==========

ACTIVITY SHEET 1 — REQUIREMENTS GATHERING (SRS)

==========

Project Title

Fitness Activity Tracker with Food Ordering System (Active Goal Tracker)

Team Members

E Ganesh Sai – 25RBAID0019

BM Nikhil Sai Deepak – 25RBCSE0007

U Shiva Chaithanya Goud – 25RBAIM0082

A Pradeep – 25RBCSE0003

K Sai Charan – 25RMAIM0085

M Shiva Teja --25RBAID0007

---

1. Stakeholder Interview

Stakeholders

- End Users (Fitness Enthusiasts)
- Nutritionists
- Gym Trainers
- Food Vendors
- System Administrator
- App Developers

Interview Questions (8–10)

1. What fitness activities should be tracked?
2. Should calorie intake and burn be monitored?
3. Is food ordering required inside the app?
4. Should diet plans be personalized?
5. What type of workout data should be stored?
6. Should reminders and notifications be included?

7. Is integration with wearable devices needed?
8. How secure should user health data be?
9. Should the system recommend healthy meals?
10. What reports or progress charts are required?

---

2. Extract Requirements

Requirement ID	Type	Description
R1	Functional	Track daily fitness activities
R2	Functional	Calculate calories burned
R3	Functional	Allow users to order healthy food
R4	Functional	Recommend diet based on fitness goals
R5	Functional	Store user fitness history
R6	Non-Functional	System should be secure
R7	Non-Functional	System should be user-friendly
R8	Non-Functional	System should give quick response

---

3. User Stories

1. As a user, I want to track my workouts so that I can improve fitness.
2. As a user, I want calorie tracking so that I can manage my diet.
3. As a user, I want to order healthy food so that I save time.
4. As a user, I want diet recommendations so that I reach my goals.
5. As an admin, I want secure data storage so that user data is protected.

Deliverable: Mini SRS Document

==========

ACTIVITY SHEET 2 — SYSTEM DESIGN

==========

1. System Architecture (Text Description)

Components

- Mobile/Web User Interface
- Fitness Tracking Module
- Calorie Calculation Module
- Food Ordering Module
- Payment Gateway
- Database

Data Flow

User → UI → Fitness/Food Modules → Database → Reports

---

2. UI Wireframes (Screens)

Home Screen
Login / Signup
Fitness Tracking Screen
Food Ordering Screen
Progress Dashboard
Settings / Profile

---

3. Database Design (ERD)

Entity	Attributes
Users	User ID, Name, Age, Height, Weight, Email
Activities	Activity ID, User ID, Activity Type, Duration, Calories Burned
Food Items	Item ID, Name, Calories, Price, Category
Orders	Order ID, User ID, Items, Total Price, Status

Deliverable: Architecture Diagram + Wireframes + ERD

==========

ACTIVITY SHEET 3 — DEVELOPMENT PHASE

==========

1. Development Backlog

Task ID	Feature	Description	Assignee	Status
T1	User Registration	Signup/Login	Ganesh	Done
T2	Fitness Tracking	Track steps & workouts	Nikhil& shiva	Done
T3	Food Ordering	Select & order food	pradeep	In Progress
T4	Calorie Calculation	Calculate intake & burn	Sai charan	Done

---

2. Code Walkthrough Notes

Features Implemented Today

- User registration
- Activity tracking
- Calorie calculation

Challenges Faced

- Mapping food calories with fitness goals

Next Steps

- Integrate payment system
- Improve UI dashboard

Deliverable: Updated backlog + coded demo

==========

ACTIVITY SHEET 4 — TESTING PHASE

==========

1. Test Cases

Test Case ID	Description	Steps	Expected Result	Actual Result	Status
TC1	User Login	Enter credentials	Login success	Same	P
TC2	Activity Tracking	Add workout	Activity saved	Same	P
TC3	Food Order	Place order	Order confirmed	Same	P

---

2. Bug Report

Bug ID	Description	Severity	Steps	Status
B1	Calorie mismatch	Medium	Add food	Fixed

---

3. Test Summary

- Total test cases: **3**
- Passed: **3**
- Failed: **0**
- Major issues found: **None**

Deliverable: Test cases + bug report

==========

ACTIVITY SHEET 5 — DEPLOYMENT & RELEASE

==========

1. Deployment Checklist

Code merged
Database configured
Environment variables set
Build successful
Final testing done
Version tagged (v1.0)

---

2. Release Notes

Release Version: v1.0

Features Included

- Fitness activity tracking
- Calorie monitoring
- Healthy food ordering
- Progress dashboard

Known Issues

- Limited food vendor support

Next Update Goals

- Wearable device integration
- AI-based diet recommendations

Deliverable: Deployment & Release Document

==========

ACTIVITY SHEET 6 — MAINTENANCE & REFLECTION

==========

1. Patch Log

Patch ID	Issue	Root Cause	Fix Implemented	Status
P1	Food calorie error	Data mismatch	Updated database	Done

---

2. Team Retrospective

What worked well

- Clear module separation
- Easy user interface

What needs improvement

- Faster food search

Changes for next time

- Early testing
- More food options

Deliverable: Final reflection

FOOD ORDERING APP SECTION

(Converted to Fitness Tracker with Food System)

==========

ACTIVITY SHEET 1 — REQUIREMENTS GATHERING

==========

Project Title

Food Ordering System Integrated with Fitness Tracker

Team Members

E Ganesh Sai – 25RBAID0019

BM Nikhil Sai Deepak – 25RBCSE0007

U Shiva Chaithanya Goud – 25RBAIM0082

A Pradeep – 25RBCSE0003

K Sai Charan – 25RMAIM0085

M Shiva Teja –25RBAID0007

1. Stakeholder Interview

Stakeholders

- Fitness App Users
- Nutritionists
- Food Vendors

- Delivery Personnel
- System Administrator
- Payment Gateway Provider

Interview Questions (8–10)

1. What type of food should be available (healthy/fitness-based)?
2. Should calorie and nutrition details be shown?
3. Should food recommendations match fitness goals?
4. Is online payment mandatory?
5. Should users track daily calorie intake from food?
6. Is order history required?
7. Should delivery tracking be included?
8. How secure should payment and user data be?
9. Should users customize meals?
10. What reports should be generated?

---

2. Extract Requirements

Requirement ID	Type	Description
R1	Functional	Display healthy food menu
R2	Functional	Show calories and nutrition info
R3	Functional	Add food items to cart
R4	Functional	Place food orders
R5	Functional	Online payment support
R6	Functional	Store order history
R7	Non-Functional	Secure transactions
R8	Non-Functional	Fast system response

---

3. User Stories

1. As a user, I want to view healthy meals so that I maintain fitness.
2. As a user, I want calorie information so that I control intake.
3. As a user, I want to order food online so that I save time.
4. As a user, I want order history so that I track my diet.
5. As an admin, I want to manage food items so that menu stays updated.

Deliverable: Mini SRS Document

==========

ACTIVITY SHEET 2 — SYSTEM DESIGN

==========

1. System Architecture (Text Description)

Components

- User Interface (Mobile/Web)
- Food Menu Module
- Cart Management Module
- Order Processing Module
- Payment Gateway
- Database

Data Flow

User → Food Menu → Cart → Payment → Order Confirmation → Database

---

2. UI Wireframes (Screens)

Home
Login / Signup
Food Menu Screen
Cart Screen
Checkout Screen
Order History
Settings / Profile

---

3. Database Design — ERD

Entity	Attributes
Users	User ID, Name, Email, Phone, Address
Food Items	Item ID, Name, Calories, Price, Category
Orders	Order ID, User ID, Total Calories, Total Price, Status
Order Items	Order Item ID, Order ID, Item ID, Quantity

Deliverable: Architecture + Wireframes + ERD

==========

ACTIVITY SHEET 3 — DEVELOPMENT PHASE

==========

1. Development Backlog

Task ID	Feature	Description	Assignee	Status
T1	User Login	Authentication	Sukanya	Done
T2	Food Menu	Display food list	Sukanya	Done
T3	Cart	Add/remove items	Sukanya	Done
T4	Order Placement	Confirm order	Sukanya	Done
T5	Payment	Online payment	Sukanya	In Progress

---

2. Code Walkthrough Notes

Features Implemented

- Food menu display
- Cart management
- Order placement

Challenges Faced

- Accurate calorie calculation

Next Steps

- Complete payment integration
- Improve UI performance

Deliverable: Updated backlog

==========

ACTIVITY SHEET 4 — TESTING PHASE

==========

1. Test Cases

Test Case ID	Description	Steps	Expected Result	Status
TC1	View Menu	Open menu	Menu displayed	P
TC2	Add to Cart	Select item	Item added	P
TC3	Place Order	Checkout	Order confirmed	P
TC4	Payment	Pay online	Payment success	P

---

2. Bug Report

Bug ID	Description	Severity	Status
B1	Calorie mismatch	Medium	Fixed

---

3. Test Summary

- Total test cases: **4**
- Passed: **4**
- Failed: **0**
- Major issues found: **None**

Deliverable: Test case sheet + bug report

==========

ACTIVITY SHEET 5 — DEPLOYMENT & RELEASE

==========

1. Deployment Checklist

Code merged
Database configured
Environment variables set
Build successful
Final testing done
Version tagged (v1.0)

---

2. Release Notes

Release Version: v1.0

Features Included

- Healthy food ordering
- Calorie tracking
- Cart & checkout
- Order history

Known Issues

- Limited payment options

Next Update Goals

- Subscription meal plans
- AI-based food recommendations

Deliverable: Deployment & Release Document

==========

ACTIVITY SHEET 6 — MAINTENANCE & REFLECTION

==========

1. Patch Log

Patch ID	Issue	Root Cause	Fix Implemented	Status
P1	Calorie error	Data mismatch	Updated database	Done

---

2. Team Retrospective

What worked well

- Smooth ordering flow
- Clear requirements

What needs improvement

- Faster menu loading

Changes for next time

- Early testing
- Performance optimization

Deliverable: Final reflection

🔄 **FLOWCHARTS**

Flowchart 1: Food Ordering Process

Start

↓

User Login

↓

View Food Menu

↓

Select Food Item

↓

Add to Cart

↓

Checkout

↓

Make Payment

↓

Order Confirmed

↓

Store Order in Database

↓

End

Flowchart 2: Calorie Tracking with Food Order

Start

↓

Select Food Item

↓

Get Calorie Value

↓

Add to Daily Intake

↓

Update Fitness Dashboard

↓

End

Flowchart 3: Payment Process

Start

↓

Select Payment Method

↓

Verify Payment

↓

Payment Successful?

↓ ↓

Yes No

↓ ↓

Confirm Retry

Order

↓

End

CODE :

```
# ActiveGoal Tracker – Fitness & Food Ordering App


users = {}

activities = []


food_menu = {

   1: ("Salad", 150, 120),

   2: ("Fruit Bowl", 180, 100),

   3: ("Protein Shake", 250, 180)

}
```

```
def register():
    u = input("Username: ")
    p = input("Password: ")
    users[u] = p
    print("Registration successful\n")

def login():
    u = input("Username: ")
    p = input("Password: ")
    if users.get(u) == p:
        print("Login successful\n")
        return u
    else:
        print("Invalid login\n")
        return None

def track_activity(user):
    act = input("Activity name: ")
    dur = int(input("Duration (min): "))
    cal = dur * 5
    activities.append((user, act, dur, cal))
    print("Activity saved | Calories burned:", cal)

def order_food():
    print("\nFood Menu")
    for k, v in food_menu.items():
        print(k, v[0], "Calories:", v[1], "Price:", v[2])
```

```
    ch = int(input("Choose item: "))
    qty = int(input("Quantity: "))
    food, cal, price = food_menu[ch]

    print("\nOrder Summary")
    print("Food:", food)
    print("Total Calories:", cal * qty)
    print("Amount to Pay: ₹", price * qty)
    print("Payment Successful!\n")

def dashboard(user):
    while True:
        print("\n1.Track Activity  2.Order Food  3.Logout")
        c = int(input("Enter choice: "))

        if c == 1:
            track_activity(user)
        elif c == 2:
            order_food()
        elif c == 3:
            break

while True:
    print("\nActiveGoal Tracker")
    print("1.Register  2.Login  3.Exit")
    ch = int(input("Choice: "))
```

```
if ch == 1:
    register()
elif ch == 2:
    user = login()
    if user:
        dashboard(user)
elif ch == 3:
    print("Thank you!")
    break
```