CSS 497 Design Specification

**Created by:** Sam Baranov

**Quarter:** Summer Quarter, 2025

Part 1▾Basic Info

Building a Unified Fitness Tracker System for Automated Progression Tracking and Planning

The current market of fitness tracking apps typically fails to provide integrated tools that effectively project and adapt serious fitness programming while incorporating relevant, in-depth metrics for advanced lifting regiments. Most available apps focus on a singular dimension, such as past data visualization, individual session planning, or gamification, but rarely combine all three venues in a singular package. This shortfall is particularly evident in the widespread preference among intermediate and advanced lifters for paper or spreadsheet-based tracking, over singular phone-based apps. This project addresses that gap by building an app that simultaneously:

* Dynamically adjusts training cycles based on the combination of real performance data and auxiliary lifestyle information
* Supports fully customizable tracking metrics for individual exercises, as well as metric visualization and analysis
* Leverages gamification to promote long term engagement and consistent data input to fuel fitness suggestions and encourage consistent record keeping

The core goals for this project are:

* Build an extensible architecture that can be modularly expanded upon, supporting both simple and advanced progression models such as linear, undulating, hybrid periodization.
* Develop an adaptive planning framework that generates workout prescriptions based on real-time performance metrics, historical trends, individualized baselines, and user defined training parameters.
* Create a flexible metric system that allows users to define, customize, and track any combination of variables for individually relevant monitoring.
* Implement gamification systems such as streaks, unlockable features, and progress visualizations that reinforce consistent data input and retention of the users.
* Design a minimal friction user interface optimized for quick interaction, but robust data entry capabilities.
* Iteratively develop and test each major feature set, ensuring practical stakeholder feedback is integrated over the course of development.

The intended end result is the publishing of this app to the Google Play Store. By the end of the first quarter, I intend to reach the MVP stage, while by the end of the second quarter I am planning on getting to the third, feature complete version. Demonstration will involve a live walkthrough of the app’s functionality on a physical Android device or emulator, and validation will include automated testing, in-field usability feedback from stakeholders at different levels of completion, and a comparison of planning outputs against similar real world use cases.

The project will be split into three distinct iterations to give concrete milestones. First iteration is an MVP, followed by two major iterations in the second quarter. This will also give natural breakpoints for stakeholder and expert meetings and refinements. More concrete feature breakdown is observed in the development planning tables for first and second quarters.

This project will contribute to my career by providing hands on experience managing a full-stack app development lifecycle from planning to deployment for a non-software related subject. It will also involve algorithmic design around dynamic programming logic, which directly addresses a major gap in my current experience. This will fulfill all of the general core competencies for analysis problem solving as well as:

* Technical competencies
  + Data Analysis and Statistics
  + Quality Assurance
  + Software Architecture
  + Technical Writing
  + Discrete Mathematics
* Advanced Competencies
  + Algorithm Design and Development
  + Contemporary Database and Design Interface Tools
  + Database Design
  + Interface Design Principles
  + OOP
  + Usability Testing

In addition to technical growth, releasing a tool on a public marketplace will support my overall long-term goal of launching independent software products. The development practices required for this capstone project contribute to my understanding of the expectations for both freelance and industry environments, thereby strengthening my positioning for future roles. Even if I do not intend to necessarily stay in this specific software ecosystem.

Part 2▾Resouces

There are two experts that will be contacted for reviews during the project development:

1. Acquaintance who is a senior engineer at Google, with experience in Android  
   development, although mostly with Java. This is the person I will go to for code reviews after completion of MVP and future iterations of the project, as well as overall architectural questions. May have limited availability, will be scheduled upon completion of individual iterations, for three total meetings online: after MVP, Iteration 2, and Iteration 3 completion.
2. Junior UI programmer at a sales firm. This is the expert I will content for my prototype review, as well as overall UI design questions once UI is solidified for each iteration. They have experience with creating an extensive number of UI focused projects in different fields, while also having some experience with fitness training. They are easy to get a hold of, so they will be contacted in person when possible. They will be contacted upon the completion of Figma prototype, MVP, and final iteration.

There are three stakeholders that will be contacted for reviews during the project development:

1. Gym coach. Will meet to review progression algorithms usefulness and accuracy,  
   and metric tracking relevance via short in person interviews after MVP and final iteration completion.
2. Intermediate technical lifter friend. Will give user centered feedback on usability,  
   data entry experience, perceived value, and other subjective metrics, at all stages of the project due to accessibility. Will review the initial project idea, Figma prototype, as well as MVP and final iterations.

The overall technical stack for this project is as follows:

* Visual Studio Code with Flutter plug-in as IDE/code editor
* Flutter and Dart as underlying framework and programming language
* SQLite 3 with Drift for Flutter as database solution
* Figma for high-fidelity mockups and UI prototyping
* Git as industry standard version control.

The necessary hardware is largely limited to generic resources needed for Android development, as well as access to real-world testing context:

* Physical testing device running Android
* PC as the primary development machine with consistent development environment
* Secondary device for testing under different environment and documentation writing as I try to keep Office on only one device

Part 3▾ Detailed Specifications

The application will follow a Clean Architecture approach structured around the MVVM presentation pattern for the front end. Diagram 1 showcases a possible architecture for the MVP, but will certainly be expanded as project evolves. The architecture will be separated into Presentation, Domain, Data, and Infrastructure layers, in order to enforce separation of concerns and support modular development over multiple iterations.

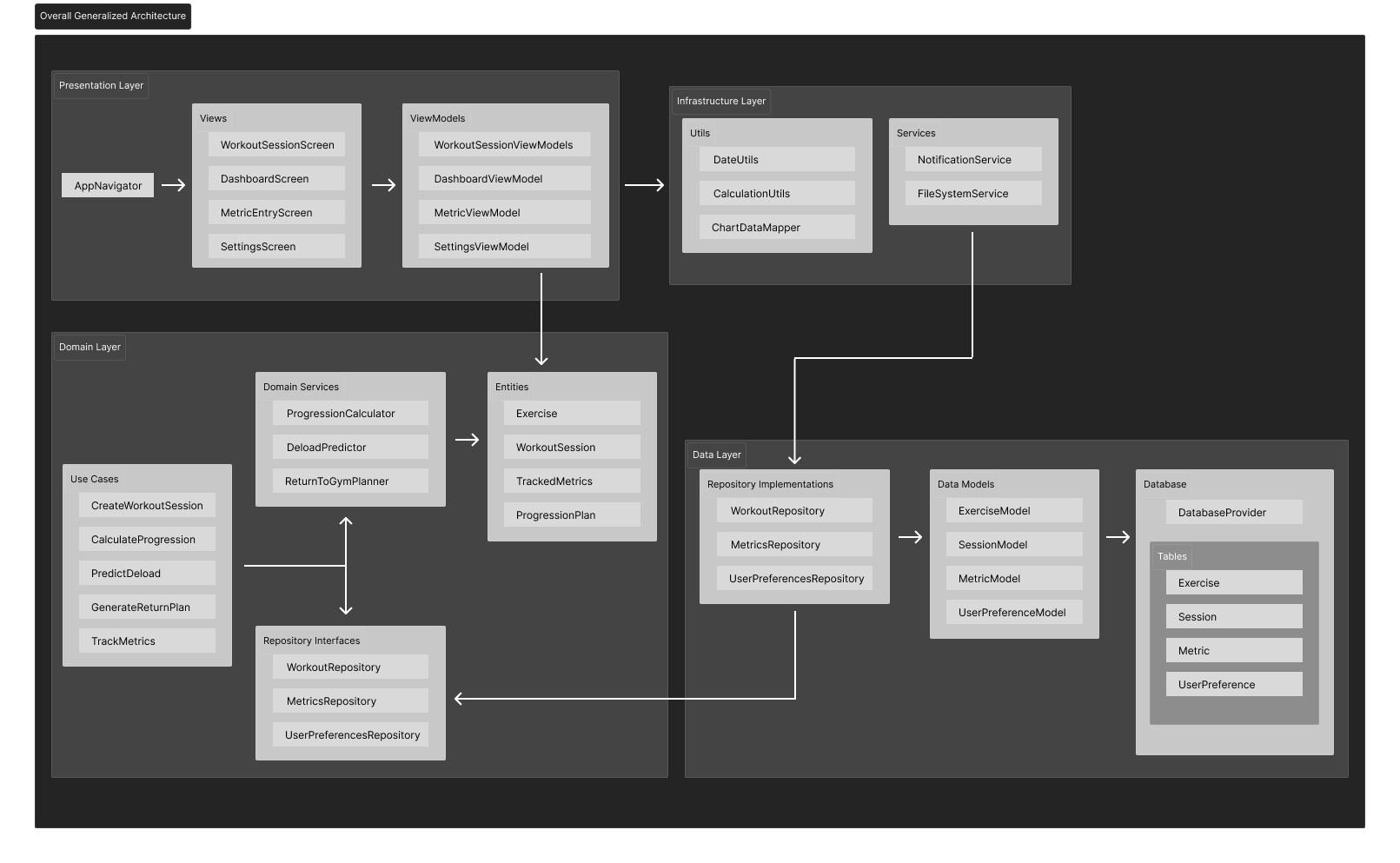


Diagram 1: Fitness App Diagram for Clean Architecture Approach

Additionally, Diagram 2 includes a preliminary organization of the database schema for the back-end of this project.

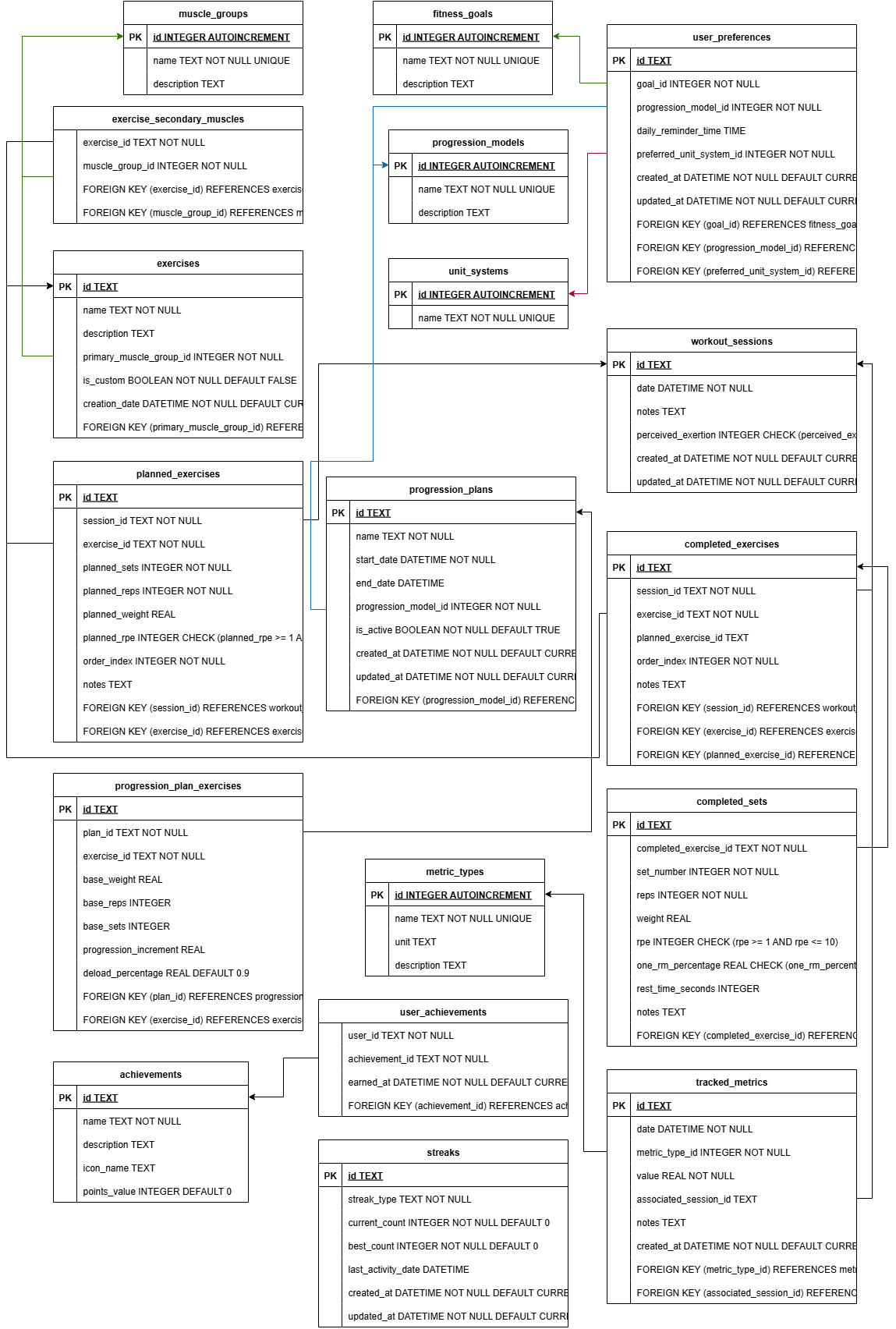


Diagram 2: Preliminary Database Schema

These are the primary domains along which this application will be tested.

Unit Testing

* Algorithms
* Data transformations
* Repository CRUD operations

Integration Testing

* ViewModel interactions simulating user flows
* Planning workflows from metric entry to plan updates

UI Testing

* Core screens rendering verification
* Edge case handling for empty states and invalid inputs

Manual Testing

* Real device runs for animations, interactions, offline data
* Testing across devices with different screen sizes and performance levels.

Edge Cases

* Logging with invalid or missing data
* Extreme numbers for sets, reps, or weights
* Corrupted user preferences
* Interrupted metric entry from crashes or forced closes.
* Return-to-gym plans after long inactivity

The following are the metrics which will be used to assess the success of the project.

* Technical Metrics
  + Consistency of progression and planning calculations across user data and statistics
  + Stability of data storage with no crashes or data corruption
  + Performance benchmarks with large user data volumes
* User Metrics
  + Consistent daily or weekly user engagement measured through streaks and statistics completion rate
  + Stakeholder feedback through built-in surveys
  + Goal progression rates of user making improvements upon chosen objectives

Part 4 ▾ Schedule

First Quarter

|  |  |  |  |
| --- | --- | --- | --- |
| **Wk** | **ACTIVITY** | **RESULT** | **EST TIME** |
| 2 | * Complete Week 2 status * Finalize project selection * Draft design specification * Begin low fidelity Figma prototype * Creation of initial architecture diagram | Project scope, tech stack, and early design direction are finalized. Preliminary Figma and architecture work is initiated. | 21.5 hrs |
| 3 | * Complete Week 3 status * Present Showcase 1 * Refactor architecture diagram * Begin transition to high fidelity Figma prototype | Overall screen flow is outlined and general architecture is finalized. | 21 hrs |
| 4 | * Complete Week 4 status * Finalize design specifications * Draft contract * Finalize high fidelity prototype * Prototype stakeholder meeting | Design specification is finalized, high fidelity prototype is implemented and discussed with stakeholder. | 24 hrs |
| 5 | * Complete Week 5 status * Implement finalized schema in Drift * Set up entity models and database * Build repository interface in Flutter * Expert meeting. | Database infrastructure and core backend repositories are established, and final architecture design is evaluated with an expert. | 20 hrs |
| 6 | * Complete Week 6 status * Implement core UI flows * Integrate basics of UI layer over domain layer | Core UI flows are implemented and wired to the backend through the domain layer. Basic features begin implementation. | 25 hrs |
| 7 | * Complete Week 7 status * Implement basic undulating progression algorithm in domain layer * Implement basic metric visualization | Basic adaptive planning algorithm is implemented, and implementation of visualization for data is started. | 25 hrs |
| 8 | * Complete Week 8 status * Implement basic or stub return-to-gym planning feature * Begin manual QA | Initial round of QA is started and more robust planning algorithm use case is implemented on top of the general session planning one. | 20 hrs |
| 9 | * Complete Week 9 status * Full testing over MVP features * Stakeholder and expert meeting to review MVP state * Create action list for next iteration | MVP is complete, tested, review with stakeholders, and an action list is created to start the next quarter of work. | 20 hrs |

Second Quarter

|  |  |  |  |
| --- | --- | --- | --- |
| **Wk** | **ACTIVITY** | **RESULT** | **EST TIME** |
| 1 | * Complete Week 1 status * Review feedback and action list from end of Q1 * Extend the metric system for custom metrics | Regained grasp on the project state, iteration 2 backlog is defined, expanded metric architecture is started. | 15 hrs |
| 2 | * Complete Week 2 status * Implement progression plan editing with training phases modality * Connect planning logic to revised metrics | User-adjusted planning system is implemented, ability to separate training into cycles is implemented. | 20 hrs |
| 3 | * Complete Week 3 status * Design and deploy recovery and fatigue modeling system in stub version * Add deload detection with trigger logic based on session patterns | Fatigue modeling and dependent deload detection are implemented in at least the stub version. | 25 hrs |
| 4 | * Complete Week 4 status * Implement streak, calendar, and reminder systems from common Flutter libraries. * Create basic visualization for generalized progression for gamification system. * Continue refinements to fatigue and deload systems. | User engagement system basics are implemented through common flutter functionalities, dashboard stub is designed and implemented, fatigue and deload systems are elaborated upon. | 15 hrs |
| 5 | * Complete Week 5 status * Expand analytics dashboard system * Start implementing muscle balance tracker based on session history and lift metrics * Finish deload and fatigue system implementation | Fatigue and deload systems are fully implemented, gamification is further expanded upon, and muscle balance system is implemented in basic sense. | 25 hrs |
| 6 | * Complete Week 6 status * Complete iteration 2 * Perform testing for iteration 2 implementation * Review UX with stakeholders. * Review system with UI expert if possible. | Iteration 2 features are finalized and feedback is received. | 20 hrs. |
| 7 | * Complete Week 7 status * Begin iteration 3 work by adding aesthetic improvements * If time permits, add quest system * Review overall system architecture | Common UI polish features and customizability are implemented using Flutter theme patterns. | 20 hrs |
| 8 | * Complete Week 8 status * Run full edge-case testing across all core features. * Run cross device testing on different android phones for feature and UI consistency. | Robust testing across all levels of architecture is conducted. | 15 hrs |
| 9 | * Complete Week 9 status * Freeze feature set and work on polish of whatever is left * Final meetings with stakeholders and experts | Finalizing the iteration 3 that will be used for the final presentation. | 25 hrs |
| 10 | * Complete Week 10 status * Continue polishing of finalized features. * Produce capstone presentation. * Publish to Google Play Store | Final version of the app for the presentation is created, and used as basis for the final presentation. Ideally the app is published to the Google Play Store. | 25 hrs |
| Finals week | * Conduct postmortem on the project * Work on presentation rehearsing * Present the capstone | Project reflection for personal portfolio is complete, capstone presentation is complete. | 15 hrs |

Part 5 ▾ Contingency Plans

These are the primary risks I anticipate for this project.

* Planning algorithm proves more complex than expected
  + I can pivot into basic linear progression model first; isolate complex calculations for later iterations of the project, while building out the underlying infrastructure first.
* If illness or personal emergencies occur
  + Front-load essential features early in the development schedule when possible and allocate buffer weeks at end of quarter for catch-up.