

| | |
|---------|--------------------------------------|
| EX NO:1 | WRITE THE COMPLETE PROBLEM STATEMENT |
| DATE | |

AIM:

To prepare a PROBLEM STATEMENT for a Daily fitness and weight training tracker.

ALGORITHM:

- The problem statement is the initial starting point for a project.
- A problem statement describes what needs to be done without describing how.
- It is generally a one-to-three-page document that all project stakeholders agree upon, describing the goals of the project at a high level.
- The problem statement is intended for a broad audience and should be written in non-technical terms.
- It helps both technical and non-technical personnel communicate effectively by providing a clear description of the problem.
- The problem statement does not describe the solution to the problem.

INPUT:

- The input to requirement engineering is the problem statement prepared by the customer.
- It may include an overview of the existing system and the broad expectations from the new system.
- The first phase of requirements engineering begins with requirements elicitation, i.e., gathering information about the requirements.

Here, requirements are identified with the help of the customer and existing system processes.

Problem:

Adopting and maintaining a consistent fitness routine poses significant challenges for many individuals. These challenges include difficulty in tracking daily activities, monitoring progress, and staying motivated. Existing methods, such as manual logs or generic fitness apps, often lack personalization and fail to provide actionable insights. Additionally, users struggle with setting achievable goals and identifying patterns in their performance, leading to reduced motivation and stagnant progress.

Background:

Fitness tracking systems have evolved with advancements in technology, enabling individuals to monitor their workouts and overall health effectively. However, there is a gap in systems that combine personalized weight training plans, daily activity tracking, and progress analysis in a user-friendly manner.

Relevance:

In a world where health consciousness is rising, a dedicated fitness and weight training tracker can empower individuals to manage their health effectively. By leveraging digital tools, such a system can streamline fitness routines, track progress with precision, and foster long-term healthy habits.

Objectives:

The primary objective of this project is to develop a comprehensive and user-friendly daily fitness and weight training tracker that enables users to monitor their progress, set goals, and stay motivated. The specific objectives include:

1. Analyze Current Methods:

Identify inefficiencies and challenges in existing fitness tracking tools to determine user needs.

2. Develop a Comprehensive Tracking Solution:

Create a system to record, store, and analyze daily fitness activities and weight training progress.

3. Integrate Goal-Setting and Progress Visualization:

Allow users to set personalized goals and track progress using intuitive charts and reports.

4. Enhance User Engagement:

Introduce notifications and reminders to maintain consistency and motivate users to achieve milestones.

5. Ensure Accessibility and Usability:

Design a user-friendly interface suitable for individuals with varying fitness levels and technical skills.

6. Prioritize Data Privacy and Security:

Implement measures to safeguard user data and ensure privacy throughout the system.

Result:

| | |
|----------------|--|
| EX NO:2 | WRITE THE SOFTWARE REQUIREMENT SPECIFICATION DOCUMENT |
| DATE | |

AIM:

To do requirement analysis and develop Software Requirement Specification Sheet(SRS) for Daily fitness and weight training tracker.

ALGORITHM:

SRS shall address are the following:

- a) **Functionality.** What is the software supposed to do?
- b) **External interfaces.** How does the software interact with people, the system's hardware, other hardware, and other software?
- c) **Performance.** What is the speed, availability, response time, recovery time of various software functions, etc.?
- d) **Attributes.** What is the portability, correctness, maintainability, security, etc. considerations?
- e) **Design constraints imposed on an implementation.** Are there any required standards in effect, implementation language, policies for database integrity, resource limits, operating environment(s) etc.?

1. Introduction

- **1.1 Purpose:**
This document defines the requirements for the Daily Fitness and Weight Training Tracker. It will help users track fitness activities, including weight training, and set fitness goals.
- **1.2 Scope:**
Describe the scope of the daily fitness and weight training tracker, including the major functionalities such as User registration and authentication, exercise logging, goal setting and tracking and data security.
- **1.3 Definitions, Acronyms, and Abbreviations:**
List all key terms, abbreviations, and acronyms used throughout the document.
- **1.4 References:**
Include any external documents, standards, or regulations that the system must comply with (e.g., Health and Fitness Data Privacy Guidelines).

2. Overall Description

- **2.1 Product Perspective:**
The tracker will be a web and mobile application integrated with fitness tracking devices. It will feature a simple, user-friendly interface.

- **2.2 Product Features:**

Provide an overview of the core features of the tracker, such as user authentication, Exercise Logging, Goal Setting & Progress Monitoring, and Data Security.

- **2.3 User Classes and Characteristics:**

Identify the different user types (e.g., users and admins) and their needs and permissions.

- **2.4 Operating Environment:**

Supports mobile (IOS, Android) and web platforms with cloud-based storage.

- **2.5 Design and Implementation Constraints:**

Comply with health data privacy regulations. Support integration with fitness trackers.

- **2.6 Assumptions and Dependencies:**

Assumes internet access for syncing data and mobile compatibility with iOS and Android.

3. System Features

- **3.1 Feature 1: Voter Registration and Authentication**

- **Description:** Secure login via email/password or social media accounts.

- **Functional Requirements:** User registration, login, password recovery

- **3.2 Feature 2: Exercise Logging**

- **Description:** Users log workouts (sets, reps, weights).

- **Functional Requirements:** Add/edit/delete workout logs.

- **3.3 Feature 3: Goal Setting and Progress Monitoring**

- **Description:** Set and track fitness goals.

- **Functional Requirements:** Display progress with visual reports

- **3.4 Feature 5: Data Security**

- **Description:** Secure data storage and transmission.

- **Functional Requirements:** Encryption, secure authentication (MFA)

-

4. External Interface Requirements

- **4.1 User Interfaces:**

The web and mobile interfaces will be designed with an intuitive layout, ensuring ease of navigation for logging workouts and tracking progress.

- **4.2 Hardware Interfaces:**

The system will sync with fitness devices like Fitbit, Apple Watch, and Garmin to import health metrics.

- **4.3 Software Interfaces:**

Integration with third-party APIs (e.g., Google Fit, Apple HealthKit) will allow syncing of health and fitness data.

- **4.4 Communication Interfaces:**

Communication between the client and server will use secure HTTPS to protect user data.

5. System Attributes

- **5.1 Performance Requirements:**

The system will handle up to 10,000 concurrent users without performance issues and load dashboards in under 3 seconds.

- **5.2 Security Requirements:**

User data will be encrypted using AES-256 encryption both during transmission (SSL/TLS) and at rest.

- **5.3 Reliability:**

The system will maintain 99.9% uptime, with only scheduled maintenance during off-peak hours.

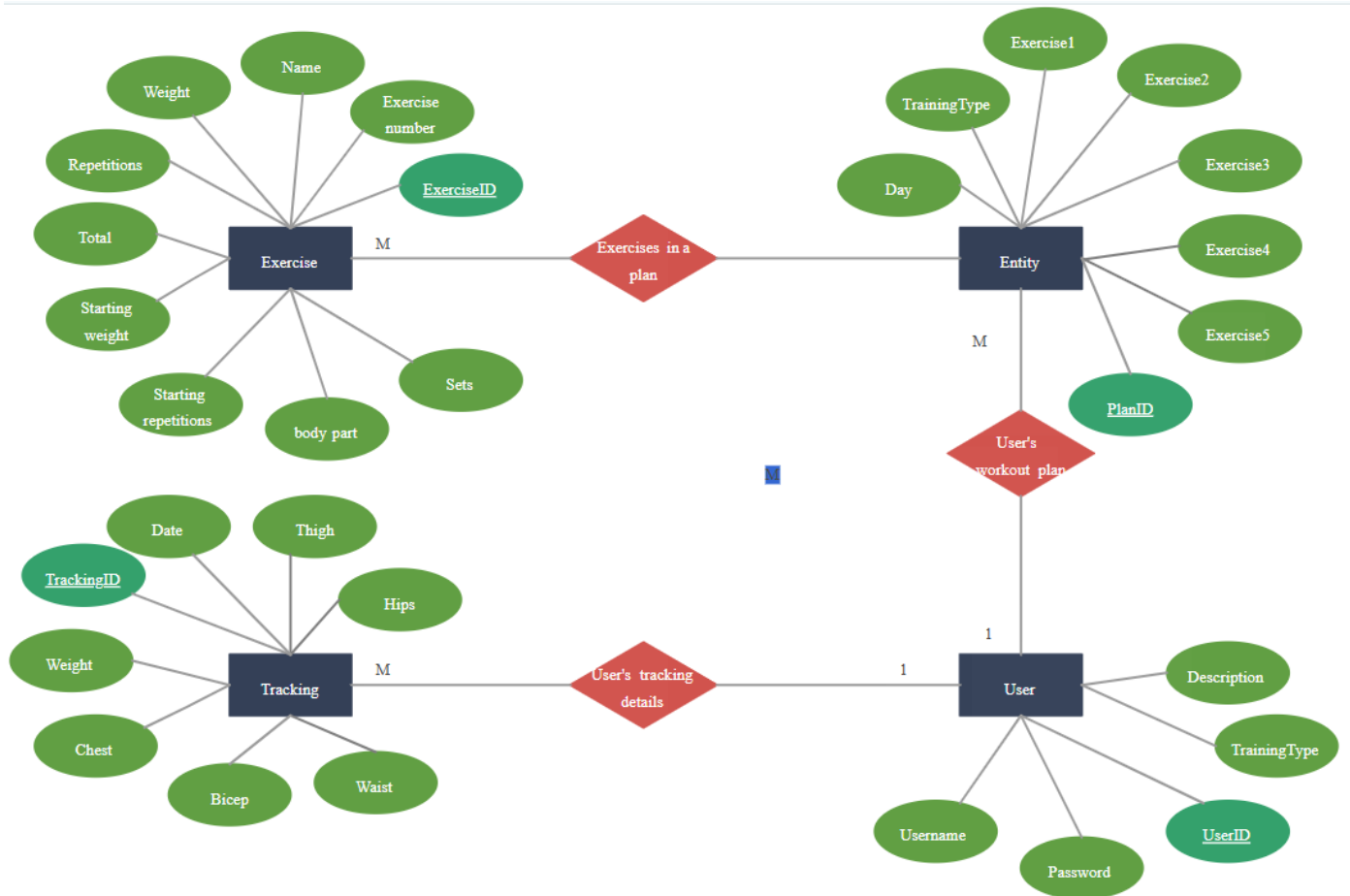
- **5.4 Availability:**

The system will be available 24/7 with minimal downtime, especially during peak usage periods.

Result:

SAMPLE OUTPUT:

ER DIAGRAM:



| | |
|----------------|---|
| EX NO:3 | DRAW THE ENTITY RELATIONSHIP DIAGRAM |
| DATE | |

AIM:

To Draw the Entity Relationship Diagram for Daily fitness and weight training tracker.

ALGORITHM:

Step 1: Mapping of Regular Entity Types

Step 2: Mapping of Weak Entity Types

Step 3: Mapping of Binary 1:1 Relation Types

Step 4: Mapping of Binary 1:N Relationship Types.

Step 5: Mapping of Binary M:N Relationship Types.

Step 6: Mapping of Multivalued attributes.

INPUT:

Entities

Entity Relationship Matrix

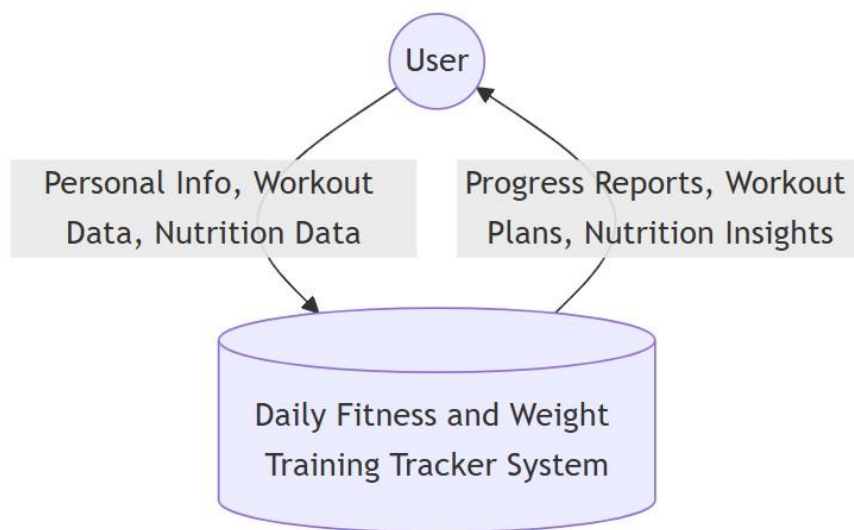
Primary Keys

Attributes

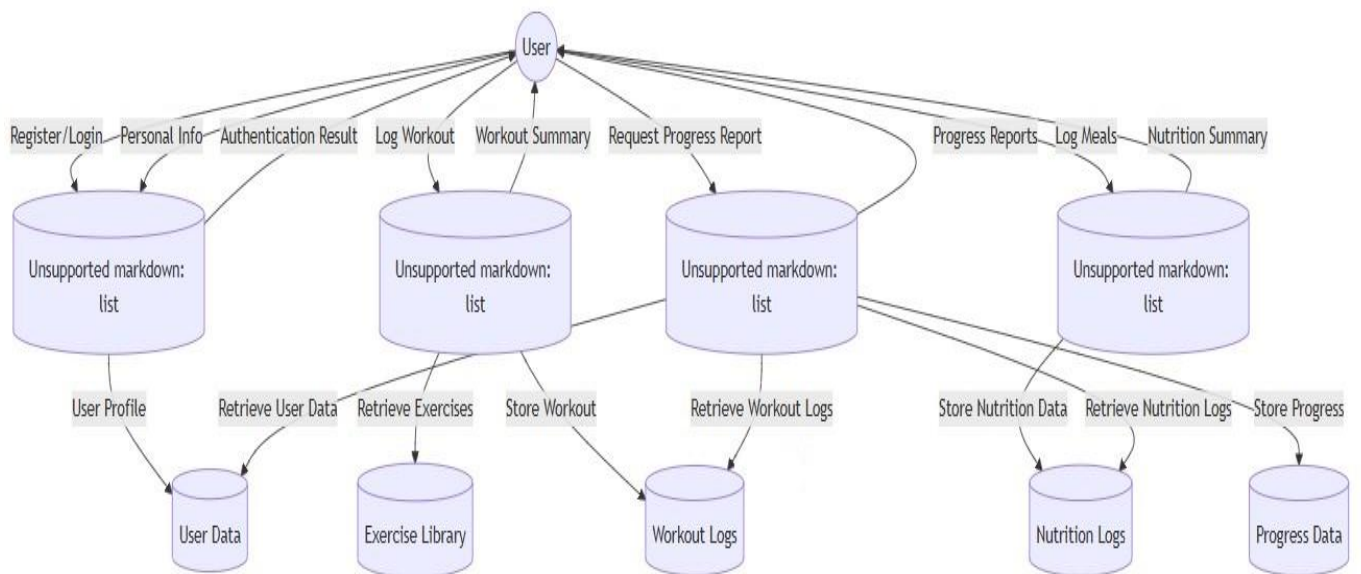
Mapping of Attributes with Entities

Result:

ZERO LEVEL:



FIRST LEVEL:



| | |
|----------------|---|
| EX NO:4 | DRAW THE DATA FLOW DIAGRAMS AT LEVEL 0 AND LEVEL 1 |
| DATE | |

AIM:

To Draw the Data Flow Diagram for Daily fitness and weight training tracker and List the Modules in the Application.

ALGORITHM:

1. Open the Visual Paradigm to draw DFD (Ex.Lucidchart)
2. Select a data flow diagram template
3. Name the data flow diagram
4. Add an external entity that starts the process
5. Add a Process to the DFD
6. Add a data store to the diagram
7. Continue to add items to the DFD
8. Add data flow to the DFD
9. Name the data flow
10. Customize the DFD with colours and fonts
11. Add a title and share your data flow diagram

INPUT:

Processes

Datastores

External Entities

Result:

SAMPLE OUTPUT:

USE CASE DIAGRAM:



| | |
|----------------|------------------------------|
| EX NO:5 | DRAW USE CASE DIAGRAM |
| DATE | |

AIM:

To Draw the Use Case Diagram for Daily fitness and weight training tracker.

ALGORITHM:

Step 1: Identify Actors

Step 2: Identify Use Cases

Step 3: Connect Actors and Use Cases

Step 4: Add System Boundary

Step 5: Define Relationships

Step 6: Review and Refine

Step 7: Validate

INPUTS:

Actors

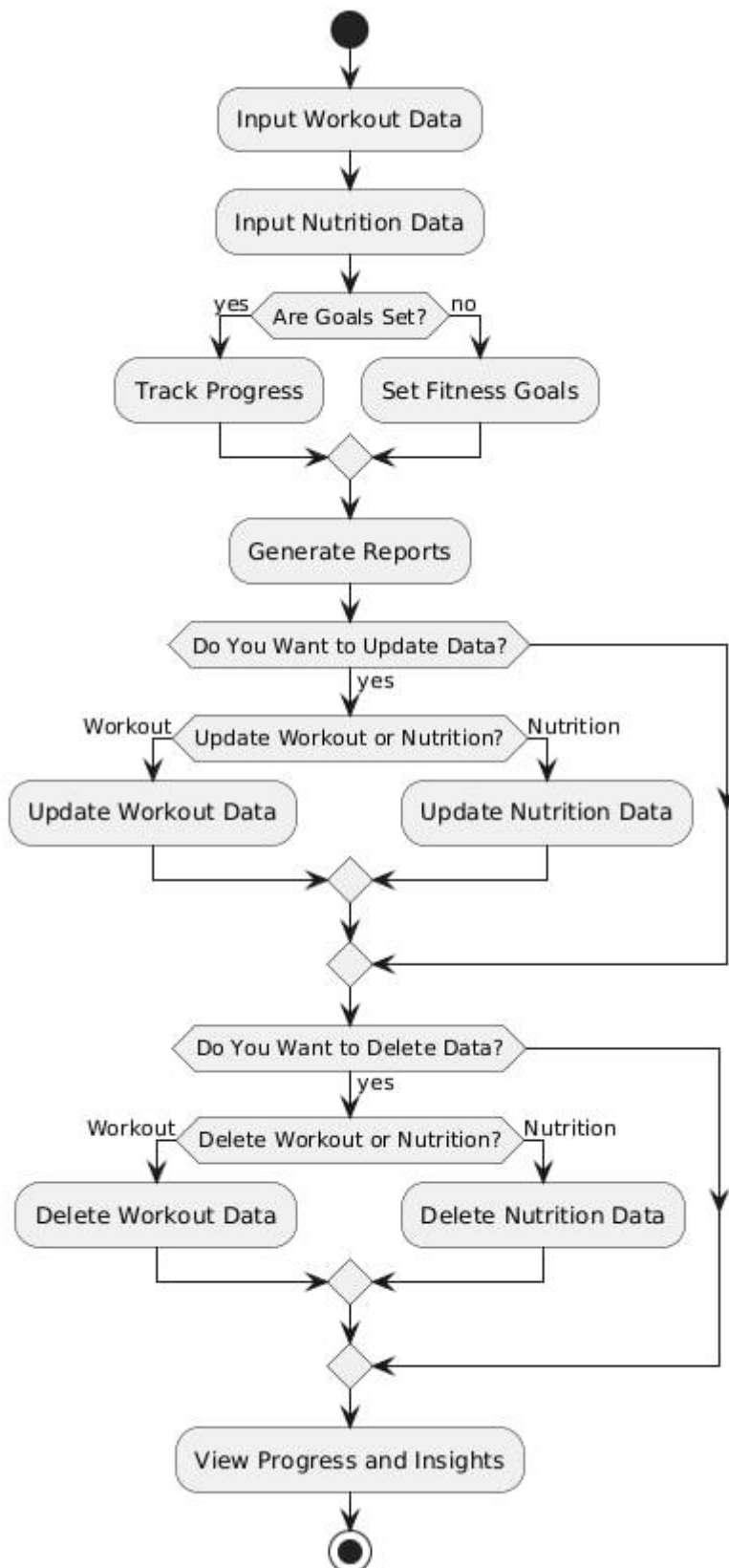
Use Cases

Relations

Result:

SAMPLE OUTPUT:

ACTIVITY DIAGRAM:



| | |
|----------------|---|
| EX NO:6 | DRAW ACTIVITY DIAGRAM OF ALL USE CASES |
| DATE | |

AIM:

To Draw the activity Diagram for Daily fitness and weight training tracker.

ALGORITHM:

Step 1: Identify the Initial State and Final States

Step 2: Identify the Intermediate Activities Needed

Step 3: Identify the Conditions or Constraints

Step 4: Draw the Diagram with Appropriate Notations

INPUTS:

Activities

Decision Points

Guards

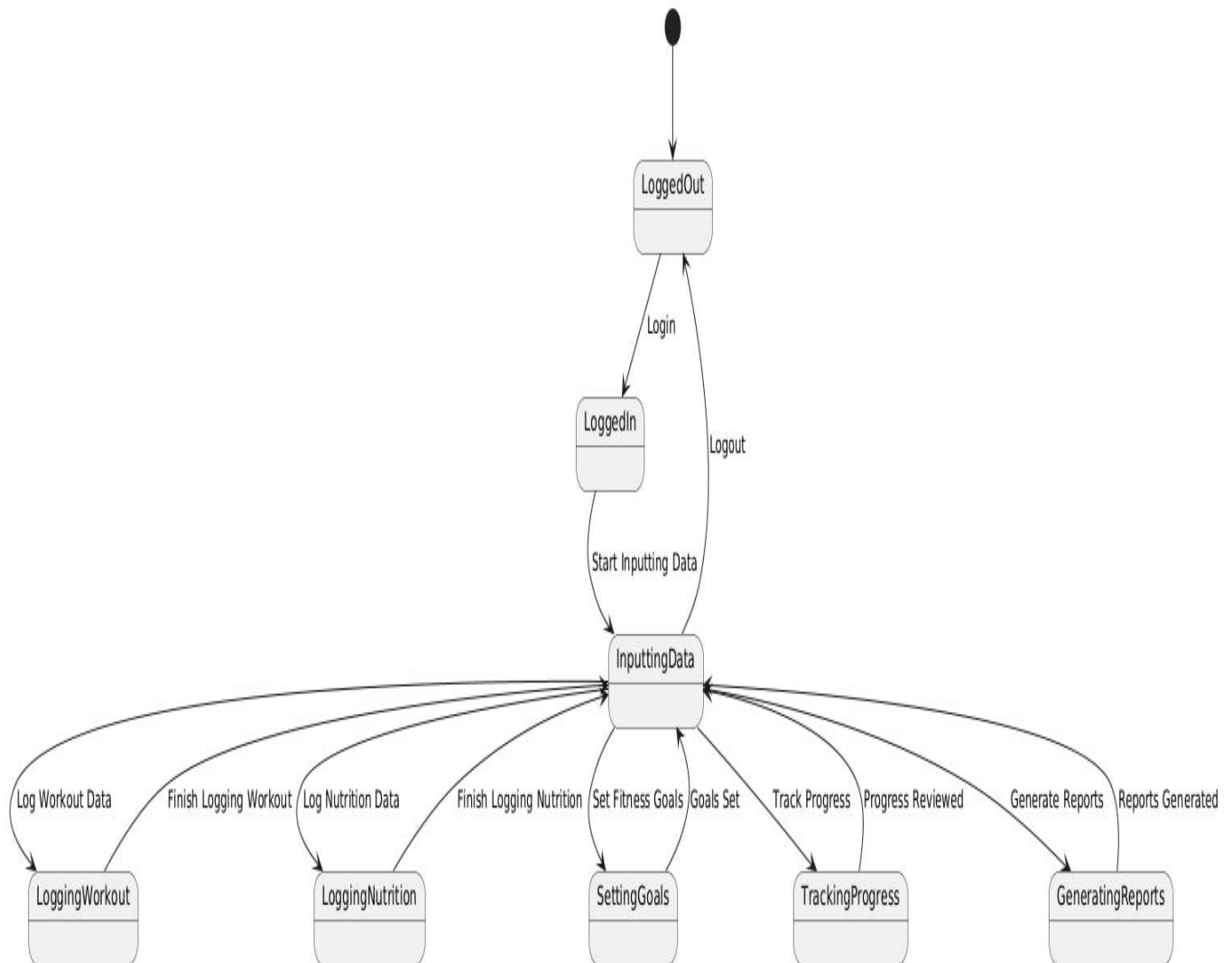
Parallel Activities

Conditions

Result:

SAMPLE OUTPUT:

STATE CHART DIAGRAM:



| | |
|----------------|---|
| EX NO:7 | DRAW STATE CHART DIAGRAM OF ALL USE CASES. |
| DATE | |

AIM:

To Draw the State Chart Diagram for Daily fitness and weight training tracker.

ALGORITHM:

STEP-1: Identify the important objects to be analysed.

STEP-2: Identify the states.

STEP-3: Identify the events.

INPUTS:

Objects

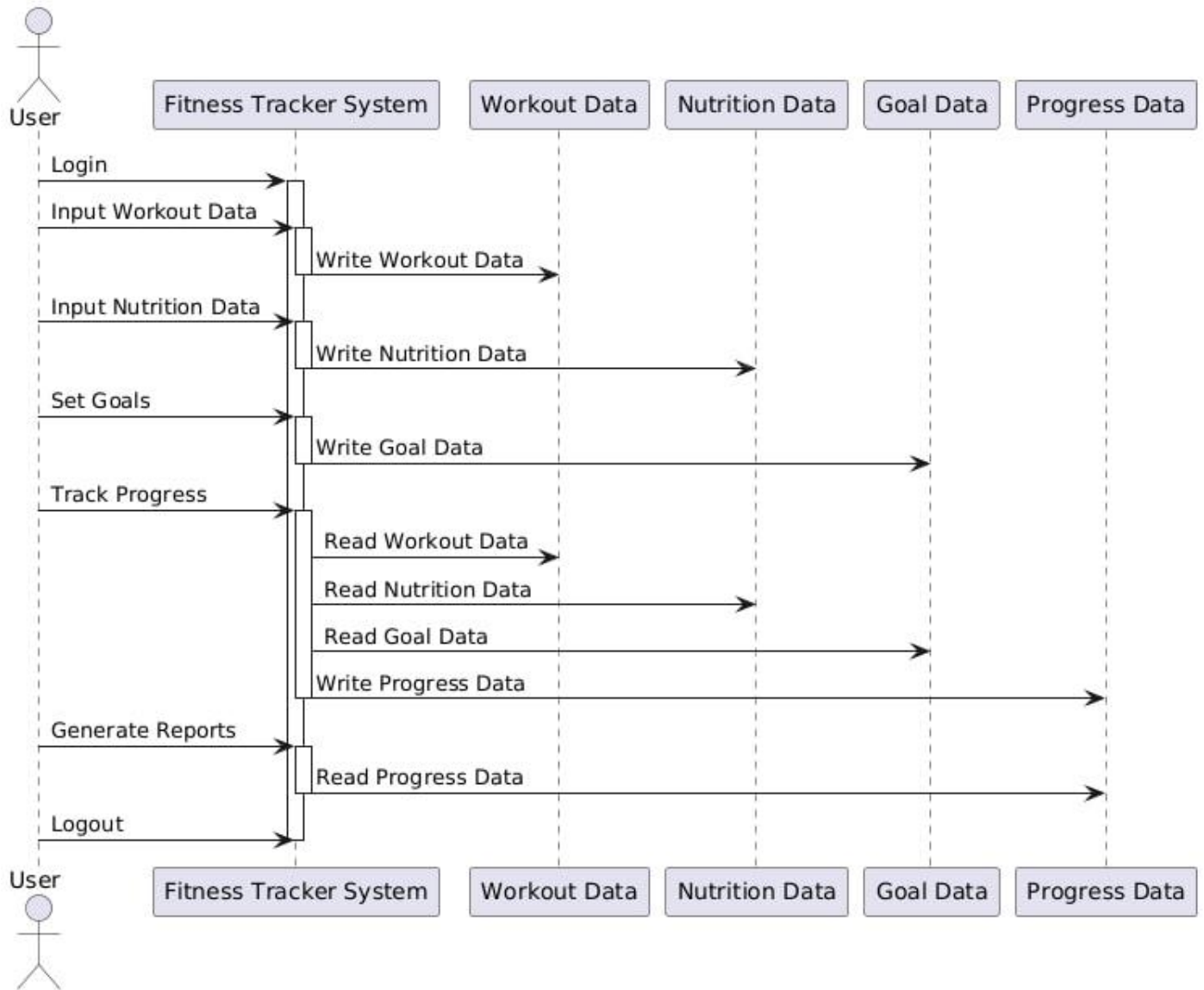
States

Events

Result:

SAMPLE OUTPUT:

SEQUENCE DIAGRAM:



| | |
|----------------|--|
| EX NO:8 | DRAW SEQUENCE DIAGRAM OF ALL USE CASES. |
| DATE | |

AIM:

To Draw the Sequence Diagram for Daily fitness and weight training tracker.

ALGORITHM:

1. Identify the Scenario
2. List the Participants
3. Define Lifelines
4. Arrange Lifelines
5. Add Activation Bars
6. Draw Messages
7. Include Return Messages
8. Indicate Timing and Order
9. Include Conditions and Loops
10. Consider Parallel Execution
11. Review and Refine
12. Add Annotations and Comments
13. Document Assumptions and Constraints
14. Use a Tool to create a neat sequence diagram

INPUTS:

Objects taking part in the interaction.

Message flows among the objects.

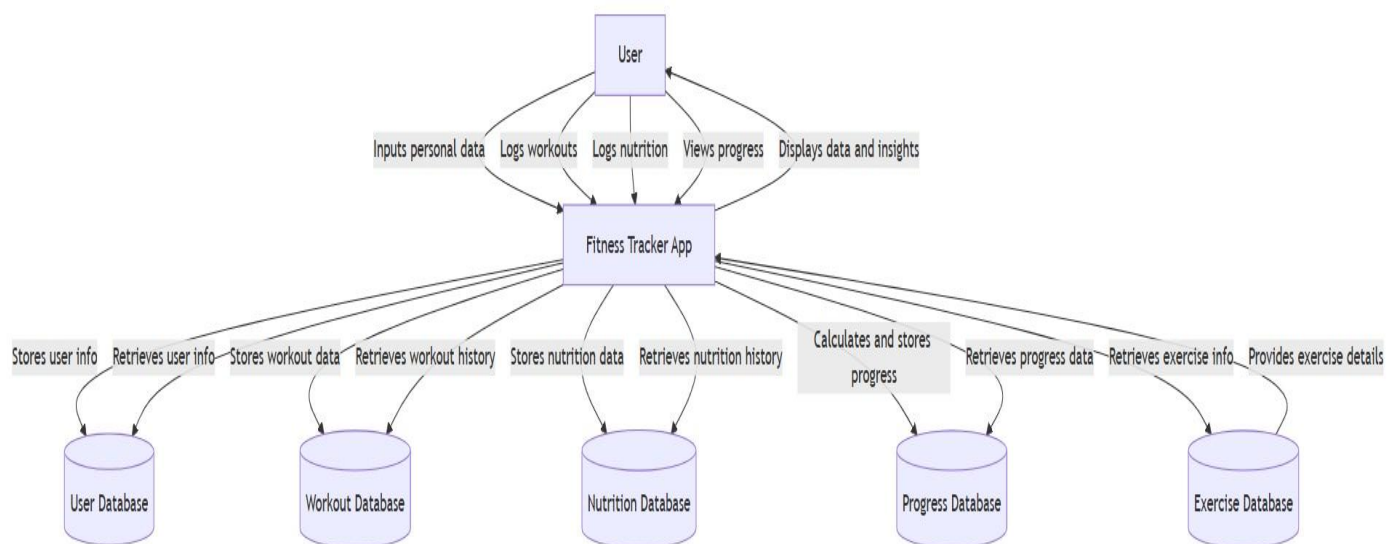
The sequence in which the messages are flowing.

Object organization.

Result:

SAMPLE OUTPUT:

COLLABORATION DIGRAM:



| | |
|----------------|--|
| EX NO:9 | DRAW COLLABORATION DIAGRAM OF ALL USE CASES |
| DATE | |

AIM:

To Draw the Collaboration Diagram for Daily fitness and weight training tracker.

ALGORITHM:

Step 1: Identify Objects/Participants

Step 2: Define Interactions

Step 3: Add Messages

Step 4: Consider Relationships

Step 5: Document the collaboration diagram along with any relevant explanations or annotations.

INPUTS:

Objects taking part in the interaction.

Message flows among the objects.

The sequence in which the messages are flowing.

Object organization.

Result:

SAMPLE OUTPUT:

CLASS DIAGRAM:



| | |
|-----------------|--|
| EX NO:10 | ASSIGN OBJECTS IN SEQUENCE DIAGRAM TO CLASSES AND MAKE CLASS DIAGRAM. |
| DATE | |

AIM:

To Draw the Class Diagram for Daily fitness and weight training tracker.

ALGORITHM:

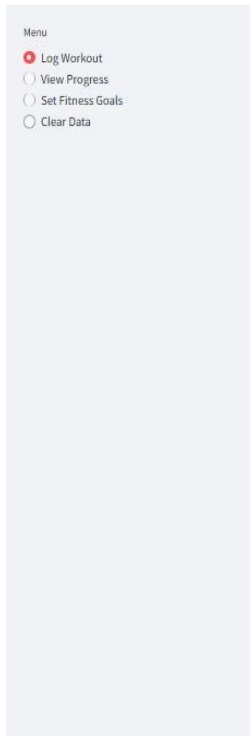
1. Identify Classes
2. List Attributes and Methods
3. Identify Relationships
4. Create Class Boxes
5. Add Attributes and Methods
6. Draw Relationships
7. Label Relationships
8. Review and Refine
9. Use Tools for Digital Drawing

INPUTS:

1. Class Name
2. Attributes
3. Methods
4. Visibility Notation

RESULT:

OUTPUT:



Daily Fitness & Weight Training Tracker

Track your workouts, log progress, and achieve your fitness goals!

Log Your Workout

Date

2024/11/21

Exercise Name

bicep curl

Weight (kg)

5.00

- +

Reps

3

- +

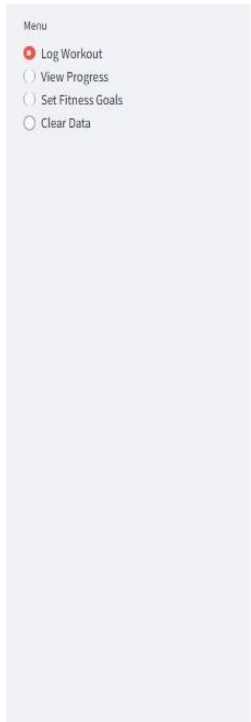
Duration (minutes)

10

- +

Add Workout

Workout logged successfully!



Daily Fitness & Weight Training Tracker

Track your workouts, log progress, and achieve your fitness goals!

Log Your Workout

Date

2024/11/19

< November 2024 >

| | | | | | | |
|----|----|----|----|----|----|----|
| Su | Mo | Tu | We | Th | Fr | Sa |
| | | | | | 1 | 2 |
| 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| 17 | 18 | 19 | 20 | 21 | 22 | 23 |
| 24 | 25 | 26 | 27 | 28 | 29 | 30 |

Add Workout

| | |
|-----------------|---|
| EX NO:11 | MINI PROJECT- DAILY FITNESS AND WEIGHT TRAINING TRACKER. |
| DATE | |

AIM:

The primary aim of the "Daily Fitness and Weight Training Tracker" project is to develop a user-friendly application that enables individuals to effectively monitor and manage their fitness and nutrition goals. The application will provide users with the ability to log their daily workouts, track their nutritional intake, set personalized fitness goals, and monitor their progress over time.

ALGORITHM:

1. User registers with valid credentials (username, password, email).
2. User logs in using their credentials (username, password).
3. System displays the main menu of options (log workout, log nutrition, set goals, track progress, generate reports, update profile, logout).
4. User selects an option from the main menu.
5. User logs workout details (exercise type, duration, calories burned) and stores the entry.
6. User logs nutrition details (food item, calories, protein, carbs, fats) and stores the entry.
7. User sets fitness goals (goal type, target value, deadline) and stores the goal.
8. User tracks progress (date, weight, body fat percentage, notes) and stores the entry.
9. User generates reports (start date, end date) and retrieves compiled data.
10. User updates their profile (new username, new password) and saves the changes.

PROGRAM:

```
import streamlit as st
```

```
import pandas as pd
```

```
# Initialize session state for tracking workouts
```

```
if 'workouts' not in st.session_state:
```

```
    st.session_state.workouts = []
```

```
if 'fitness_goal' not in st.session_state:
```

```
    st.session_state.fitness_goal = "Stay consistent with workouts!"
```

Menu

- ☐ Log Workout
- ☒ View Progress
- ☐ Set Fitness Goals
- ☐ Clear Data

Daily Fitness & Weight Training Tracker

Track your workouts, log progress, and achieve your fitness goals!

Workout Progress

| | Date | Exercise | Weight (kg) | Reps | Duration (minutes) |
|---|------------|-------------|-------------|------|--------------------|
| 0 | 2024-11-21 | bicep curl | 5.0000 | 3 | 10 |
| 1 | 2024-11-19 | tricep curl | 5.0000 | 2 | 15 |

```
# Application title
```

```
st.title("Daily Fitness & Weight Training Tracker")
```

```
st.write("Track your workouts, log progress, and achieve your fitness goals!")
```

```
# Sidebar menu
```

```
menu = st.sidebar.radio("Menu", ["Log Workout", "View Progress", "Set Fitness Goals", "Clear Data"])
```

```
# Log Workout Section
```

```
if menu == "Log Workout":
```

```
    st.header("Log Your Workout")
```

```
    date = st.date_input("Date")
```

```
    exercise = st.text_input("Exercise Name")
```

```
    weight = st.number_input("Weight (kg)", min_value=0.0, step=0.5)
```

```
    reps = st.number_input("Reps", min_value=1, step=1)
```

```
    duration = st.number_input("Duration (minutes)", min_value=0, step=1)
```

```
    if st.button("Add Workout"):
```

```
        if exercise:
```

```
            st.session_state.workouts.append({
```

```
                "Date": date,
```

```
                "Exercise": exercise,
```

```
                "Weight (kg)": weight,
```

```
                "Reps": reps,
```

```
                "Duration (minutes)": duration,
```

```
            })
```

```
            st.success("Workout logged successfully!")
```

```
        else:
```

Menu

☐ Log Workout

☐ View Progress

☒ Set Fitness Goals

☐ Clear Data

Daily Fitness & Weight Training Tracker

Track your workouts, log progress, and achieve your fitness goals!

Set Your Fitness Goals

Your Fitness Goal

Improve my arm and chest!!!!

Save Goal

Fitness goal saved!

Current Fitness Goal:

Improve my arm and chest!!!!


```

        st.error("Please enter an exercise name.")

# View Progress Section

elif menu == "View Progress":

    st.header("Workout Progress")

    if st.session_state.workouts:

        df = pd.DataFrame(st.session_state.workouts)

        st.table(df)

    else:

        st.write("No workout data available. Log some workouts first!")


# Set Fitness Goals Section

elif menu == "Set Fitness Goals":

    st.header("Set Your Fitness Goals")

    goal = st.text_area("Your Fitness Goal", value=st.session_state.fitness_goal)

    if st.button("Save Goal"):

        st.session_state.fitness_goal = goal

        st.success("Fitness goal saved!")


    st.write("### Current Fitness Goal:")

    st.write(st.session_state.fitness_goal)


# Clear Data Section

elif menu == "Clear Data":

    st.header("Clear All Data")

    if st.button("Clear Workouts"):

        st.session_state.workouts = []

        st.success("Workout data cleared!")

```

Menu

- ☐ Log Workout
- ☐ View Progress
- ☐ Set Fitness Goals
- ☒ Clear Data

Daily Fitness & Weight Training Tracker

Track your workouts, log progress, and achieve your fitness goals!

Clear All Data

Clear Workouts

Clear Fitness Goal

```
if st.button("Clear Fitness Goal"):

    st.session_state.fitness_goal = "Stay consistent with workouts!"

    st.success("Fitness goal reset!")


# Footer

st.markdown("---")

st.markdown("Powered by [Streamlit](https://streamlit.io)")
```

Conclusion:

The Daily Fitness and Weight Training Tracker project is a valuable tool that helps individuals enhance their fitness journeys by logging workouts, tracking nutrition, and monitoring progress. Its user-friendly interface promotes accountability and motivation, allowing users to analyze their performance and make informed adjustments. This project highlights the role of technology in personal health management, encouraging healthier lifestyles and supporting users in achieving their fitness goals. Future enhancements could include social sharing and integration with wearable devices to further improve the experience.