SW Engineering CSC648-848-05
Summer 2024

FitNutri Hub
Team 03

Student	Full Name	SFSU Email	Role
# 1	Michelle Nguyen	mnguyen62@sfsu.edu	Team-Lead
#2	Mitchell Caine	mcaine@sfsu.edu	Frontend-Lead
#3	Shreejana Bartaula	sbartaula@mail.sfsu.edu	Docs-editors
#4	Ali Almusawi	aalmusawi@mail.sfsu.edu	GitHub-master
#5	Eduardo Enrique Muñoz Alvarez	emunozalvarez@sfsu.edu	Database-admin
#6	Nilofar Ali	nmohammadali@mail.sfsu.edu	Backend-Lead
#7	Uzair Hamed Mohammed	umohammed@sfsu.edu	Doc-editor and Backend
#8	John Collins	jcollins9@sfsu.edu	Backend
#9	Ali Hadwan	hadwanali41@gmail.com	Frontend

MILESTONE 2
Date: 06/11/2024

History Table

Milestone	**Version**	**Date Submitted**
Milestone 2	V1	7/9/2024

Table of Contents

1. Data Definitions

1. User Data

1.1. User ID: A unique identifier for each user.

1.2 Username: The chosen display name of the user.

1.3. Email: The user's email address for communication and account recovery.

1.4. Password: A hashed string used for user authentication.

1.5. Date of Birth: The user's date of birth.

1.6. Gender: The user's gender.

1.7. Height: The user's height.

1.8. Weight: The user's weight.

2. Activity Data

2.1. Activity ID: A unique identifier for each recorded activity.

2.2. Activity date: The Time and date when the activity took place

2.3. Activity type: The type of physical activity.

2.4. Activity duration: How long the user was doing the activity for.

2.5. Activity distance: The distance covered during the activity.

2.6. Calories burned: Estimated number of calories burned during the activity.

2.7. Heart rate: The user's average heart rate during the activity.

3. Nutrition Data

3.1. Meal ID: A unique identifier for each meal entry.

3.2. Meal type: The type of meal (breakfast, lunch, dinner, snack).

3.3. Food item: The name of the food item consumed.

3.4. Calories consumed: The number of calories in the food item.

3.5. Macronutrients: The amount of carbohydrates, proteins, and fats in the food item.

3.6. Micronutrients: List of Vitamins (A, B, C, D, E, and K) and Minerals (Ca, Mg, Zn, Fe) plus fiber content

4. Recipe Data

4.1. Recipe ID: A unique identifier for each recipe.

4.2. Recipe name: The name of the recipe.

4.3. Description: A description of the recipe.

4.4. Category: The recipe category (dessert, appetizer, main course).

4.5. Preparation time: The time required to prepare the ingredients.

4.6. Cooking time: The time required to cook the dish.

4.7. Total time: The total time required to prepare and cook the recipe.

4.8. Servings: The number of servings the recipe yields.

4.9. Ingredients: the name of the ingredients.

4.10. Nutrition information: nutritional information (calories, carbohydrates, proteins, fats, fiber, sodium).

5. Health Metrics

5.1. Health Goal: Defined health goal for the user (Health, Fitness, Performance, etc. goal)

5.2. Healthline: User health condition or treatments, if any restrictions

5.3. BMI: Simplistic measurement of a healthy range of weight related to height

5.4. RHR: The resting heart rate of the user

5.5. A1C Reading: Measure of the average level of blood sugar over a given time

5.6. Dietary Restrictions: List of foods that must be left out of a person's diet

5.7. Diet Plans: Categories and quantities of food that a person eats for a health outcome

- Diabetes Management: Specific diet to manage diabetes
- Ketogenic: A diet high in fats influences metabolism
- Vegan: Diet consisting of no animal products
- Vegetarian: Diet consisting of mainly plant sources
- Paleo: Diet focussing on whole foods with high protein, low carbohydrates

6. Health Recommendations

6.1. Nutrition recommendation: A list of nutrients that will help user achieve their set goal

6.2. Meal recommendation: A list of meals or recipes that contain recommended nutrition

- Daily Recommendation: Recommendation for that day
- Weekly Recommendation: Recommendation for that week
- Monthly: Recommendation for that Month

6.3. Workout recommendation: A prepared workout that will help the user achieve their goal

7. Tracking/Logging/Analytics

7.1. Weight: The weight that the user registered that day

7.2. Daily Calories: The number of calories the user consumed that day

- Daily Protein: The amount of protein consumed that day

- Daily Carbohydrate: The amount of carbohydrates consumed that day
- Daily Fat: The amount of fat consumed that day

7.3. Exercise time: The amount of time the user exercised that day

7.4. Stress level: A 1-10 rating of the level of stress the user experienced that day

2. Prioritized Functional Requirements

1. User Account Management
1.1. A user shall securely sign up.
1.2. A user shall securely sign in.
1.3. A user should be able to create their profile (age, weight, height, fitness goals).
1.4. A user should be able to do password recovery/reset.
1.5. A user should be able to update their account.
1.6 A user should be able to delete their account.
1.7. A user shall be able to upload and manage profile pictures.
1.8. Users should be able to establish and monitor fitness goals with the app.

2. Social Features
2.1. Users shall be able to like, comment and share other user's posts.
2.2 A user shall be able to create posts.
2.3. A user shall be able to go back and edit posts.
2.4. A user shall be able to go back and delete posts.
2.5. Users should receive notifications when other users like their posts.
2.6. Users should receive notifications when other users comment on their posts.
2.7. The app shall provide moderation tools to manage community interactions and ensure a positive and supportive environment.

3. Tracking
3.1. A user shall be able to track their calories burned throughout their day.
3.2. A user shall be able to track their total steps throughout their day.
3.3. A user should be able to log their meals and snacks throughout their day.
3.4. The app shall provide feedback on the user's nutritional intake (calories, macronutrients carbohydrates, proteins, fats, vitamins, and minerals).

4. Analytics
4.1 The app shall provide information about the user's health.
4.2. The app shall provide messages in intervals to check up on the user.
4.3. A user shall be able to view statistics about their health & workout.
4.4. The app will provide an option for feedback so that the developers would be able to update the program.

5. Meal planning
5.1. The app shall offer customized meal recommendations based on the users' tastes and dietary constraints.
5.2. The application should produce meal plans.

5.3. Users should be able to personalize their meal plans using the app.
5.4. All meal nutritional information should be provided by the app.
5.5. Cooking directions and recipe recommendations should be included in the app.
5.6. The app shall allow customers to track their meals and read barcodes on packaged goods.
5.7. The app should provide users recipes.
5.8. All recipes shall list the nutritional values per serving.

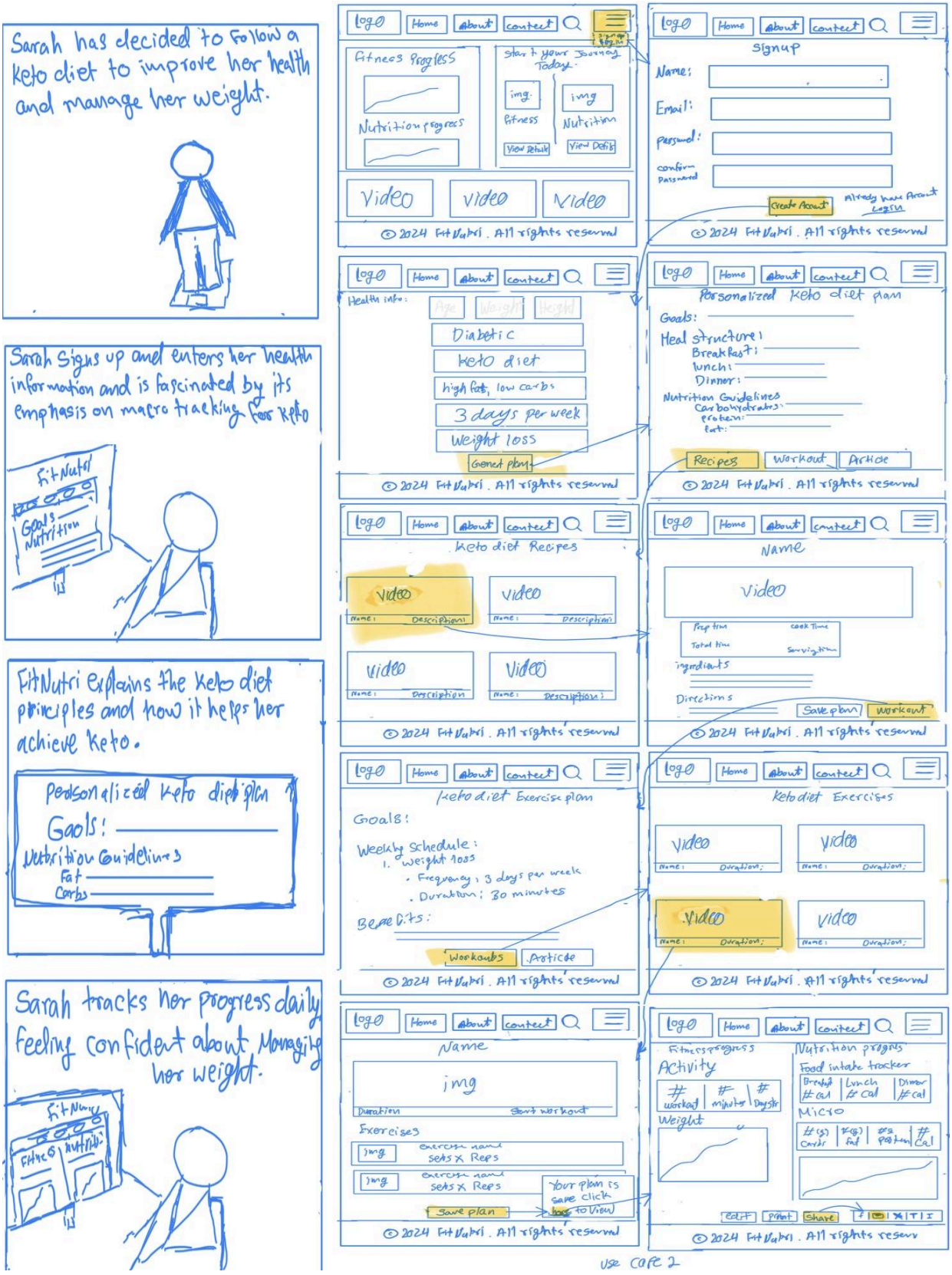
6. Workout Planning
6.1. A user shall be able to create personalized workout plans based on their fitness goals.
6.2. The app shall suggest pre-designed workout routines categorized by fitness level (beginner, intermediate, advanced) and specific goals (weight loss, muscle gain, endurance).
6.3. Customers shall be able to schedule workouts.
6.4. Customers shall be able to set reminders in the app.
6.5. Customers shall be able to track completed exercises.
6.6. A user shall have the option to follow live-streaming workout sessions led by fitness instructors.

7. Monitoring Fitness and Health
7.1. The app must work with other fitness tracker apps and devices.
7.2. The customers' physical activity must be tracked by the app.
7.3. The customers' physical activity must be linked to their dietary intake.
7.4. Taking into account the customers' activity levels, the app ought to offer insights and suggestions.
7.5. Customers should be able to establish and monitor fitness goals with the app.
7.6. The software ought to alert customers to their accomplishments and activity benchmarks.
7.7. customers should be able to check their overall health and nutritional status on a dashboard provided by the app.
7.8. The app needs to provide pointers and guidance on keeping an active lifestyle and a balanced diet.
7.9. Physical activity manual entry should be supported by the app.
7.10. Customers ought to be able to record their physical dimensions and weight on the app.

3. UI Mockups and Storyboards (high level only)

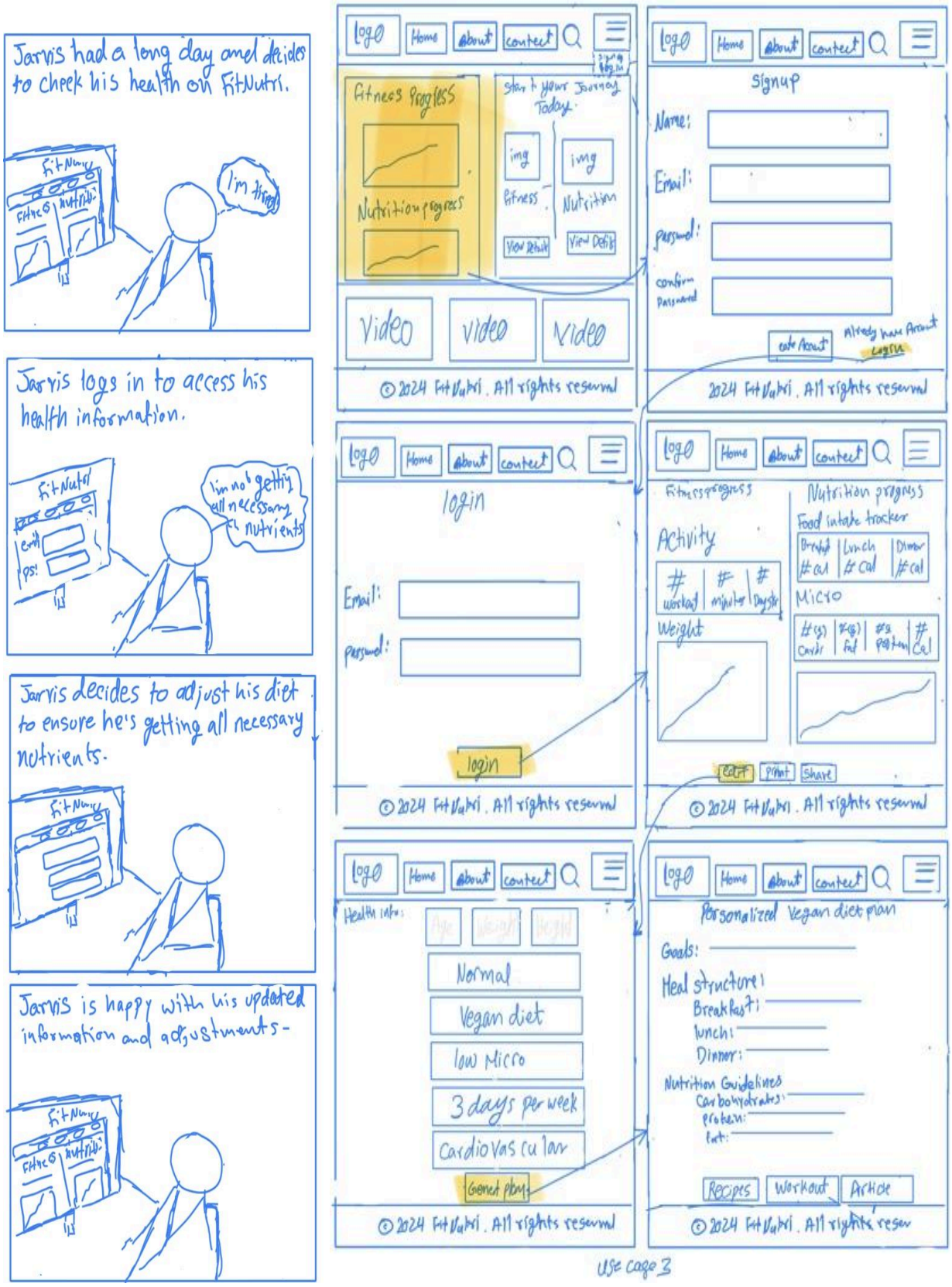
1. Use case 1:

Use case 2:



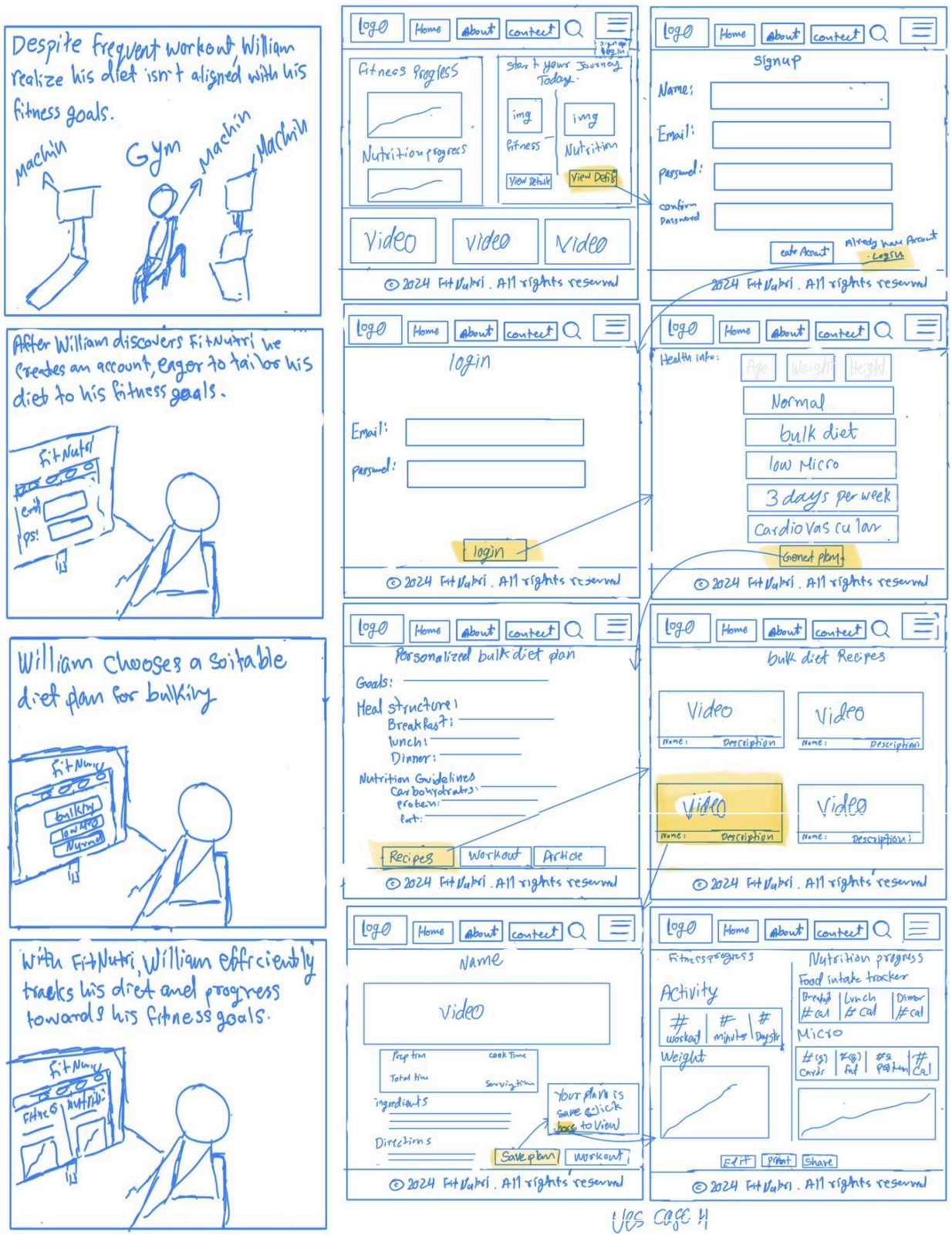
use case 2

Use case 3:

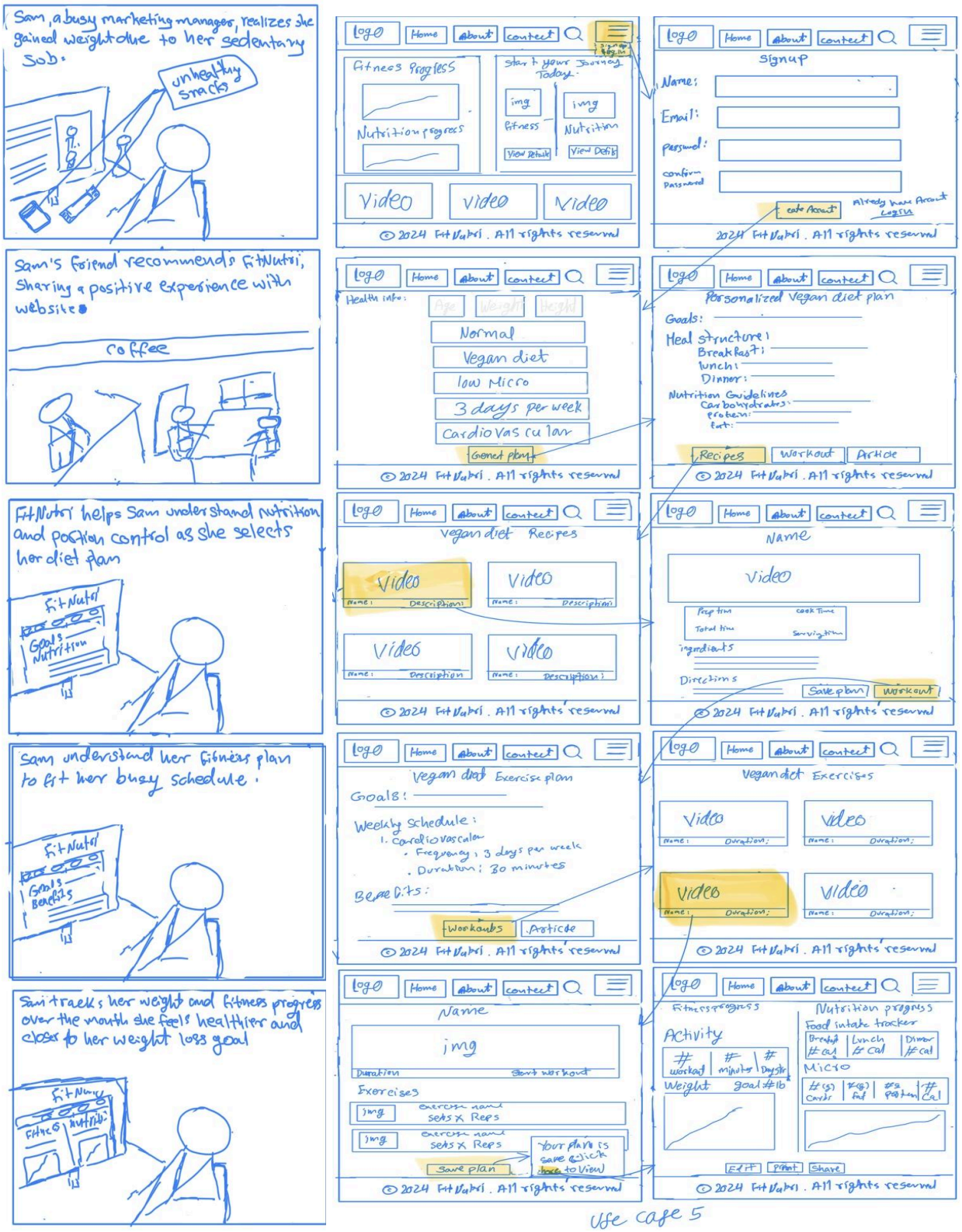


Use case 3

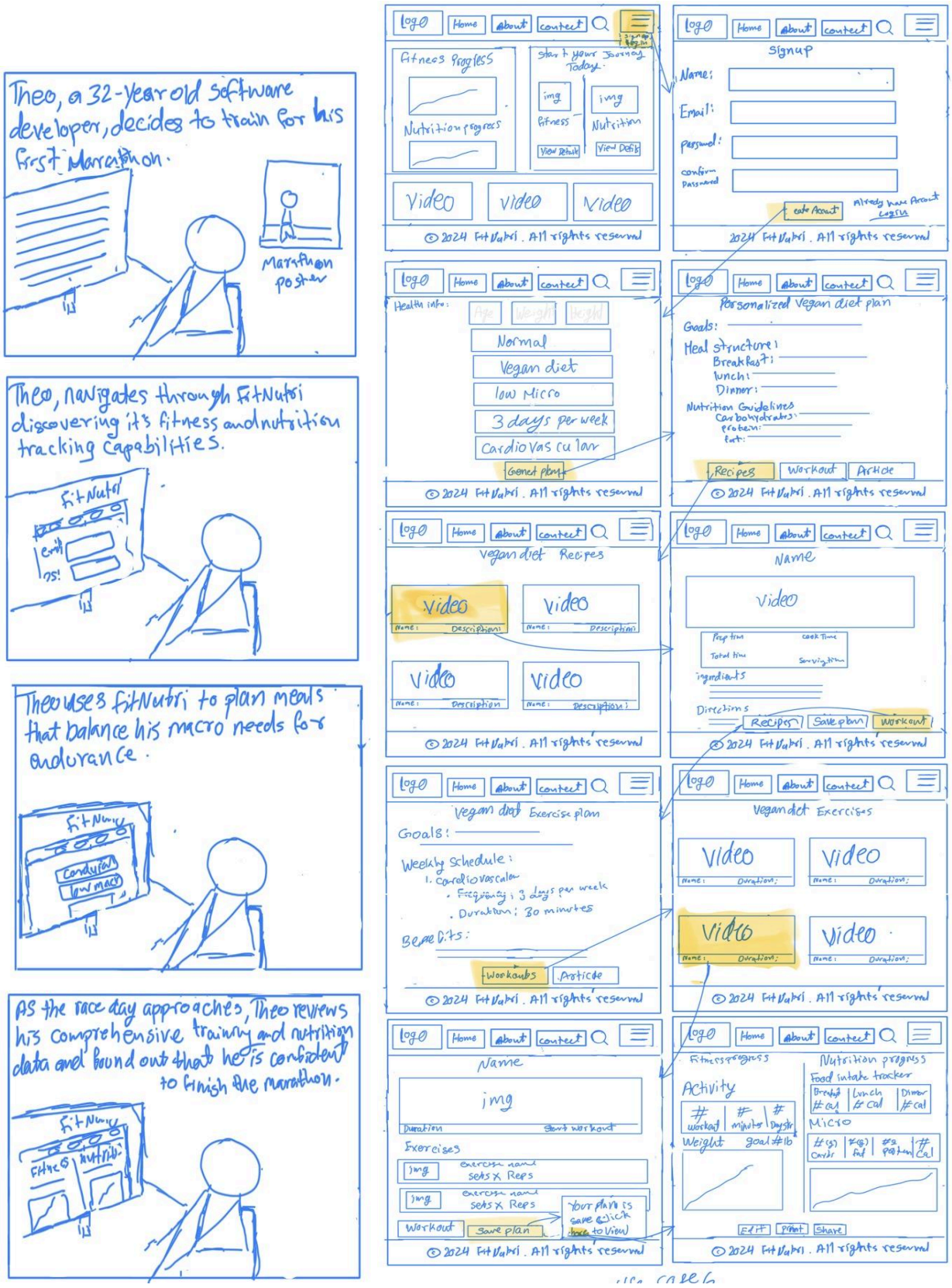
Use case 4:


Despite frequent workout, William realize his diet isn't aligned with his fitness goals.
Gym
After William discovers FitNutri he creates an account, eager to tailor his diet to his fitness goals.
William chooses a suitable diet plan for bulking
With FitNutri, William effciently tracks his diet and progress towards his fitness goals.
Logo
Home
About
Contact
Fitness Progress
Nutrition progress
Start your Journey Today
Video
Signup
Name:
Email:
Password:
Confirm Password
Create Account
Already have Account
Login
login
Health info:
Normal
bulk diet
low Micro
3 days per week
Cardiovascular
Genert plan
Personalized bulk diet plan
Goals:
Meal structure:
Breakfast:
lunch:
Dinner:
Nutrition Guidelines
Carbohydrates:
protein:
fat:
Recipes
Workout
Article
bulk diet Recipes
Name:
Description
Name
Prep time
cook Time
Total time
Servings
ingredients
Directions
Your plan is save click here to View
Save plan
Workout
Fitness progress
Nutrition progress
Activity
Food intake tracker
Micro
Weight
EDIT
Print
Share
© 2024 FitNutri. All rights reserved
Use Case 4

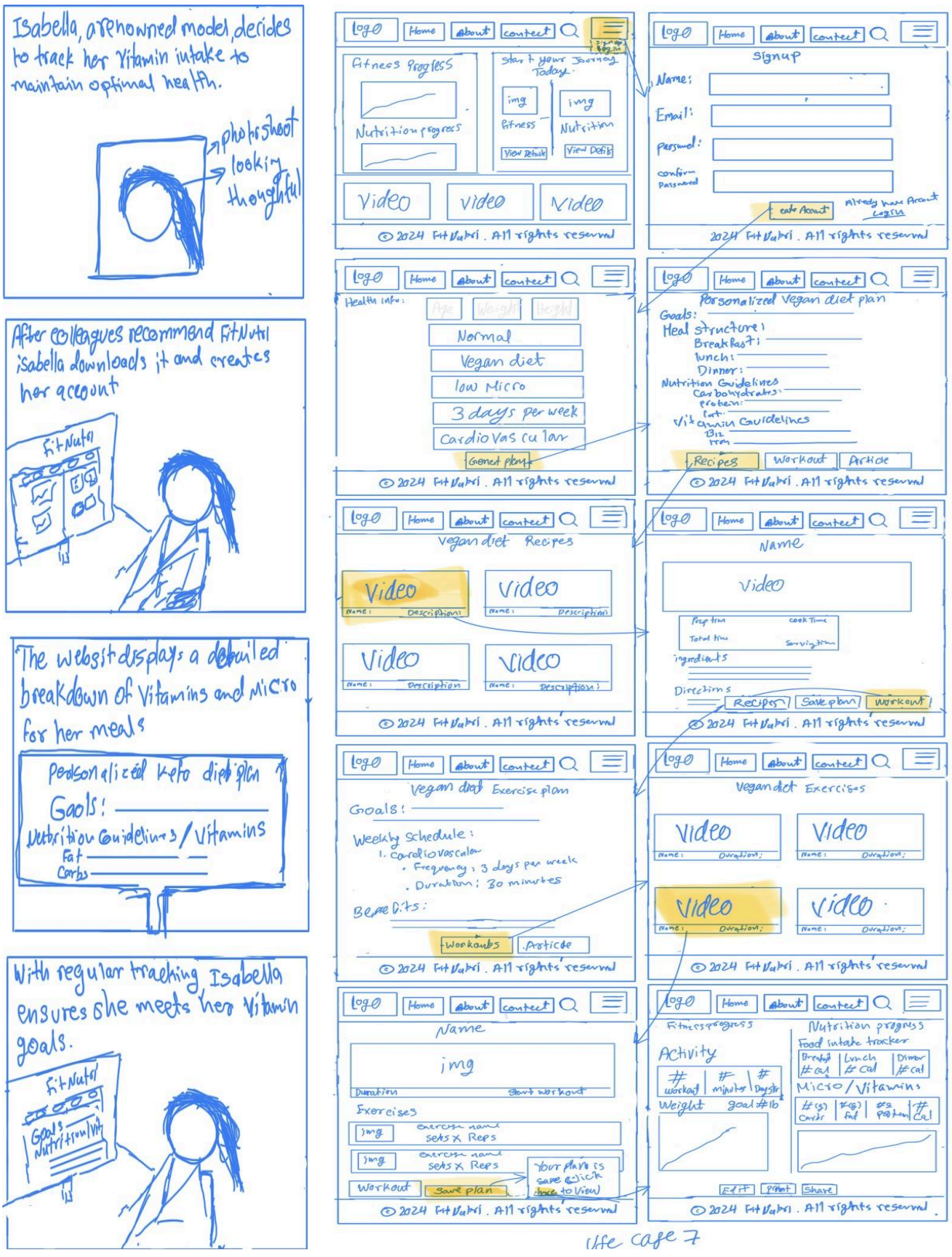
Use case 5:


Sam, a busy marketing manager, realizes she gained weight due to her sedentary job.
unhealthy snacks
Sam's friend recommends FitNutri, sharing a positive experience with website.
coffee
FitNutri helps Sam understand nutrition and portion control as she selects her diet plan
Sam understand her fitness plan to fit her busy schedule.
Sam tracks her weight and fitness progress over the month she feels healthier and closer to her weight loss goal
logo
Home
about
content
Fitness Progress
Nutrition progress
Start your Journey Today
Video
© 2024 FitNutri. All rights reserved
Signup
Name:
Email:
Password:
confirm Password
Create Account
Already have Account Login
Health info:
Age
Weight
Height
Normal
Vegan diet
low Micro
3 days per week
Cardiovascular
Genret plan
Personalized Vegan diet plan
Goals:
Meal structure:
Breakfast:
lunch:
Dinner:
Nutrition Guidelines
Carbohydrates:
Protein:
Fat:
Recipes
Workout
Article
Vegan diet Recipes
Name:
Description:
Name
Prep time
cook Time
Total time
Servings
Ingredients
Directions
Save plan
Workout
Vegan diet Exercise plan
Goals:
Weekly Schedule:
1. Cardiovascular
• Frequency: 3 days per week
• Duration: 30 minutes
Benefits:
Workouts
Article
Vegan diet Exercises
Duration:
Name
img
Duration
Start workout
Exercises
exercise name
sets X Reps
Save plan
Your plan is save quick here to View
Fitness progress
Activity
workout
minutes
Day str
Weight
goal #lb
Nutrition progress
Food intake tracker
Breakfast
Lunch
Dinner
#cal
Micro
Carbs
Fat
Protein
Exit
Print
Share
Use case 5

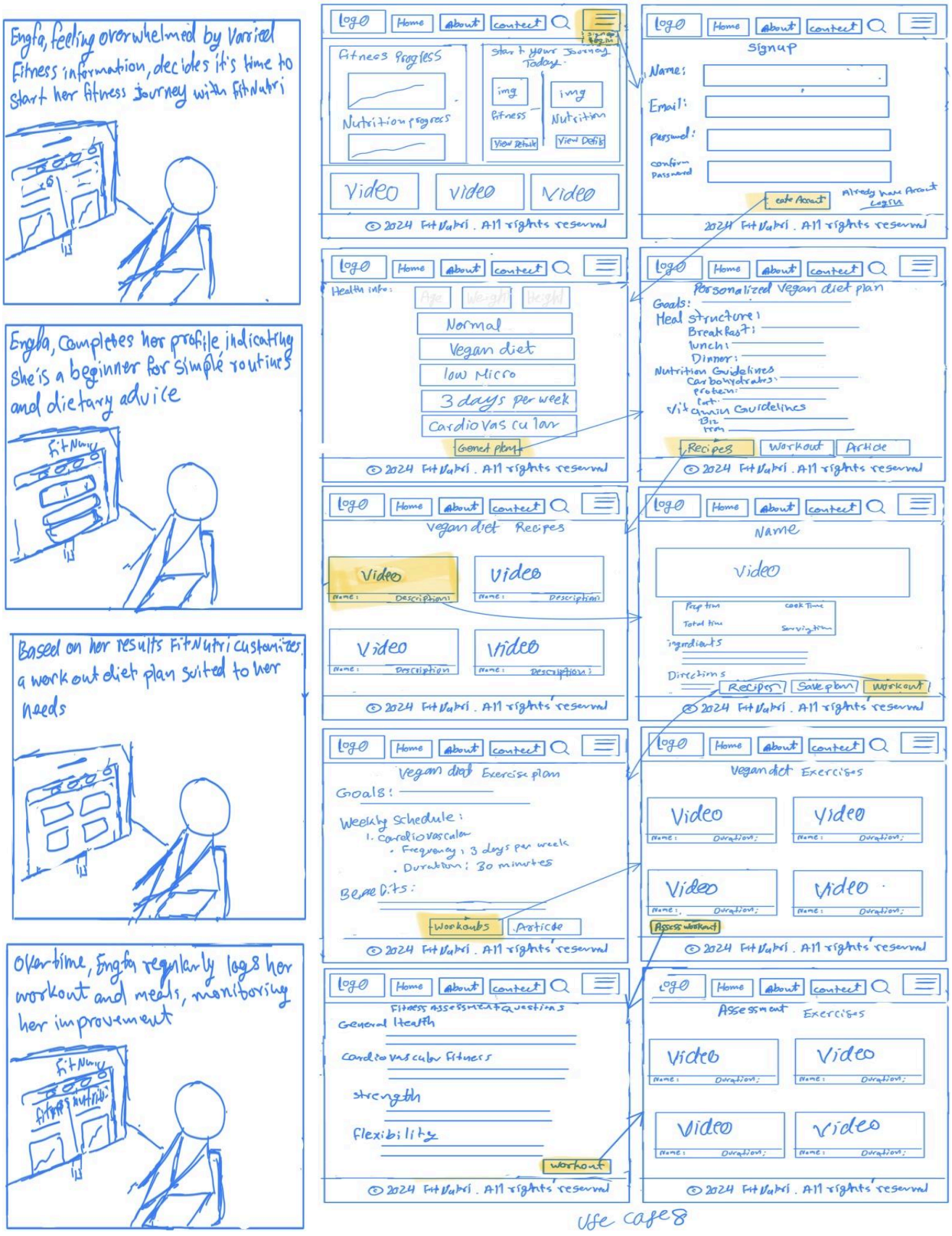
Use case 6:


Theo, a 32-year old software developer, decides to train for his first Marathon.
Marathon poster
Theo, navigates through FitNutri discovering it's fitness and nutrition tracking capabilities.
Theo uses fitNutri to plan meals that balance his macro needs for endurance.
As the race day approaches, Theo reviews his comprehensive training and nutrition data and found out that he is confident to finish the marathon.
logo
Home
about
content
Fitness Progress
Nutrition progress
Start your Journey Today
Fitness
Nutrition
View Details
Video
© 2024 FitNutri. All rights reserved
Signup
Name:
Email:
password:
confirm password
Create Account
Already have Account Login
Health info:
Age
Weight
Height
Normal
Vegan diet
low Micro
3 days per week
Cardiovascular
Genrat plan
Personalized Vegan diet plan
Goals:
Meal structure:
Breakfast:
lunch:
Dinner:
Nutrition Guidelines
Carbohydrates:
protein:
fat:
Recipes
Workout
Article
Vegan diet Recipes
Name:
Description:
Name
Prep time
Cook Time
Total time
Servings
ingredients
Directions
Recipes
Save plan
Workout
Vegan diet Exercise plan
Weekly schedule:
1. cardiovascular
. Frequency: 3 days per week
. Duration: 30 minutes
Benefits:
Workouts
Article
Vegan diet Exercises
Duration:
img
Exercises
exercise name
sets x Reps
Workout
Save plan
Your plan is save click here to View
Fitness progress
Nutrition progress
Activity
workout
minutes
Days/wk
Food intake tracker
Breakfast
Lunch
Dinner
#cal
Micro
Weight
goal #lb
Edit
Print
Share

Use case 7:

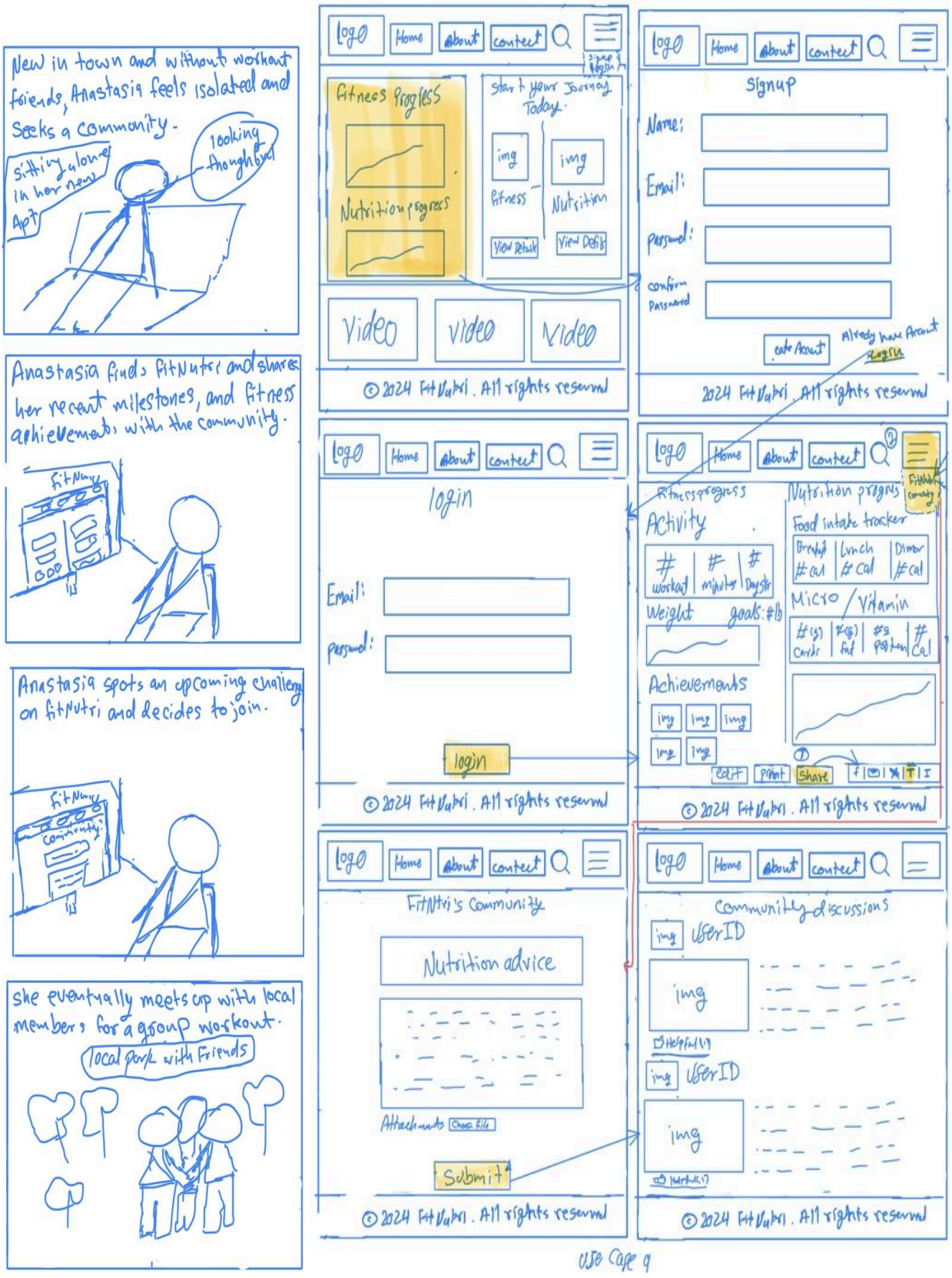

Isabella, a renowned model, decides to track her Vitamin intake to maintain optimal health.
photoshoot looking thoughtful
After colleagues recommend FitNutri isabella downloads it and creates her account
FitNutri
The website displays a detailed breakdown of Vitamins and Micro for her meals
Personalized keto diet plan
Goals:
Nutrition Guidelines / Vitamins
Fat
Carbs
With regular tracking Isabella ensures she meets her Vitamin goals.
FitNutri
Goals
Nutrition info
logo
Home
about
content
Fitness Progress
Nutrition progress
Start your Journey Today
img
Fitness
Nutrition
View Details
View Details
Video
video
Video
© 2024 FitNutri. All rights reserved
Signup
Name:
Email:
Password:
confirm Password
Create Account
Already have Account Login
Health info:
Age
Weight
Height
Normal
Vegan diet
low Micro
3 days per week
Cardiovascular
Generate plan
Personalized Vegan diet plan
Goals:
Meal structure:
Breakfast:
lunch:
Dinner:
Nutrition Guidelines
Carbohydrates:
protein:
fat:
vitamin Guidelines
Recipes
Workout
Article
Vegan diet Recipes
Video
Name:
Description
Name
Prep time
Cook Time
Total time
Servings
ingredients
Directions
Recipes
Save plan
Workout
Vegan diet Exercise plan
Goals:
Weekly Schedule:
1. Cardiovascular
• Frequency: 3 days per week
• Duration: 30 minutes
Benefits:
Workouts
Article
Vegan diet Exercises
Duration:
Video
Name
img
Duration
Start workout
Exercises
exercise name
sets X Reps
Workout
Save plan
Your plan is saved click to View
Fitness progress
Nutrition progress
Activity
workout
minutes
Days
Food intake tracker
Breakfast
Lunch
Dinner
Cal
Micro / Vitamins
Weight
goal #lb
Edit
Print
Share
Use case 7

Use case 8:



Use case 8

Use case 9:


New in town and without workout friends, Anastasia feels isolated and seeks a community.
sitting alone in her new Apt
looking thoughtful
Anastasia finds FitNutri and shares her recent milestones, and fitness achievements with the community.
FitNutri
Anastasia spots an upcoming challenge on fitNutri and decides to join.
FitNutri
Community
she eventually meets up with local members for a group workout.
local park with Friends
Logo
Home
About
contact
Fitness Progress
Nutrition progress
Start your Journey Today
img
Fitness
Nutrition
View Details
View Details
Video
Video
Video
© 2024 FitNutri. All rights reserved
Signup
Name:
Email:
password:
confirm password
Create Account
Already have Account
Login
2024 FitNutri. All rights reserved
login
Email:
password:
login
© 2024 FitNutri. All rights reserved
Fitness progress
Activity
workout
minutes
Days/wk
Weight
goals: #lb
Achievements
Nutrition progress
Food intake tracker
Breakfast
Lunch
Dinner
cal
Micro / Vitamin
Carbs
Fat
Protein
Edit
Print
Share
© 2024 FitNutri. All rights reserved
FitNutri's Community
Nutrition advice
Attachments
Choose File
Submit
© 2024 FitNutri. All rights reserved
Community discussions
img
UserID
Helpful(1)
Use Case 9

4. High-Level DB Architecture & Organization

a. User (Strong)
   i. Attributes:
   UserID(PK)/Username/Email/Password/DateOfBirth/Gender/AccountID(FK)
      1. Each user shall have exactly one account
      2. A user shall log into their account using secure authentication.
      3. A user shall fill out one health information form.
      4. A user shall be able to choose many types of nutrition
      5. A user shall be able to choose many types of workouts
      6. Users shall write many dietary restrictions
      7. A user shall generate one personal nutrition plan
      8. A user shall generate one personal workout plan
      9. A user shall be able to share many personal workout plans on social media.
      10. A user shall be able to share many personal nutrition plans on social media.
      11. A user shall view many recipes
      12. A user shall view many exercises
      13. Users shall submit many forms and attach files to these forms.
      14. Users shall view forms from other users.
      15. Users shall like and receive notifications for many posts.
      16. Users shall customize their notification settings.
      17. Users shall search for recipes and workouts using keywords.
      18. Users shall be able to share, edit and print fitness and nutrition progress many times.

b. Account (Strong)
   i. Attributes:
   AccountID(PK)/UserID(FK)/Email/Password/RecoveryEmail/RecoveryPhone/ProfileUpdateTime
      1. An account shall belong to one user at most
      2. An account shall be associated with one email at most

c. 3. Health Information(weak)
   i. Attributes:
   HealthInfoID(PK)/UserID(FK)/CurrentWeight/GoalWeight/FitnessLevel/DietaryPreferences/NutritionTypeID(FK)/WorkoutTypeID(FK)
      1. A health information shall be filled by one user
      2. A health information shall contain many Types of Nutrition
      3. A health information shall contain many Types of workout
      4. A health information shall contain one article base on personal nutrition plan

5. A health information shall contain one article base on personal fitness plan
6. A Health information shall include a history of user health data over time.
7. A Health information shall include user goals related to weight, fitness level, and dietary preferences

d. Types of Nutrition(weak)
   i. Attributes: NutritionTypeID (PK)/Description/RecommendedCalories
      1. A type of Nutrition shall be chosen by many users
      2. A type of Nutrition shall contain one article
e. Types of Workout(weak)
   i. Attributes: WorkoutTypeID (PK)/Description/ExpectedCalorieBurn
      1. A type of workout shall be chosen by many users
      2. A type of workout shall contain one article
f. Exercises(Weak)
   i. Attributes:WorkoutID (PK)/WorkoutName/Duration/IntensityLevel/CaloriesBurnedEstimate, WorkoutTypeID (FK)
      1. An exercise shall include instructional videos and images
      2. An exercise shall provide estimated calories burned
      3. An exercise shall be viewed by many users
g. Recipe(Weak)
   i. Attributes:RecipeID(PK)/DietType/PreparationTime/CookingTime/IngredientList/UserRatingAverage/NutritionTypeID(FK)
      1. A recipe shall consist of many dishes
      2. A recipe shall be categorized by diet type (ex., keto, vegan)
      3. A recipe shall allow users to comment and rate its effectiveness and taste
      4. A recipe shall be viewed by many users
      5. A recipe shall include instructional videos and images
h. Personal nutrition plan(weak)
   i. Attributes: PlanID (PK)/UserID (FK)/PlanDetails/CreationDate
      1. A personal nutrition plan shall be generated by many users
      2. A personal nutrition plan shall be shared by many users
i. Personal fitness plan(weak)
   i. Attributes: FitnessPlanID (PK)/UserID (FK)/WorkoutDetails/StartDate/EndDate
      1. A personal workout plan shall be generated be many users
      2. A personal workout plan shall be shared be many users and social media

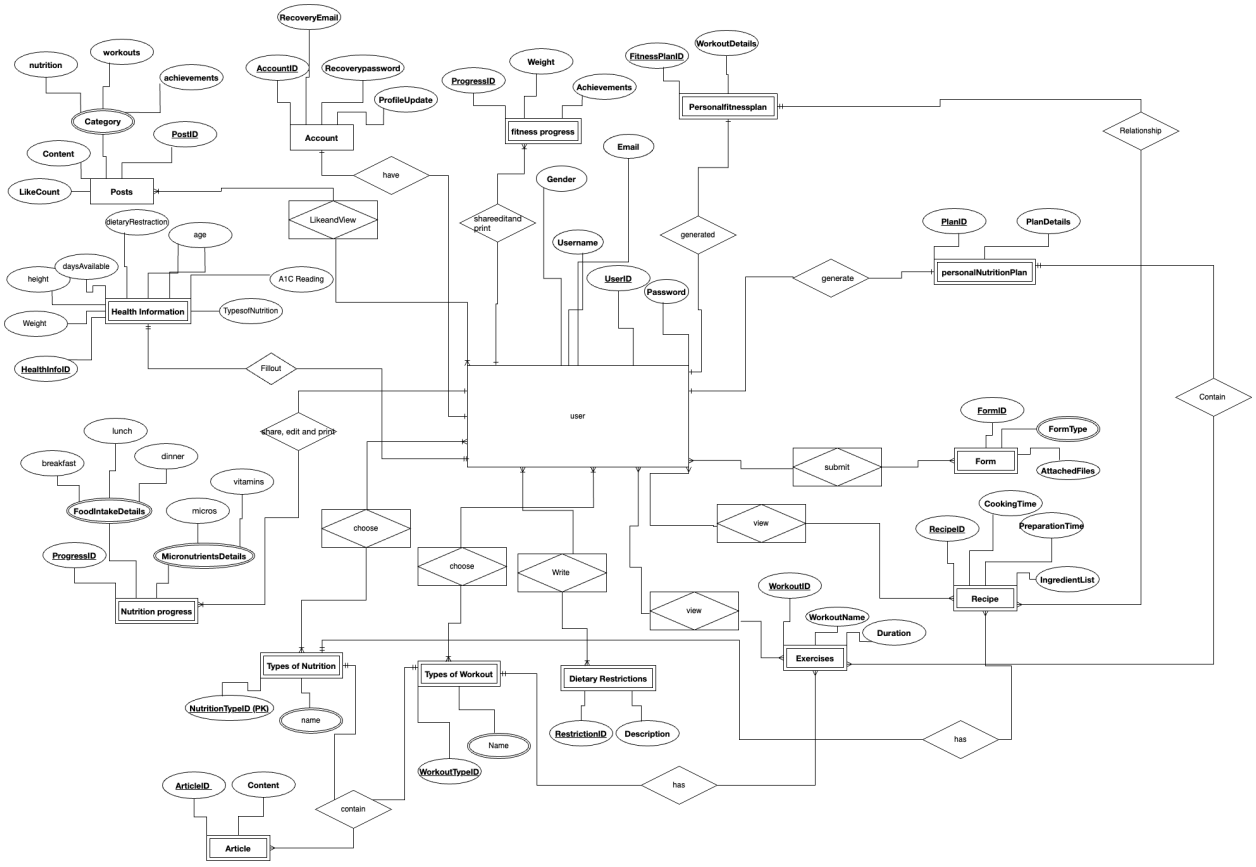
j. Dietary Restrictions(weak)
    i. Attributes: RestrictionID (PK)/UserID (FK)/RestrictionType/Description
        1. A Dietary Restrictions shall be written by many users
k. Assess(weak)
    i. Attributes: AssessID (PK)/UserID (FK)/DateAssessed/AssessDetails
        1. An assess shall be generated by many users
l. Form(weak)
    i. Attributes:FormID(PK)/UserID(FK)/SubmissionDate/FormType/Attached Files (count of files attached)
        1. A form shall be submitted by one users
        2. A form shall contain many files sent by the users
        3. A form shall be categorized by topic (ex., nutrition, workouts, achievements)
        4. A form shall allow users to request support or submit feedback directly to the service team
m. Posts(strong)
    i. Attributes: PostID (PK)/UserID (FK)/PostDate/Content (textual content of the post)/Category (e.g., nutrition, workouts, achievements)/LikeCount (number of likes received)
        1. A post shall be liked by many users
        2. A post shall allow users to engage through comments and replies
        3. Posts can be categorized by topic (Ex, nutrition, workouts, achievements)
n. Notifications(strong)
    i. Attributes: NotificationID (PK), UserID (FK), NotificationType, NotificationDate, NotificationContent
        1. Notifications shall be received by many users for likes and comments on their posts
o. Article(weak)
    i. Attributes:ArticleID (PK), RelatedToNutritionTypeID (FK), RelatedToWorkoutTypeID (FK), Content
        1. An article shall contain one type of nutrition
        2. An article shall contain one type of workout
p. Nutrition progress
    i. Attributes: ProgressID (PK), UserID (FK), Date, FoodIntakeDetails, MicronutrientsDetails
        1. Nutrition progress shall be editable by one user
        2. Nutrition progress shall display food intake(ex., breakfast, lunch ,dinner), micros and vitamins to one user
q. Fitness progress

i. Attributes: ProgressID (PK), UserID (FK), Date, ActivitiesDetails, Weight, Achievements
    1. Fitness progress shall be editable by one user
    2. Fitness progress shall display activity(ex. number of workout, minutes and days of strike), weight, lb and achievements to one user

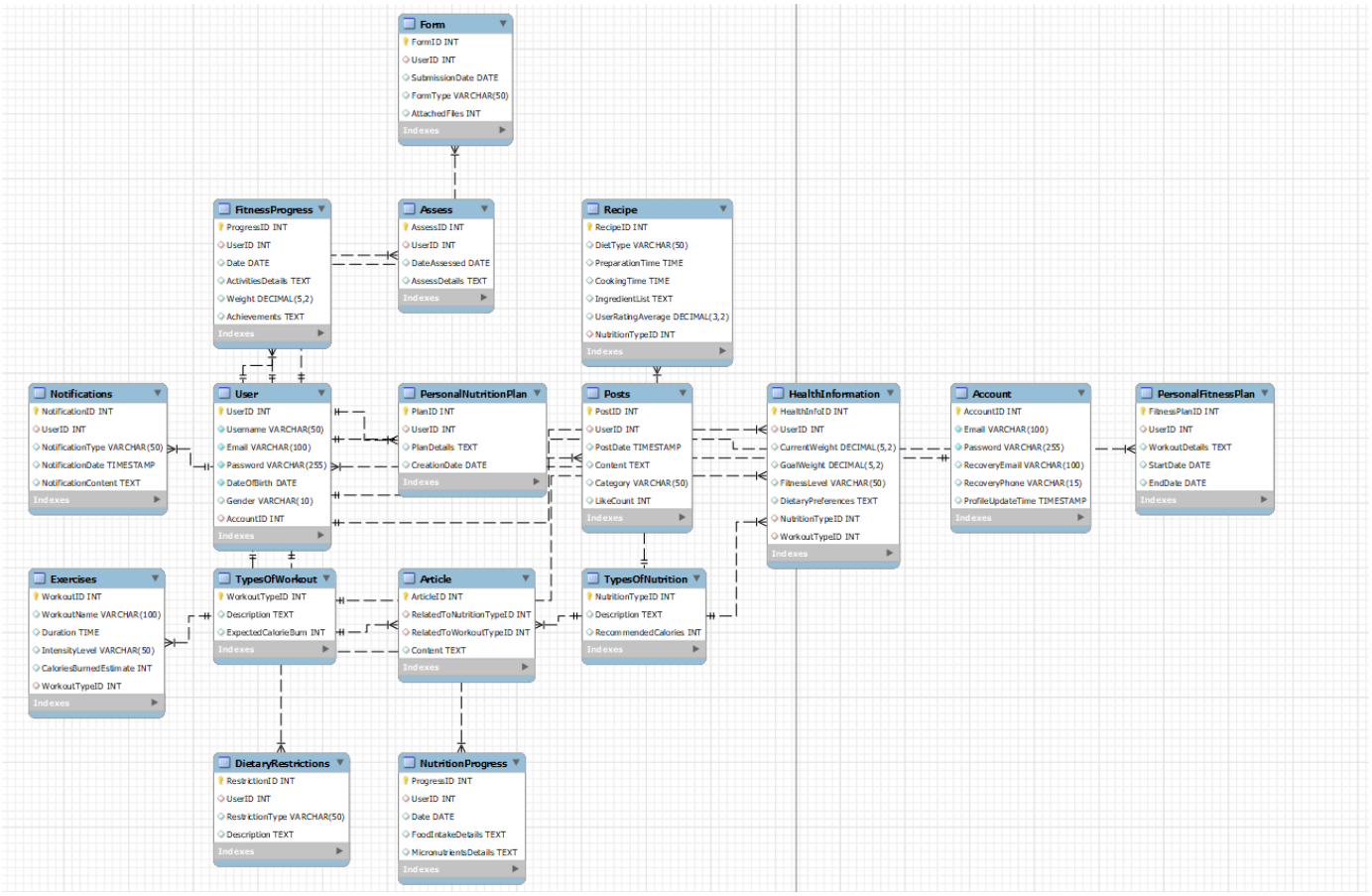
Define the DBMS

We decided to use MYSQL as the database management system for the project due to its simplicity and ease of use. Integration of anything needed is a breeze and very smooth. Frameworks and programming languages are essentially global with MYSQL which makes using it very easy.

ERD:


RecoveryEmail
workouts
nutrition
achievements
AccountID
Recoverypassword
ProfileUpdate
ProgressID
Weight
FitnessPlanID
WorkoutDetails
Achievements
Category
PostID
Account
fitness progress
Personalfitnessplan
Relationship
Content
Email
have
LikeCount
Posts
Gender
dietaryRestraction
age
LikeandView
shareeditand print
generated
PlanID
PlanDetails
Username
daysAvailable
height
A1C Reading
UserID
Password
generate
personalNutritionPlan
Weight
Health Information
TypesofNutrition
HealthInfoID
Fillout
user
Contain
FormID
lunch
share, edit and print
FormType
breakfast
dinner
submit
Form
vitamins
AttachedFiles
FoodIntakeDetails
micros
CookingTime
choose
view
RecipeID
PreparationTime
ProgressID
MicronutrientsDetails
choose
Write
IngredientList
WorkoutID
Recipe
view
Nutrition progress
WorkoutName
Duration
Types of Nutrition
Types of Workout
Dietary Restrictions
Exercises
NutritionTypeID (PK)
name
Name
RestrictionID
Description
has
WorkoutTypeID
has
ArticleID
Content
contain
Article

EER:


Form
FormID INT
UserID INT
SubmissionDate DATE
FormType VARCHAR(50)
AttachedFiles INT
Indexes
FitnessProgress
ProgressID INT
UserID INT
Date DATE
ActivitiesDetails TEXT
Weight DECIMAL(5,2)
Achievements TEXT
Indexes
Assess
AssessID INT
UserID INT
DateAssessed DATE
AssessDetails TEXT
Indexes
Recipe
RecipeID INT
DietType VARCHAR(50)
PreparationTime TIME
CookingTime TIME
IngredientList TEXT
UserRatingAverage DECIMAL(3,2)
NutritionTypeID INT
Indexes
Notifications
NotificationID INT
UserID INT
NotificationType VARCHAR(50)
NotificationDate TIMESTAMP
NotificationContent TEXT
Indexes
User
UserID INT
Username VARCHAR(50)
Email VARCHAR(100)
Password VARCHAR(255)
DateOfBirth DATE
Gender VARCHAR(10)
AccountID INT
Indexes
PersonalNutritionPlan
PlanID INT
UserID INT
PlanDetails TEXT
CreationDate DATE
Indexes
Posts
PostID INT
UserID INT
PostDate TIMESTAMP
Content TEXT
Category VARCHAR(50)
LikeCount INT
Indexes
HealthInformation
HealthInfoID INT
UserID INT
CurrentWeight DECIMAL(5,2)
GoalWeight DECIMAL(5,2)
FitnessLevel VARCHAR(50)
DietaryPreferences TEXT
NutritionTypeID INT
WorkoutTypeID INT
Indexes
Account
AccountID INT
Email VARCHAR(100)
Password VARCHAR(255)
RecoveryEmail VARCHAR(100)
RecoveryPhone VARCHAR(15)
ProfileUpdateTime TIMESTAMP
Indexes
PersonalFitnessPlan
FitnessPlanID INT
UserID INT
WorkoutDetails TEXT
StartDate DATE
EndDate DATE
Indexes
Exercises
WorkoutID INT
WorkoutName VARCHAR(100)
Duration TIME
IntensityLevel VARCHAR(50)
CaloriesBurnedEstimate INT
WorkoutTypeID INT
Indexes
TypesOfWorkout
WorkoutTypeID INT
Description TEXT
ExpectedCalorieBurn INT
Indexes
Article
ArticleID INT
RelatedToNutritionTypeID INT
RelatedToWorkoutTypeID INT
Content TEXT
Indexes
TypesOfNutrition
NutritionTypeID INT
Description TEXT
RecommendedCalories INT
Indexes
DietaryRestrictions
RestrictionID INT
UserID INT
RestrictionType VARCHAR(50)
Description TEXT
Indexes
NutritionProgress
ProgressID INT
UserID INT
Date DATE
FoodIntakeDetails TEXT
MicronutrientsDetails TEXT
Indexes

5. High-Level APIs and Main Algorithms

5.1. APIs Exposed

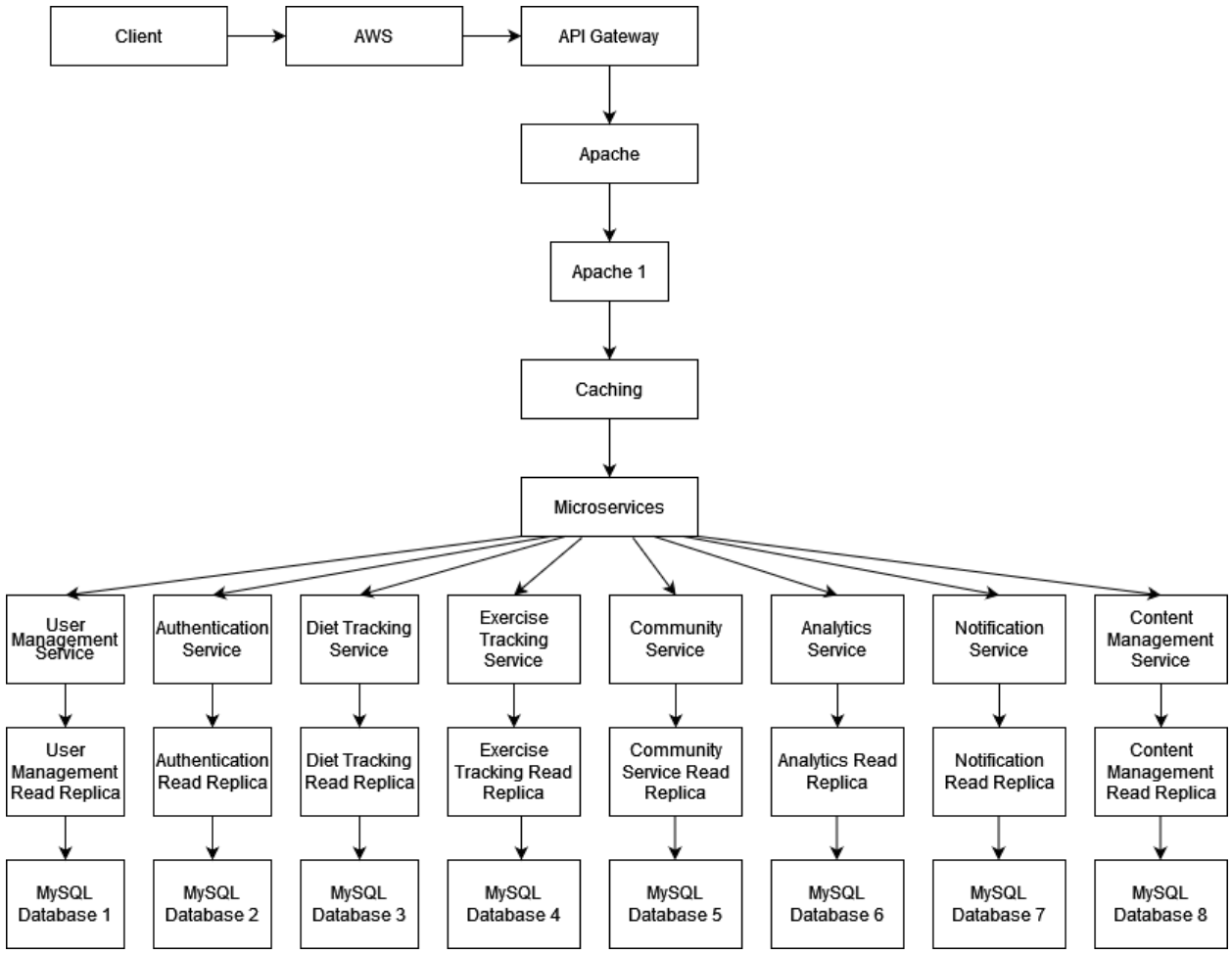
1. **User Management API**: This API will handle user related operations such as:
    a. Endpoints for registration, login, profile retrieval, update, and deletion.
    b. Authentication endpoints for user login and token management.
    c. Authorization mechanisms to control access to user data.
2. **Nutrition Tracking API**: This API will manage food logging and nutrition data:
    a. Endpoints for searching, retrieving, and logging food items.
    b. Functionality to calculate and retrieve nutritional information based on logged food items.
    c. Integration with external food databases or APIs for comprehensive food data.
3. **Workout Planning API**: This API will handle workout creation, scheduling, and tracking:
    a. Endpoints for creating, retrieving, updating, and deleting workout plans.
    b. Possible/future integration with wearable devices or fitness trackers to import and synchronize workout data.
4. **Community Engagement API**: This API will handle social interactions within the app:
    a. Endpoints for creating, retrieving, updating, and deleting posts, comments, and likes/encouragements.
    b. Functionality to manage user groups and group challenges (motivation inducing mechanism).
5. **Content Management API**: This API will handle the management of static content such as articles, recipes, and educational resources:
    a. Endpoints for managing recipes and their ingredients, instructions, and nutritional information.

5.2 Main Algorithms

1. **Personalized Recommendations** that will deliver users specific workout plans, dietary insight, recommendations, and other experiences based on their experience and goals.
2. **Content-based filtering** so we can show the user things they might be interested in. By analyzing the user's workout history, dietary preferences, and stated goals, FitNutri could recommend similar workouts, recipes, or meal plans that align with their patterns or choices. Some users might have a preferred cuisine (subject to change) or type of workout.

3. **Keyword-Based Search** will allow users to search using keywords related to workouts, recipes, ingredients, dietary restrictions, and fitness goals.
4. **Faceted Search** will allow users to further refine their search results using multiple filters, enchanting the search experience. For example, users could filter recipes by dietary restrictions, cuisine (subject to change), ingredients, and calorie range.
5. **Trending Content** algorithms could be used to identify and highlight trending topics in the community section of FitNutri. This would surface engaging content and promote community involvement.
6. **Personalized Content Feeds** that, based on a user's interests, connections, and activity, could display a personalized community feed. This personalized feed would prioritize content from groups or users they follow and engage with.

6. System Design



Summary of Components:

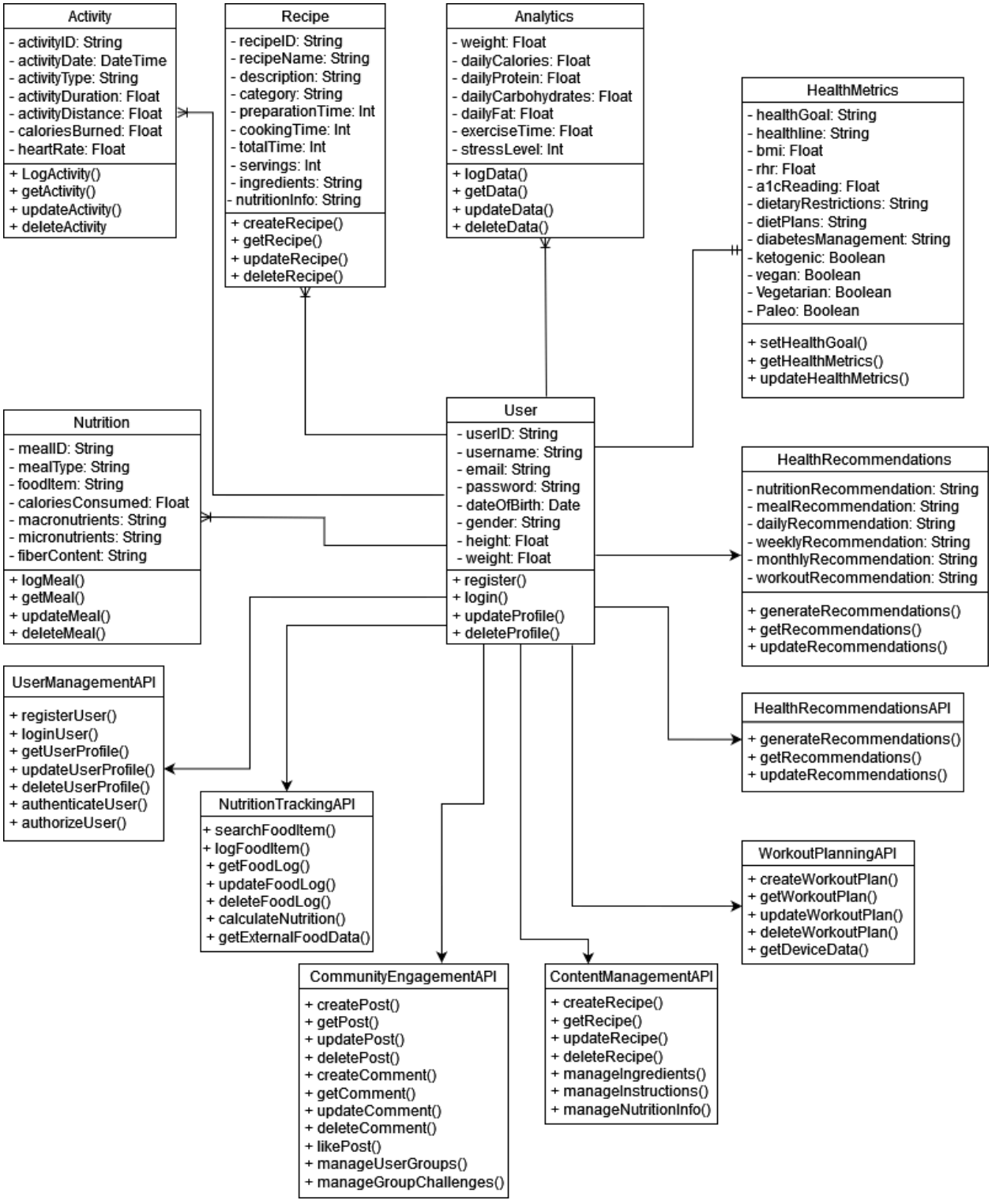
The FitNutri site will make use of several microservices which include user management, authentication, diet, and exercise. Breaking down the services in this way will allow for easier maintenance and scaling.

Load balancing will be handled using multiple servers and microservices so that servers will be able to distribute traffic as needed.

Caching will be used to prevent prevalent requests from always needing server response. By caching user information and other important features, the load on servers will be reduced further.

Database replication will be used to increase reliability by allowing data to still be available even if the original server disconnects.

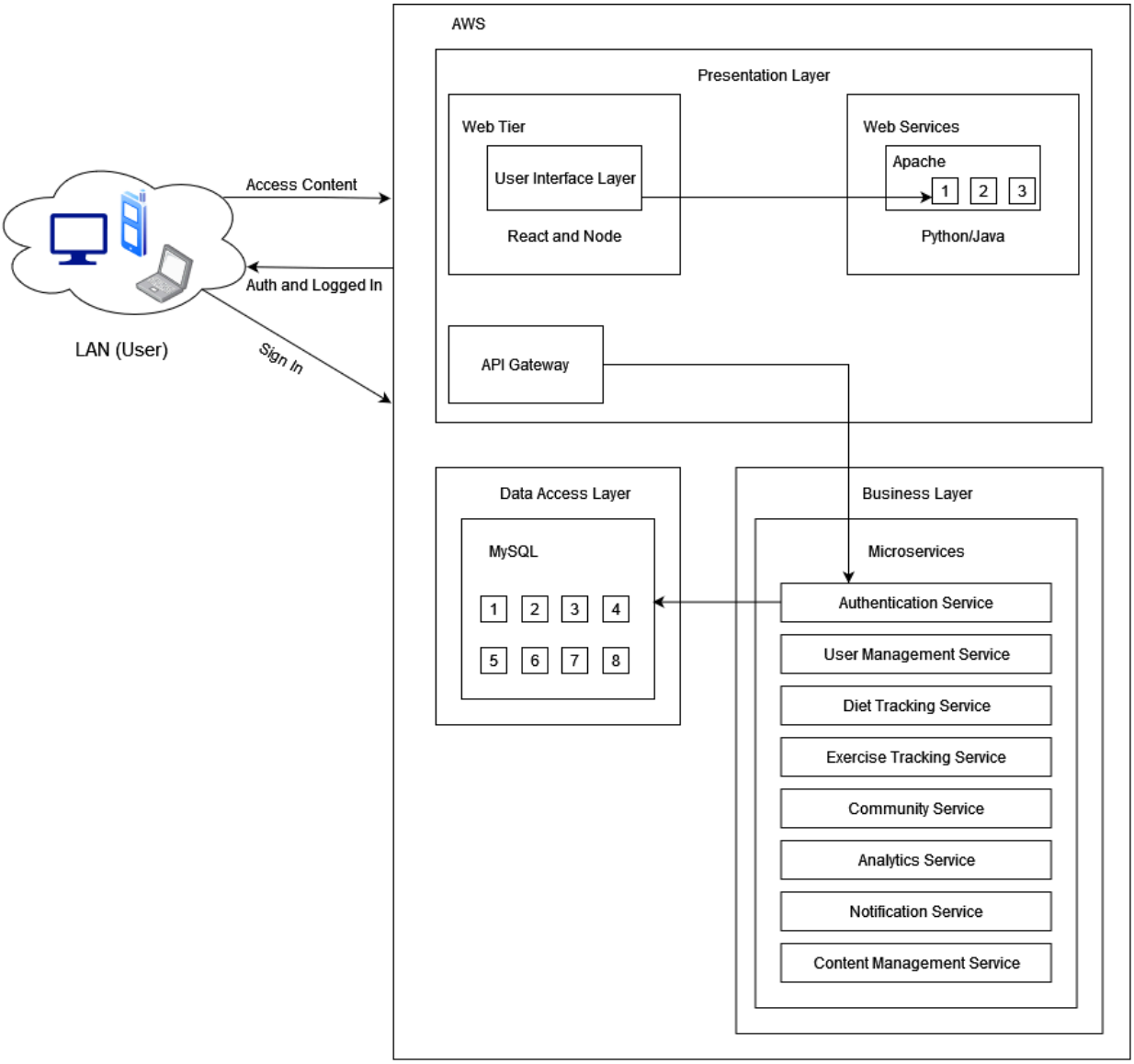
Authentication will be handled using an API Gateway making it so that data is encrypted while it is transmitted.


Activity
- activityID: String
- activityDate: DateTime
- activityType: String
- activityDuration: Float
- activityDistance: Float
- caloriesBurned: Float
- heartRate: Float
+ LogActivity()
+ getActivity()
+ updateActivity()
+ deleteActivity
Recipe
- recipeID: String
- recipeName: String
- description: String
- category: String
- preparationTime: Int
- cookingTime: Int
- totalTime: Int
- servings: Int
- ingredients: String
- nutritionInfo: String
+ createRecipe()
+ getRecipe()
+ updateRecipe()
+ deleteRecipe()
Analytics
- weight: Float
- dailyCalories: Float
- dailyProtein: Float
- dailyCarbohydrates: Float
- dailyFat: Float
- exerciseTime: Float
- stressLevel: Int
+ logData()
+ getData()
+ updateData()
+ deleteData()
HealthMetrics
- healthGoal: String
- healthline: String
- bmi: Float
- rhr: Float
- a1cReading: Float
- dietaryRestrictions: String
- dietPlans: String
- diabetesManagement: String
- ketogenic: Boolean
- vegan: Boolean
- Vegetarian: Boolean
- Paleo: Boolean
+ setHealthGoal()
+ getHealthMetrics()
+ updateHealthMetrics()
Nutrition
- mealID: String
- mealType: String
- foodItem: String
- caloriesConsumed: Float
- macronutrients: String
- micronutrients: String
- fiberContent: String
+ logMeal()
+ getMeal()
+ updateMeal()
+ deleteMeal()
User
- userID: String
- username: String
- email: String
- password: String
- dateOfBirth: Date
- gender: String
- height: Float
- weight: Float
+ register()
+ login()
+ updateProfile()
+ deleteProfile()
HealthRecommendations
- nutritionRecommendation: String
- mealRecommendation: String
- dailyRecommendation: String
- weeklyRecommendation: String
- monthlyRecommendation: String
- workoutRecommendation: String
+ generateRecommendations()
+ getRecommendations()
+ updateRecommendations()
UserManagementAPI
+ registerUser()
+ loginUser()
+ getUserProfile()
+ updateUserProfile()
+ deleteUserProfile()
+ authenticateUser()
+ authorizeUser()
HealthRecommendationsAPI
+ generateRecommendations()
+ getRecommendations()
+ updateRecommendations()
NutritionTrackingAPI
+ searchFoodItem()
+ logFoodItem()
+ getFoodLog()
+ updateFoodLog()
+ deleteFoodLog()
+ calculateNutrition()
+ getExternalFoodData()
WorkoutPlanningAPI
+ createWorkoutPlan()
+ getWorkoutPlan()
+ updateWorkoutPlan()
+ deleteWorkoutPlan()
+ getDeviceData()
CommunityEngagementAPI
+ createPost()
+ getPost()
+ updatePost()
+ deletePost()
+ createComment()
+ getComment()
+ updateComment()
+ deleteComment()
+ likePost()
+ manageUserGroups()
+ manageGroupChallenges()
ContentManagementAPI
+ createRecipe()
+ getRecipe()
+ updateRecipe()
+ deleteRecipe()
+ manageIngredients()
+ manageInstructions()
+ manageNutritionInfo()

Summary of Design Patterns:

Of the various UML design patterns, three were specifically utilized because of their importance in reliability and consistency. First is the usage of the singleton pattern. This was important for making sure that there would not be multiple instances needed to perform their actions which reduces the overall resources required. Second is the factory pattern which allows for subclasses to create objects on their own. This is important for allowing the subclasses to make new objects as needed without having to rely on changing code. Third is the observer pattern which makes it so that all states of an object stay updated. This allows the objects to always be in sync so there is more consistency when relaying data.

7. High-Level App Network and Deployment Design



8. Identify Actual Key Risks

8.1 Skills Risks:

- FitNutri uses Java and Spring Boot for the backend and JavaScript and React for the frontend. This requires team members to quickly learn and adopt. Insufficient expertise could lead to delays, errors, and challenges in implementing features effectively.

8.2 Schedule Risks:

- A significant schedule risk is the shorter development time frame compared to teams during Fall and Spring semesters. This compressed schedule increases pressure and the likelihood of encountering delays and feature goals not being met.

8.3 Technical Risks:

- Integrating various technologies poses a technical risk. FitNutri leverages an impressive technology stack, including AWS, MySQL, Linux/Bash, Apache, and more. Ensuring seamless integration and compatibility among these technologies could be challenging and might lead to unforeseen technical issues, causing delays in development.
- FitNutri may face performance risk if the backend is not adequately optimized to handle high traffic as the user base grows. Ensuring the app scales effectively to accommodate a large number of users without degradation in performance is critical.
- Ensuring the app runs smoothly involves continuous monitoring and maintenance. Any downtime or operational failure can lead to a loss of user trust and revenue.
- Relying on third-party services (e.g., AWS, APIs) introduces a risk if those services experience outages, change terms, or discontinue services, potentially disrupting FitNutri's functionality.
- If the app is not user-friendly, it could lead to low retention rates and negative reviews.

8.4 Teamwork Risks:

- Effective collaboration and communication are important for a software development team, especially one that is fully remote like ours. Miscommunication, coordination issues, failure to attend meetings, and the lack of in-person interaction could lead to delay-inducing misunderstandings and decreased team cohesion.

8.5 Legal/Content Risks:

- Ensuring compliance with data privacy regulations like GDPR and CCPA is essential. Failure to adequately protect user information or obtain necessary consent could result in legal consequences.
- The reliance on user-created content within FitNutri, specially on the topic of health and fitness, introduces a whole world of content risks. Ensuring the accuracy, reliability, and appropriateness of this content is vital to mitigate legal and reputational risks.
- Protecting user data from breaches and cyber-attacks is crucial. Any vulnerability can lead to data loss, which could damage the app's reputation and lead to legal repercussions.

8.6 Market Risks:

- There is a risk that the market demand for fitness and nutrition apps could change due to new trends, economic conditions, or changes in user preferences. This could affect the app's growth and profitability.

9. Project Management

To manage our M2 tasks, our team used Google spreadsheet and Notion as our primary task management tools. I (team lead) created our own channel called “FitNutri” to be able to keep track of our progress. Notion already gave me an outline of things I might need to keep everything organized. We have a task folder where we can add tasks, assign team members, add deadlines, and if it is a high or low priority. We are also able to update our status to say if we are “in progress” or “done”. This layout provides a clear overview of the workload and progress for each member and team lead. There is also a document area for important information that is needed when doing this milestone.

We think having questions when doing this milestone when we are not in a meeting is important. That’s why it is important to have our Discord channel for Team 03 to communicate in. This serves as our primary communication channel for discussions, questions, and collaboration. By having this Discord channel everyone is able to call and communicate.

Google Docs is also an essential tool for us because we write everything on a shared doc. We are able to collaborate and the team lead is able to write feedback. Having the doc ensures that everyone is on the same page regarding the details and expectations of the project. With docs, we are able to have all our work done on the same page. With all these tools we will continue utilizing Notion, Discord, and Google Docs for our task management and communication.

10. Detailed List of Contributions

NO.	Member	Contribution	Rating
1	Michelle Nguyen (Team Lead)	● Helped work on the mockup drawings. ● Created a notion to keep track of tasks. ● Looked over the document for corrections and grammar, ● Planned meetings. ● Helped with the EDR. ● Wrote out the Project Management.	
2	Mitchell Caine	● The first 4 use cases for the UI/mockup. ● Helped Ali Hadwan with the storyboard and had a 1 on 1 meeting to discuss the design of the application/website. ● Worked on the flow chart diagram for our system design. ● The system design. ● Wrote out the summary components. ● Used mySQL to organize our functions on our application. ● Summary of the design pattern.	10
3	Shreejana Bartaula	● n/a	0
4	Eduardo Enrique Muñoz Alvarez	● Planned out the EDR diagram with Ali Hadwan. ● Worked on the high-level database architecture. ● Worked on mySQL to create the eer. ● Defined our DBMS.	10
5	Nilofar Ali	● Worked on data definitions. ● Worked on prioritized functional requirements	8
6	Uzair Hamed Mohammed	● programmed the backend of our prototype. ● Wrote out the high-level APIs and Main Algorithms. ● Collaborated with John to create the high-level app network and deployment design. ● Identify actual risks with John.	10
7	John Collins	● Worked on data definitions. ● Collaborated with Uzair to create the high-level app network and deployment design. ● Worked and collaborated with Uzair on the actual risks.	8

9	Ali Hadwan	● Planned out the application/website design. ● Drew out the use cases. ● UI Mockups and storyboards. ● Drew all, planned, and collaborated with the team if everyone agreed with everything. ● Worked on the erd on draw.io. ● Programmed the front end of our prototype and collaborated with Uzair.	10
10	Ali A	● n/a	0