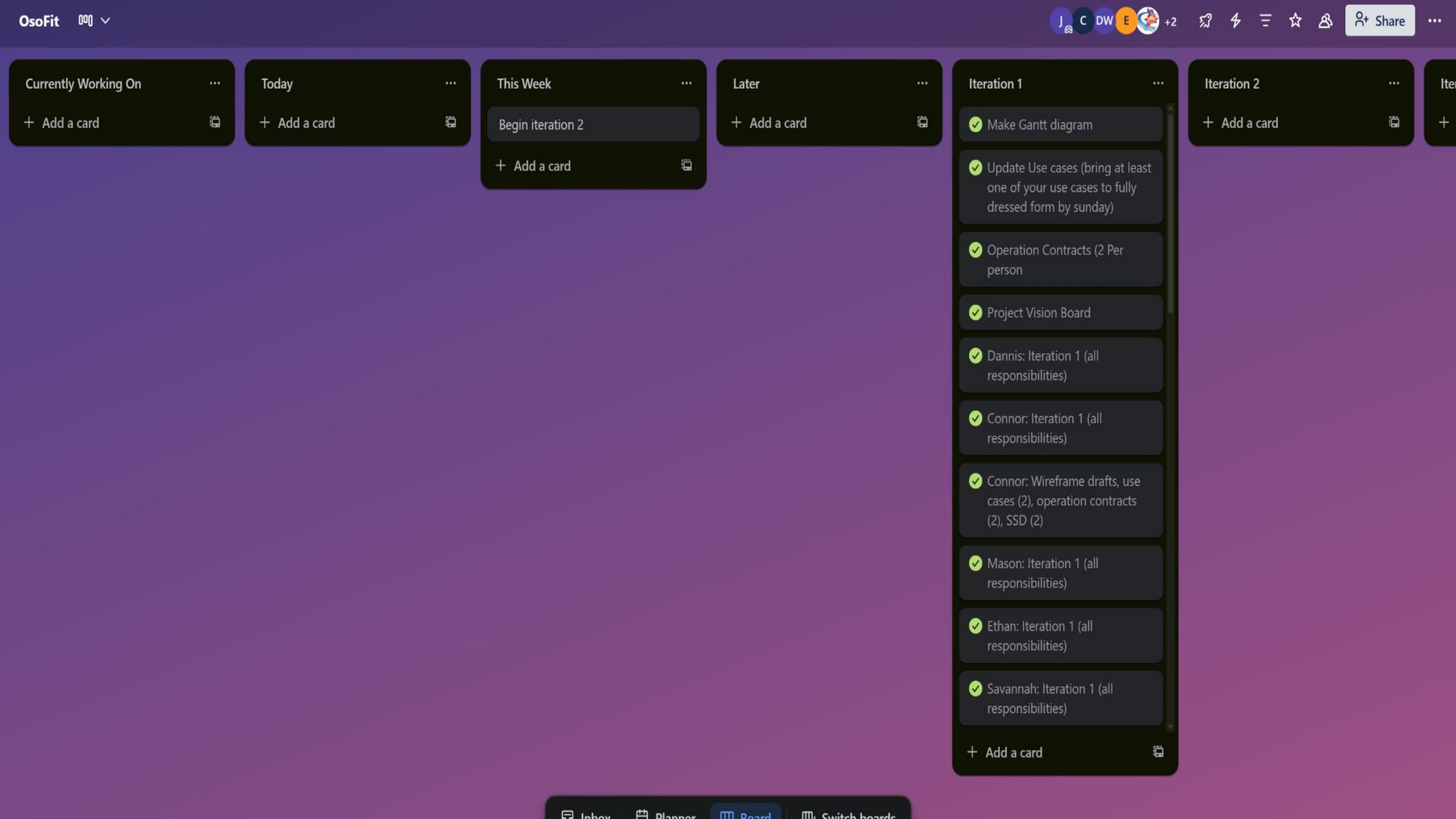
OsoFit

OsoFit- Unified Health & Fitness App

- What are we making?
  - A mobile app for fitness tracking, goal setting and trainer-led plans
- Who is the target audience?
  - Health-focused individuals, trainers, admins
- Why better than competitors?
  - Combines tracking, reporting, and social features into one unified platform

Problem & Solution

- Problem Statement: Inconsistent health tracking & poor goal monitoring
- Impact: Users and trainers lack unified, guided tools
- Solution: OsoFit integrates tracking, structured plans, reporting, and community features
- Competitors: Fitbit, strava, whoop
- Advantage: Single platform for all goals



Currently Working On

+ Add a card

Today

+ Add a card

This Week

Begin iteration 2

+ Add a card

Later

+ Add a card

Iteration 1

- ✓ Make Gantt diagram
- ✓ Update Use cases (bring at least one of your use cases to fully dressed form by sunday)
- ✓ Operation Contracts (2 Per person
- ✓ Project Vision Board
- ✓ Dannis: Iteration 1 (all responsibilities)
- ✓ Connor: Iteration 1 (all responsibilities)
- ✓ Connor: Wireframe drafts, use cases (2), operation contracts (2), SSD (2)
- ✓ Mason: Iteration 1 (all responsibilities)
- ✓ Ethan: Iteration 1 (all responsibilities)
- ✓ Savannah: Iteration 1 (all responsibilities)

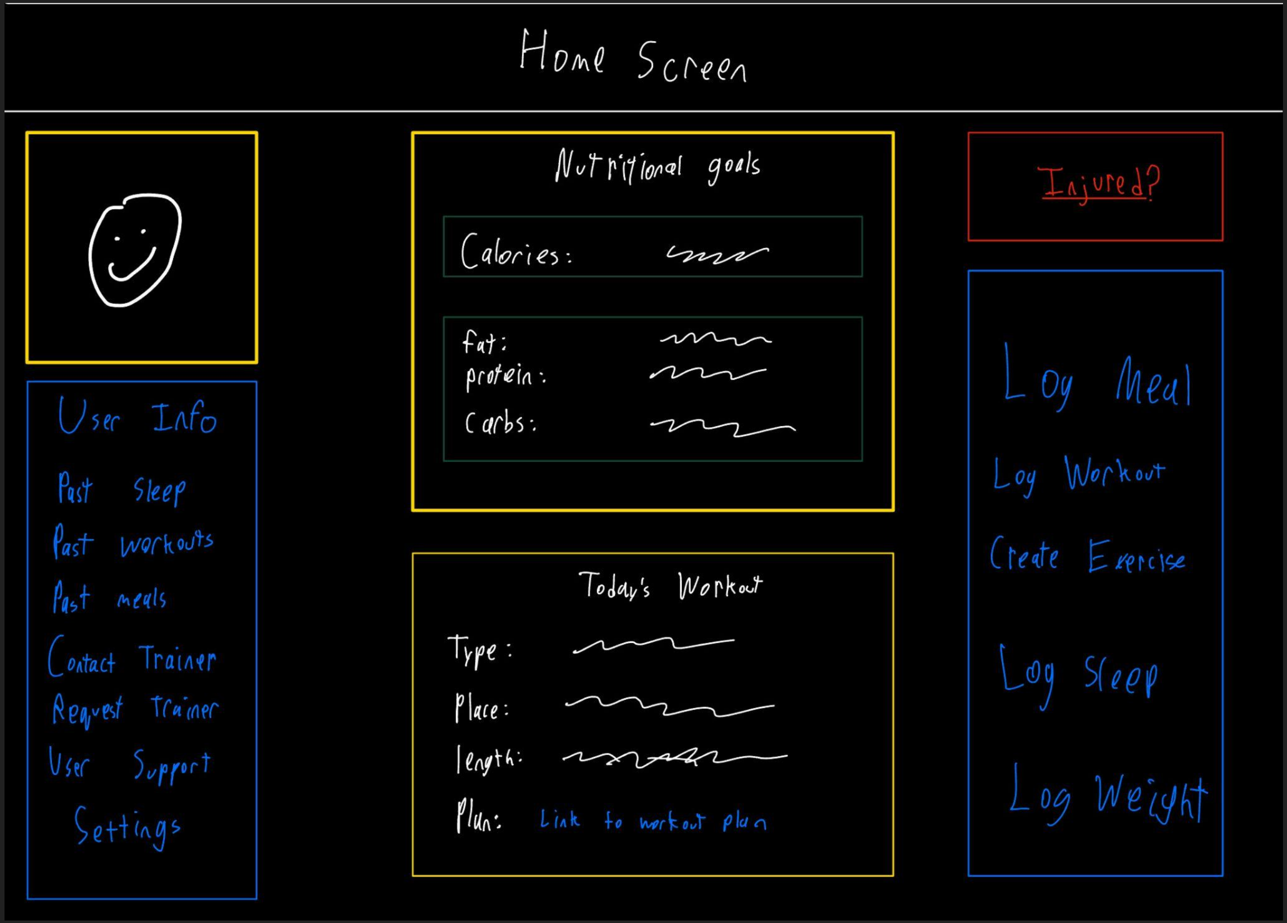
+ Add a card

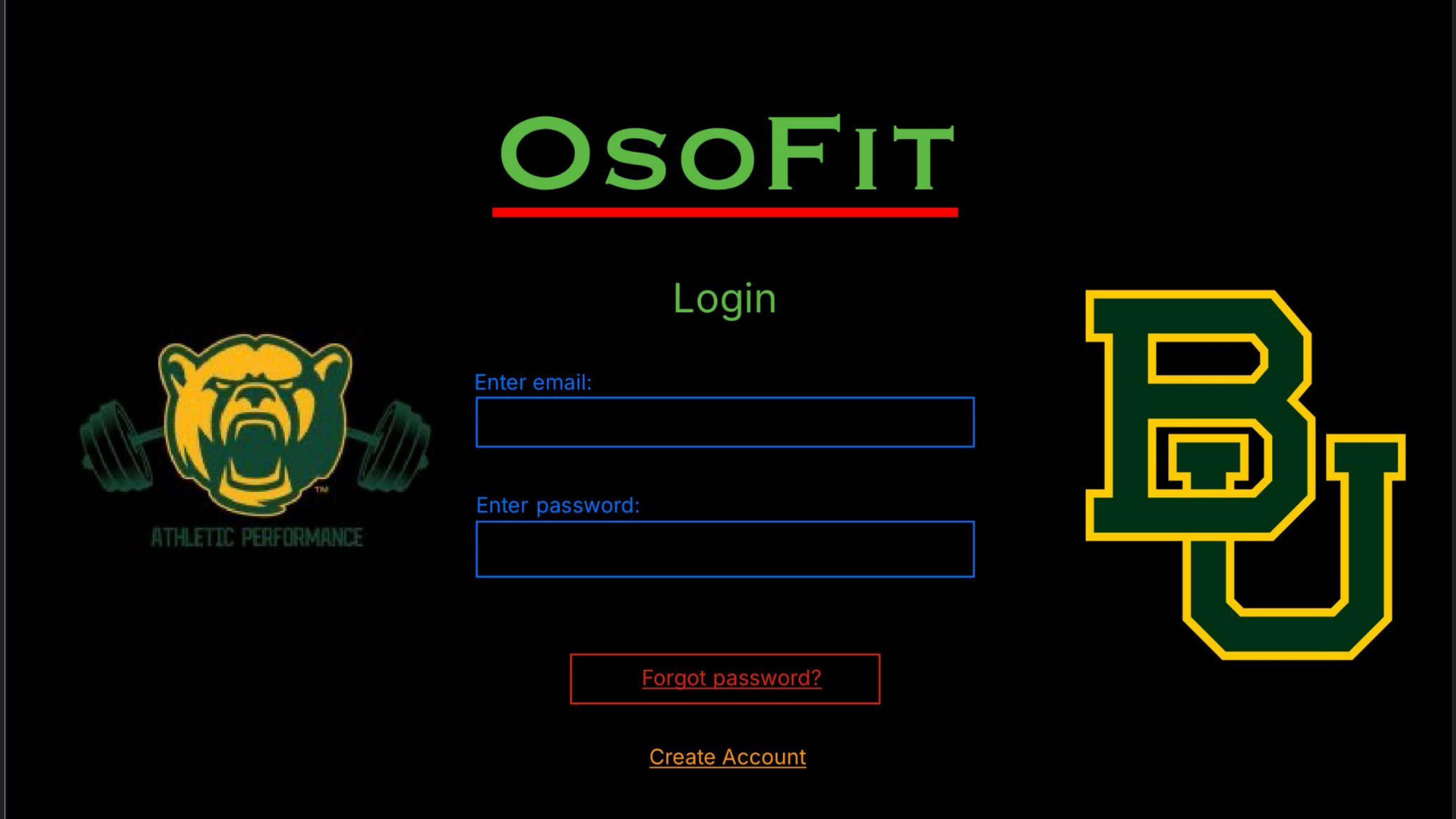
Iteration 2

+ Add a card

Ite

Wireframes


Home Screen
Nutritional goals
Calories:
fat:
protein:
carbs:
Today's Workout
Type:
Place:
length:
Plan: Link to workout plan
User Info
Past Sleep
Past workouts
Past meals
Contact Trainer
Request Trainer
User Support
Settings
Injured?
Log Meal
Log Workout
Create Exercise
Log Sleep
Log Weight



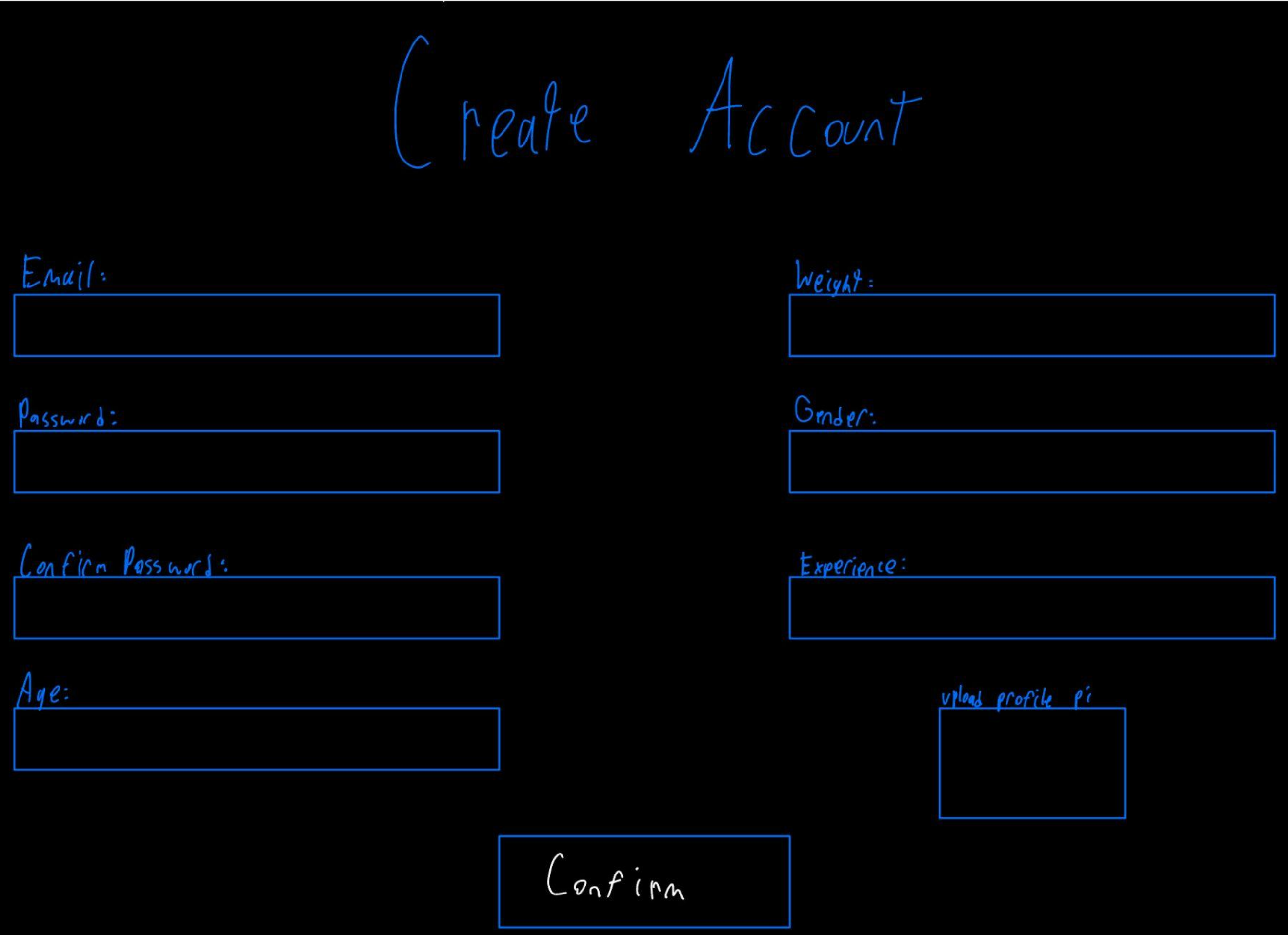
OsoFit

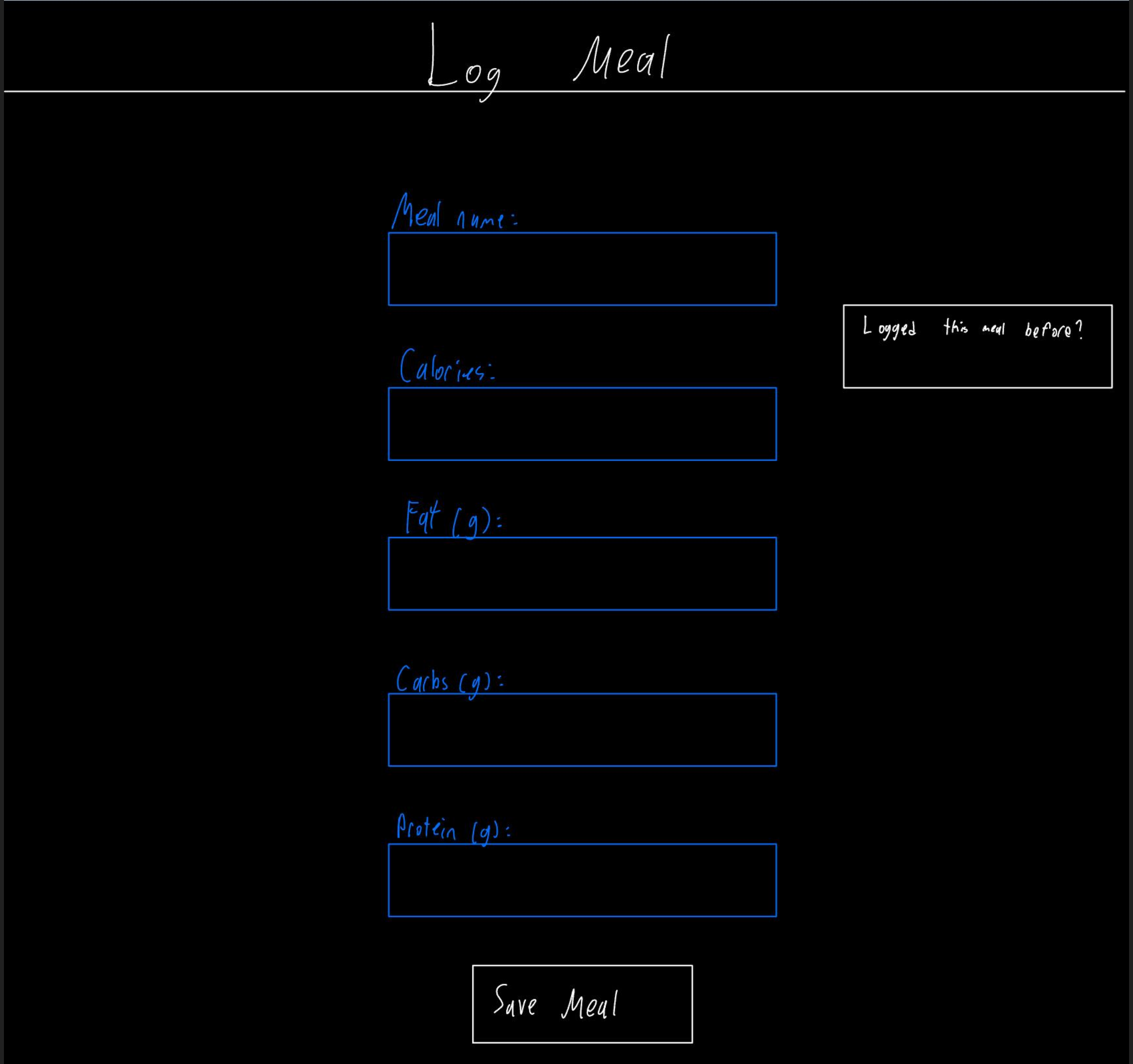
Login

Enter email:

Enter password:

Account Creation





Log Meal

Meal name:

Calories:

Fat (g):

Carbs (g):

Protein (g):

Save Meal

Logged this meal before?

Domain Model

Use Case 1 (Example: Log Workout)

Title: Use Case: *Log Workout Session*

Actors: User, System.

Precondition: User authenticated.

Postcondition: Workout saved, dashboard updated.

Main Flow (summarized):

1. User opens workout page.
2. Enters type, duration, calories.
3. Clicks save.
4. System validates, stores, updates dashboard.

Alternates: Missing field, invalid entry, or system unavailable.

SSD for Use Case 1

- **User** → **System**: openWorkoutPage()
- **System** → **User**: displayWorkoutForm()
- **User** → **System**: createWorkout(type, duration, calories, notes)
- **System**: validate(), saveWorkout(), updateDashboard()

System responds with updated dashboard reflecting workout progress.

Contract for Use Case 1

- **Operation**: createWorkout(userId, type, durationMin, calories, notes, dateTime)
- **Preconditions**: user authenticated, valid inputs.
- **Postconditions**:
  - Workout object created.
  - Dashboard aggregates recalculated.
  - Confirmation message returned.
- **Exceptions**: ValidationError, DuplicateWarning, ConnectionError.

Use Case 2 (Example: Register for class)

- **Title**: Use Case: *Register for a Class*
- **Actors**: User, System.
- **Precondition**: User logged in.
- **Postcondition**: User enrolled in class or waitlisted.
- **Main Flow**:
  1. User browses classes.
  2. Checks capacity/prereqs.
  3. Registers.
  4. System confirms enrollment.
- **Alternates**: Class full → waitlist, unqualified → error message.

SSD for Use Case 2

Diagram:

- **User** → **System**: viewClasses()
- **System** → **User**: displayClassList()
- **User** → **System**: registerClass(classId)
- **System**: validateCapacity(), validatePrereqs(), confirmRegistration()

Caption: *System confirms registration or shows error/waitlist notification.*

Contract for Use Case 2

Operation: registerClass(userId, classId)

Preconditions: user logged in, class exists.

Postconditions:

- Enrollment created OR waitlist updated.
- User’s dashboard updated with upcoming session.

Exceptions:

- ClassFull, NotQualified, ClassPendingReview.

Next steps (Iteration 2)

Polish wireframes into mid-fi prototypes.

Refine SSDs with more detail.

Expand reporting features and add missing requirements clarifications.

Prepare refined demo for presentation.