REGISTRATION-1 SCREEN



SCREEN 4:

EI: 3

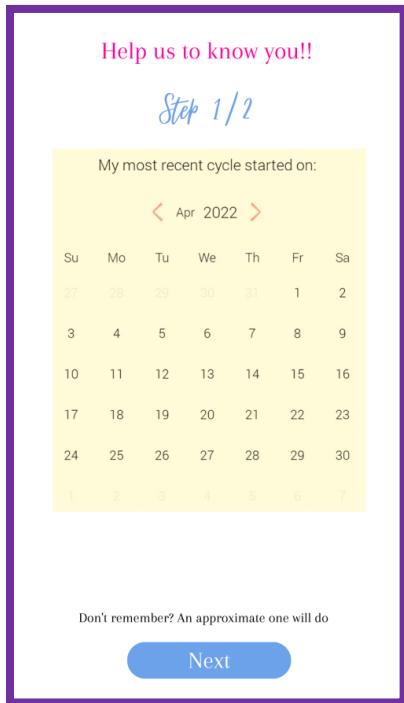
EO: 1

EQ: 0

ILF: 1

EIF: 0

9.4 LOGIN SCREEN



SCREEN 5:

EI: 1

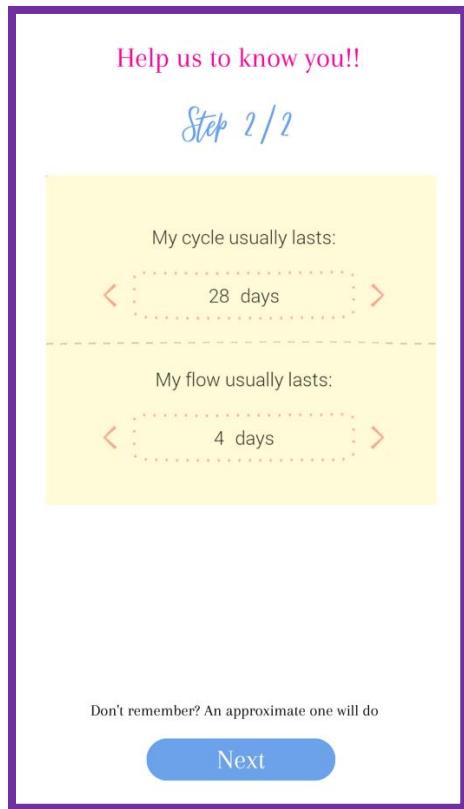
EO: 0

EQ: 0

ILF: 1

EIF: 0

9.5 UPLOAD DETAILS SCREEN



SCREEN 6:

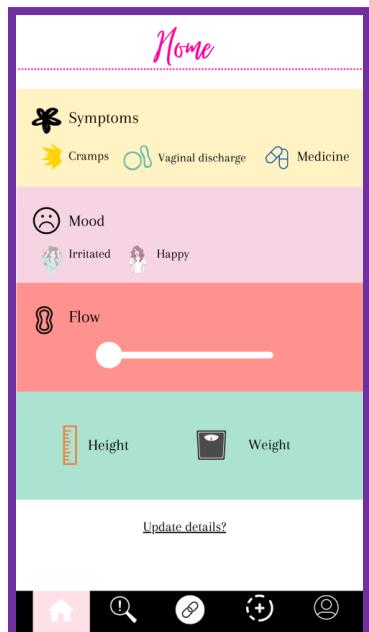
EI: 2

EO: 0

EQ: 0

ILF: 1

EIF: 0



SCREEN 7:

EI: 0

EO: 5

EQ: 0

ILF: 1

EIF: 0

9.7 HOMESCREEN OR CHECK DETAILS SCREEN

Update details

Symptoms

Flow

Height

Weight

Mood

Age

Done

SCREEN 8:

EI: 6

EO: 0

EQ: 0

ILF: 1

EIF: 0

9.8 UPDATE DETAILS SCREEN

< >

Height & weight	Symptoms	Mood
-----------------	----------	------

Spotting

Cramps

Vaginal discharge

Acne

Medicine

Fever

Tired

Dizzy

PMS

SCREEN 9:

EI: 1

EO: 0

EQ: 0

ILF: 1

EIF: 0

9.9 UPDATE DETAILS-1 SCREEN

< ✓

Height & weight	Symptoms	Mood
-----------------	----------	------

Irritated
Cranky
Calm
Blue
Happy
Anxious
Depressed
Confused
Lazy

SCREEN 10:

EI: 1

EO: 0

EQ: 0

ILF: 1

EIF: 0

9.10 UPDATE DETAILS-2 SCREEN

< ✓

Height & weight	Symptoms	Mood
-----------------	----------	------

160
60

170
70

SCREEN 11:

EI: 2

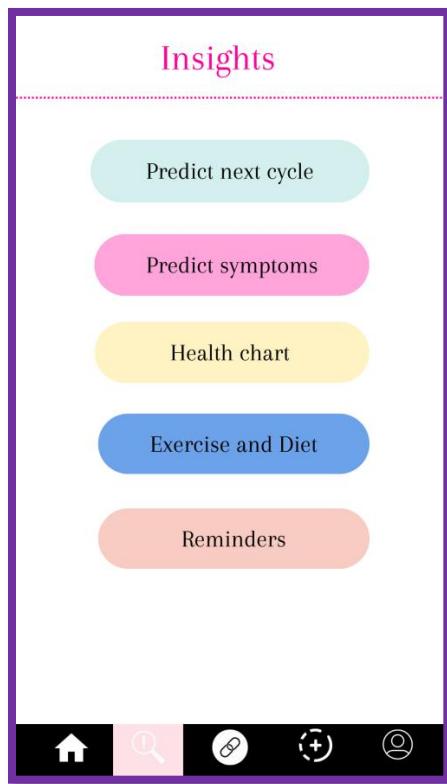
EO: 0

EQ: 0

ILF: 1

EIF: 0

9.11 UPDATE DETAILS-3 SCREEN



SCREEN 12:

EI: 0

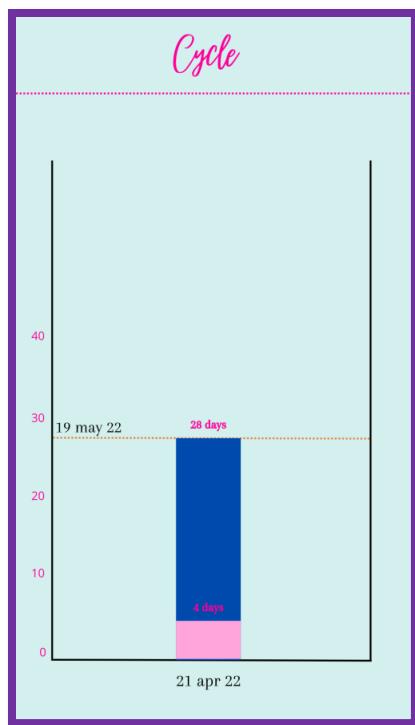
EO: 5

EQ: 0

ILF: 0

EIF: 0

9.12 CHECK DETAILS SCREEN



SCREEN 13:

EI: 0

EO: 1

EQ: 0

ILF: 0

EIF: 0

9.13 CHECK DETAILS -1 SCREEN



SCREEN 14:

EI: 0

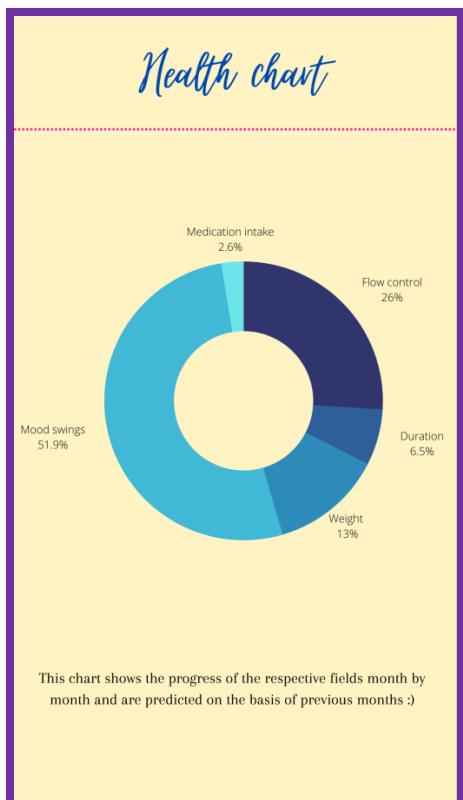
EO: 1

EQ: 0

ILF: 0

EIF: 0

9.14 CHECK DETAILS-2 SCREEN



SCREEN 15:

EI: 0

EO: 1

EQ: 0

ILF: 0

EIF: 0

9.15 CHECK DETAILS-3 SCREEN

The screenshot shows a mobile application interface. At the top, there are two tabs: "EXERCISES" (yellow) and "Diet" (light blue). The "Diet" tab is selected. Below the tabs, the word "WALKING" is listed under the "EXERCISES" section. A detailed description follows: "A simple, light walk is the best exercise you can do during your periods. This low-intensity aerobic exercise helps your lungs work properly later in your cycle. So, lace up your favourite sneakers and hit the road for a stroll or a brisk walk around." Under the "Diet" section, the word "YOGA" is listed. A detailed description follows: "Yoga can relax your cranky and irritable mood just by stretching and breathing exercises. Many yoga poses help to increase your blood circulation and provide ease to your noxious complaints. It is scientifically proven and tested that yoga helps to relax your body and relieve your period symptoms such as cramps and bloating." Under the "EXERCISES" section, the word "DANCING" is listed. A detailed description follows: "Dancing is a fun activity that can uplift your mood and burn extra calories too. So, enrol yourself in a Zumba class if you are up for it." Under the "Diet" section, the word "LIGHT LIFTING" is listed. A detailed description follows: "If you are unable to go for a walk or gym, then at least you can do light weight lifting at your home. Try light lifting and power-based moves that will result in increasing muscle flexibility and strength."

SCREEN 16:

EI: 0

EO: 1

EQ: 0

ILF: 0

EIF: 0

9.16 CHECK DETAILS-4 SCREEN

The screenshot shows a mobile application interface. At the top, there are two tabs: "EXERCISES" (yellow) and "Diet" (light blue). The "Diet" tab is selected. Below the tabs, the word "FRUITS" is listed under the "Diet" section. A detailed description follows: "Water-rich fruits, such as watermelon and cucumber, are great for staying hydrated. Sweet fruits can help you curb your sugar cravings without eating a lot of refined sugars, which can cause your glucose levels to spike and then crash." Under the "Diet" section, the word "LEAFY GREEN VEGETABLES" is listed. A detailed description follows: "It's common to experience a dip in your iron levels during your period, particularly if your menstrual flow is heavy. This can lead to fatigue, bodily pain, and dizziness. Leafy green vegetables such as kale and spinach can boost your iron levels. Spinach is also rich in magnesium." Under the "Diet" section, the word "NUTS" is listed. A detailed description follows: "Most nuts are rich in omega-3 fatty acids, and they're a great source of protein. They also contain magnesium and various vitamins. If you don't want to eat nuts on their own, try nut butters or nut-based milks or add these ingredients to smoothies." Under the "Diet" section, the word "TOFU" is listed. A detailed description follows: "A popular source of protein for vegetarians and vegans, tofu is made from soybeans. It's rich in iron, magnesium, and calcium."

SCREEN 17:

EI: 0

EO: 1

EQ: 0

ILF: 0

EIF: 0

9.17 CHECK DETAILS-5 SCREEN



SCREEN 18:

EI: 3

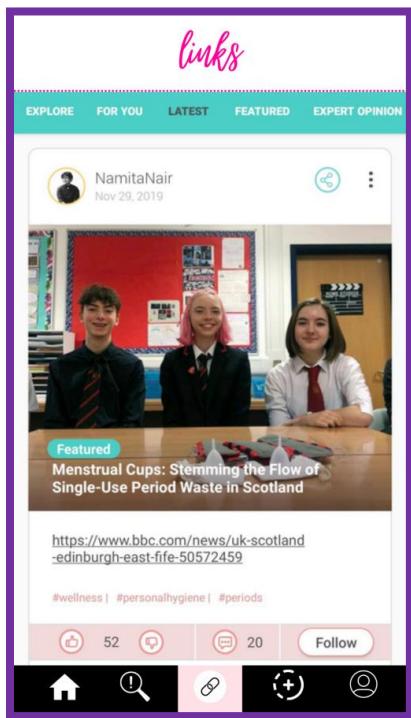
EO: 3

EQ: 0

ILF: 1

EIF: 0

9.18 REMINDERS SCREEN



SCREEN 19:

EI: 0

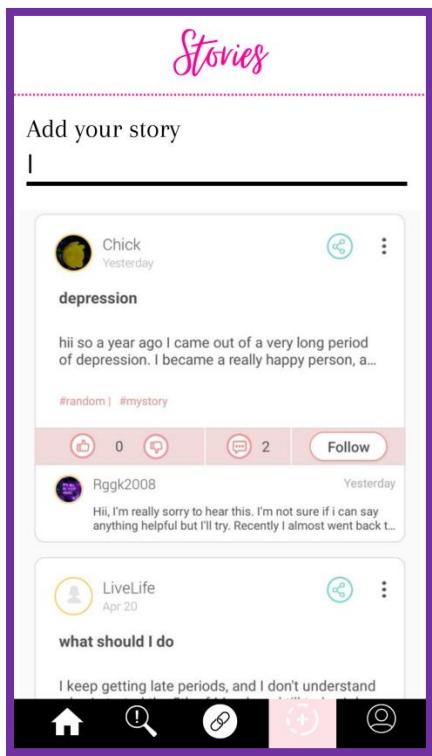
EO: 1

EQ: 1

ILF: 1

EIF: 1

9.19 ONLINE LINKS SCREEN



SCREEN 20:

EI: 1

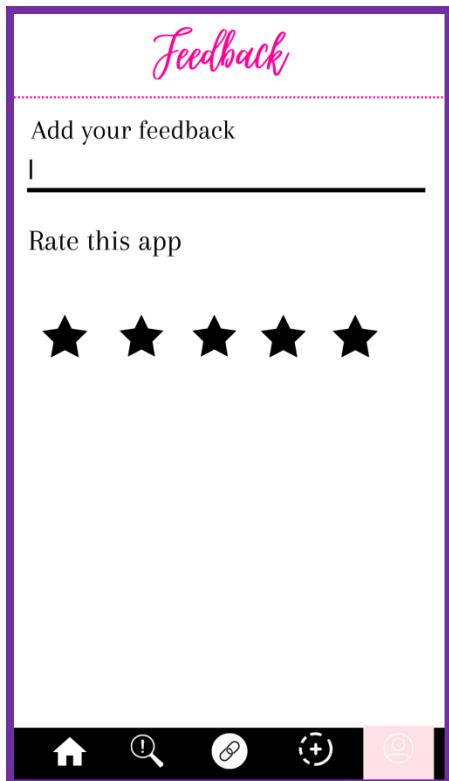
EO: 1

EQ: 1

ILF: 1

EIF: 0

9.20 SHARE&VIEW STORIES



Screen 21:

EI: 2

EO: 0

EQ: 0

ILF: 2

EIF: 0

9.21 FEEDBACK&RATING SCREEN