FITQUEST

PROJECT PROPOSAL

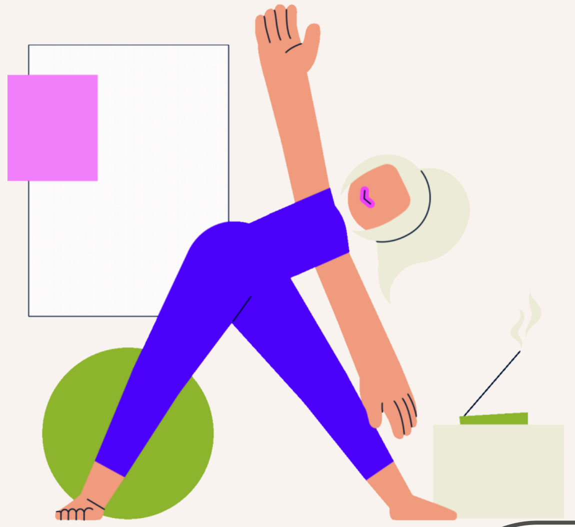
FITQUEST

PRESENTED BY : YOL-EMMA VEILLARD

GAMIFYING EXERCISE

CORE CONCEPT

A fitness app where users log their exercises and earn points, badges, or levels based on their physical activity. They can compete with friends or others nearby, view leaderboards, and take part in challenges. It encourages healthy competition while tracking their fitness journey.



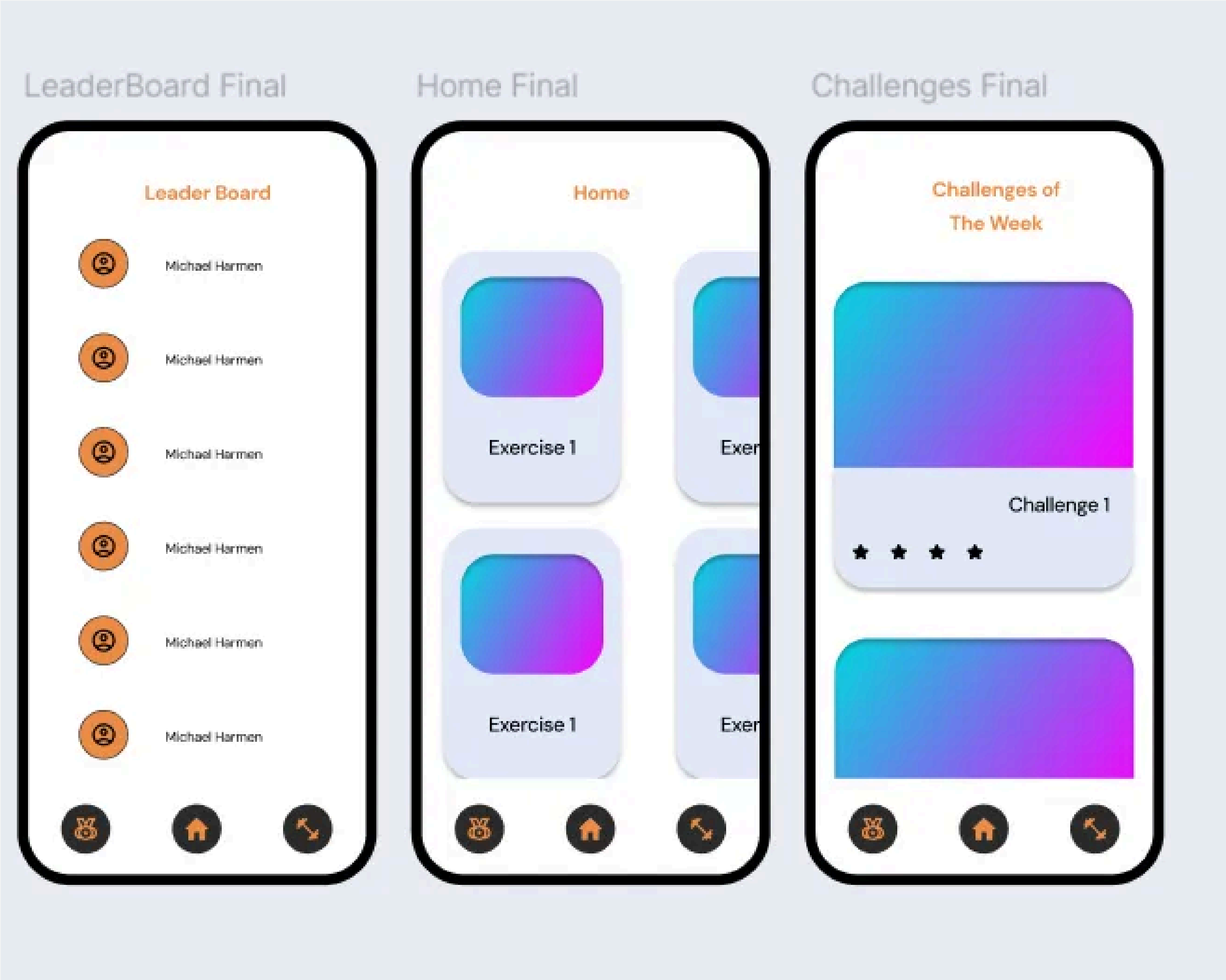
WHO IS THIS FOR?

- Fitness Enthusiasts:
  - Those who thrive on competition and goal-setting to push their limits.
  - They seek leaderboards, challenges, and progress tracking.
- Casual Users:
  - People looking to stay active in a low-pressure, social environment.
  - They benefit from fun challenges and friendly battles with friends.
- Solo Exercisers Seeking Community:
  - Individuals who work out alone but want to feel connected to a wider fitness community.
  - Features like virtual leaderboards, shared challenges, and activity feeds create a sense of belonging.

FITQUEST

KEY FEATURES

1. Home Screen:
    - After account creation, users land on the home screen, featuring a curated list of recommended exercises.
2. Challenges Tab:
    - Users can explore various fitness challenges designed to make workouts engaging.
    - Battle Mode:
        - Users can invite friends to compete in specific challenges.
        - The app tracks progress and performance metrics (e.g., speed, form, or consistency) to determine who "won" the challenge.
3. Leaderboard:
    - Tracks and displays top performers globally or among friends.
    - Users can filter leaderboards by specific metrics.



PROJECT PROPOSAL
PRESENTATION

FITQUEST

TECH OVERVIEW

1. Tech Stack
   - Frontend: Kotlin with Jetpack Compose for UI.
   - Backend: Firebase (Firestore, Authentication, Cloud Functions).
   - APIs: Health Connect.
   - Offline Support: Room for local data storage.
   - Background Tasks: WorkManager for periodic data syncing and reminders.
2. External Libraries/Tools
   - Retrofit: For networking.
   - Firebase Crashlytics: For real-time error logging.
   - Coil: For image loading (e.g., profile pictures).
3. Data Flow
   - Fitness data is fetched from the Health Connect API and synced to Firebase Firestore.
   - Room handles offline data storage and caching.
   - Leaderboards and challenges are dynamically updated in real-time.

PROJECT PROPOSAL

PRESENTATION