

Graduate Team 1's Semester Project Final Presentation

**CS 5320: Software Design
Semester: Fall
Aaron Elofson**

**Permission is granted for
publication on the UCCS website.**

Professor: Dr. Armin Moin

Project Idea

- **Service to pull in step and activity data from multiple services (Garmin, Google Fit, Coros)**
- **Normalize the Data**
- **Allow user to prioritize data based on source, time and activity**
- **Display combined activity data in app or widget**

Customer / End User

- **Athletes / Runners who have different devices or are part of different ecosystems.**
- **Those who need more control over their fitness data.**

What makes it Unique

- **Related Project - Health Sync**
 - Only transfers activities between services
 - Doesn't allow user to set preferences
 - No UI to display activities
 - Not open source
- **Ecosystem Health Apps (Apple, Google)**
 - Only for one brand of products
 - Doesn't support high end sports gear

Example APIs to be included

Google Health

- Requires android device to sync

Coros

- Requires development updates

Garmin

- Requires development updates

Apple (Health Kit)

- Requires apple device to sync data

Course Connection

- **Application of ADD 3.0**
- **Architectural Style and Design Patterns**
- **Software System Modeling and Views**
- **Project Development**

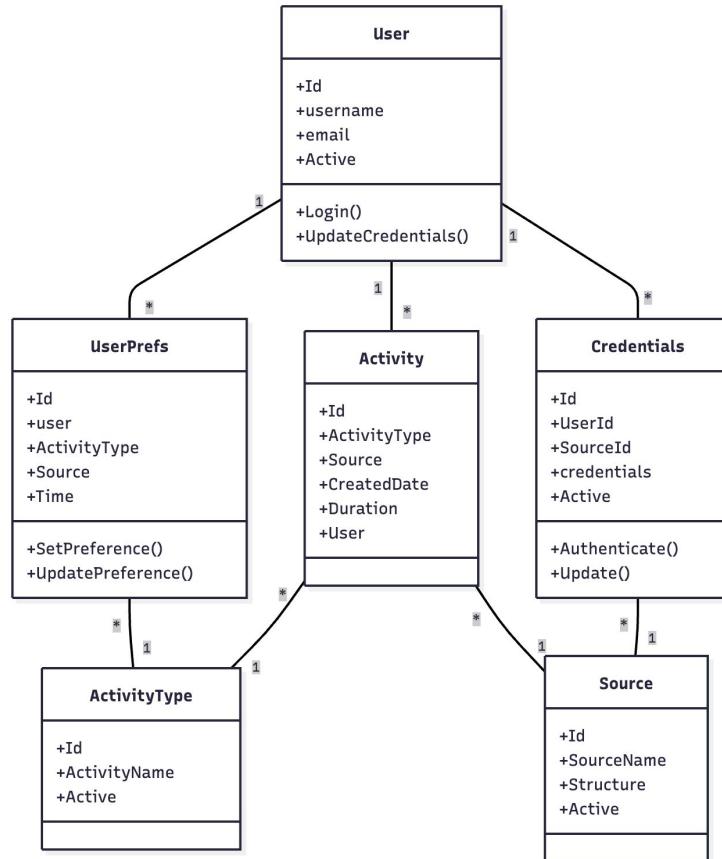
Project Path

- **Planning Path**
 - 2 Main reasons
 - *Complex back end that would take longer than semester months to develop*
 - *API Connection requests could take longer too long to return and require too much upkeep*
 - Diagrams and modeling will provide good plan for starting development

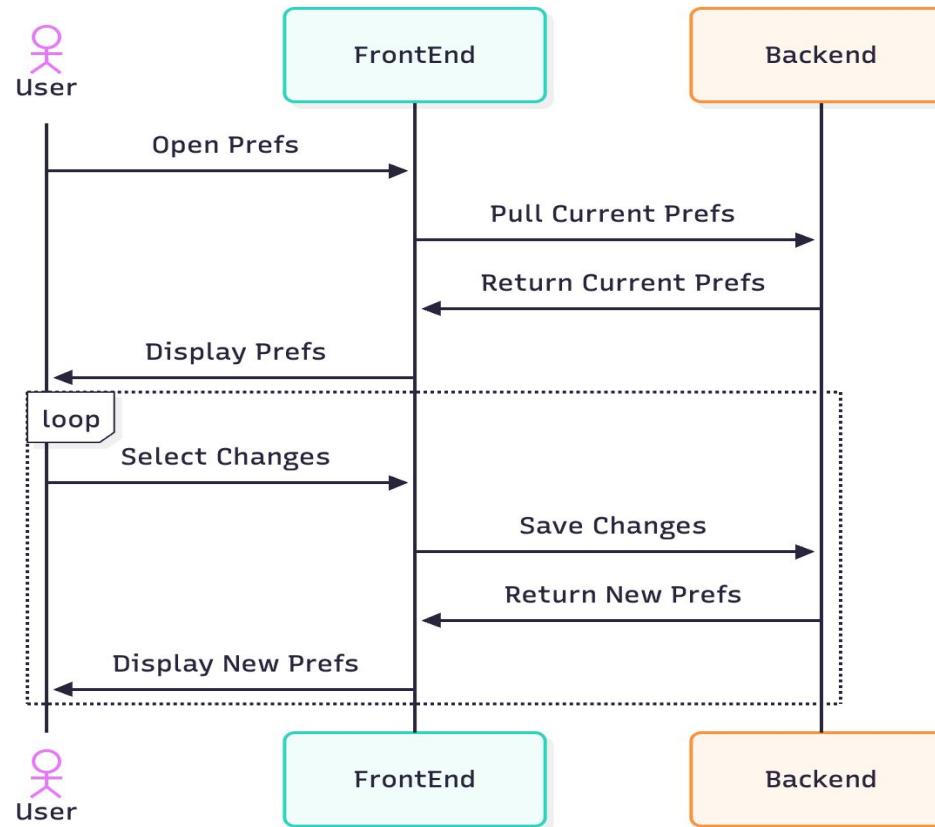
Design & Architecture

- **Patterns**
 - Thread Pool
- **Architecture**
 - Pipe and filter
 - Repository

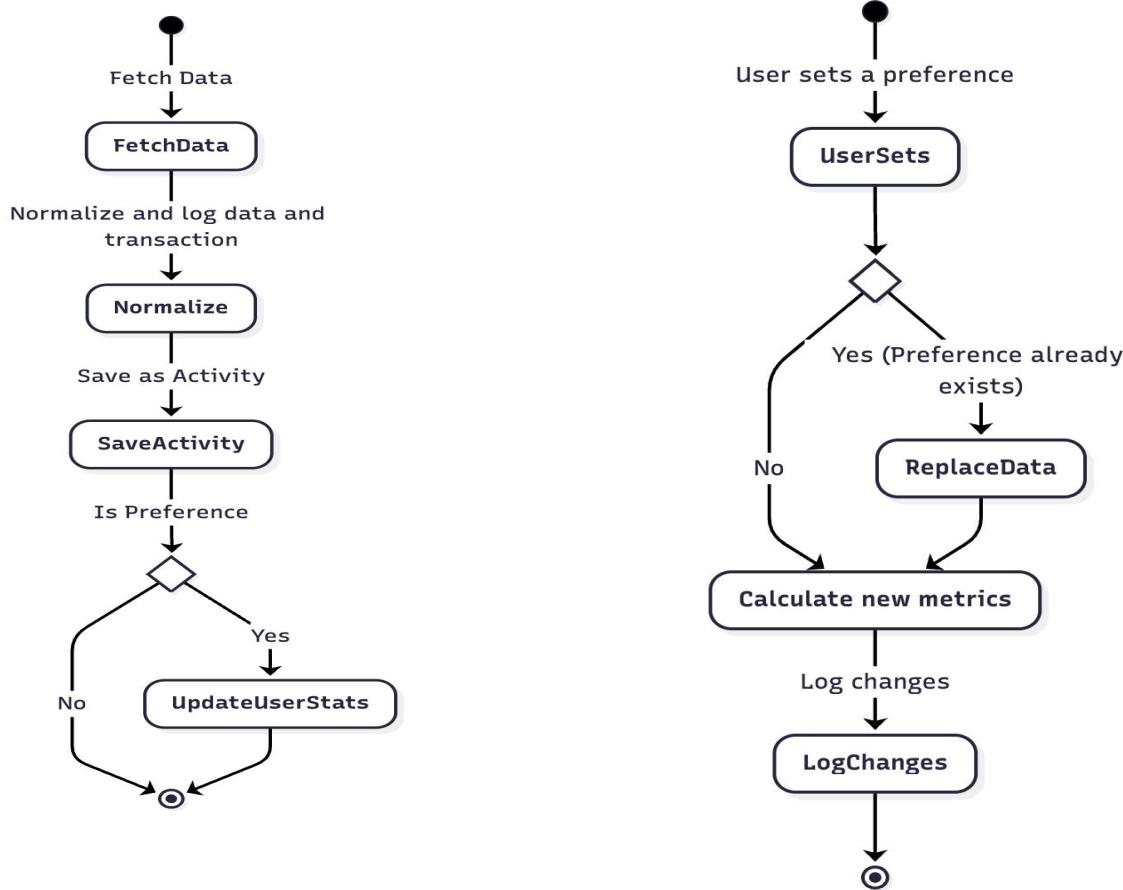
Class Diagram (Structure)



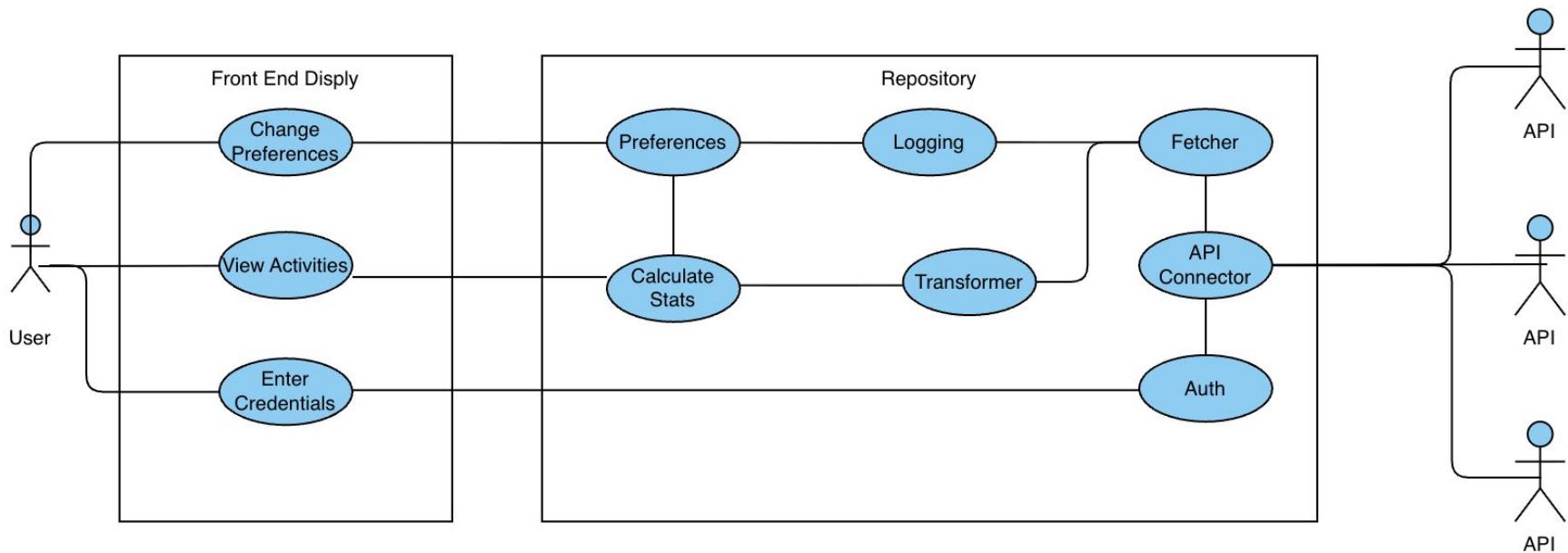
Sequence Diagram (Interaction)



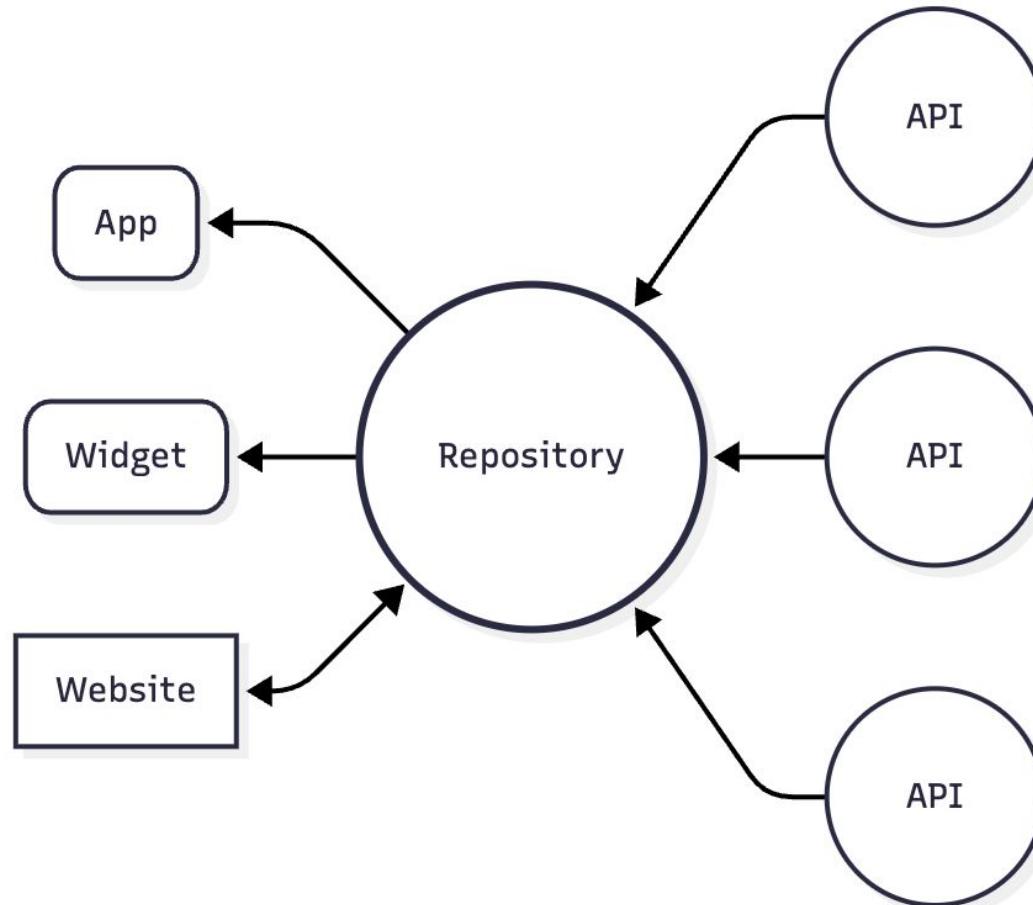
Activity Diagram (Behavior)



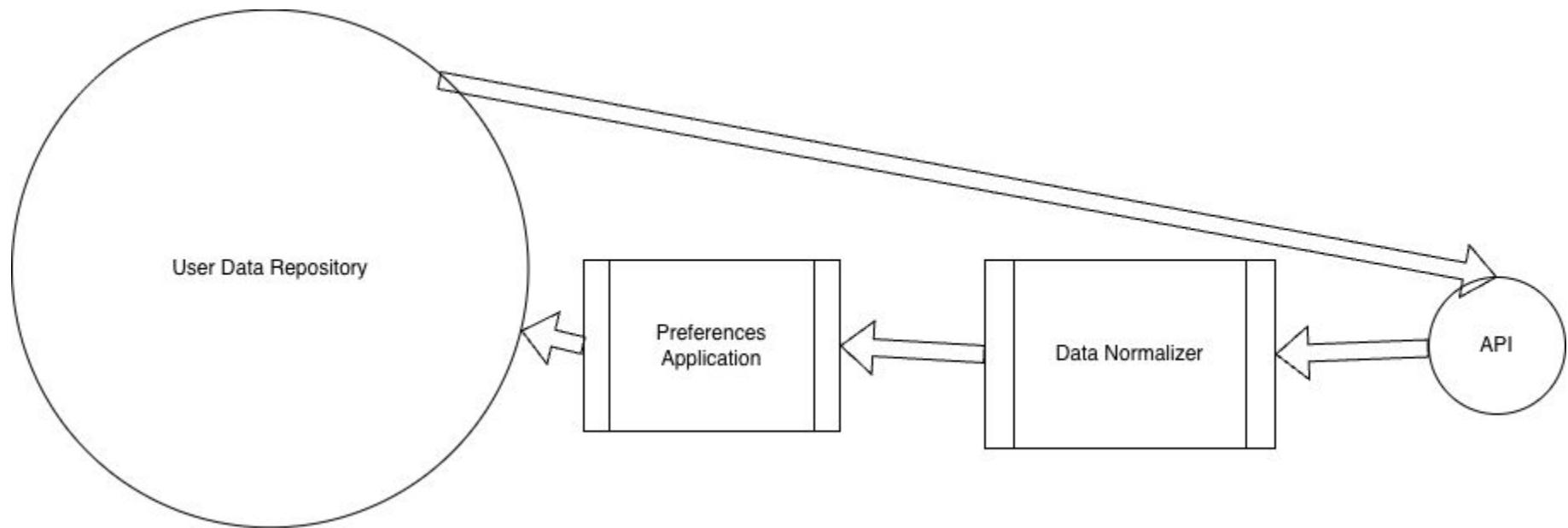
Use Case Diagram (Behavior)



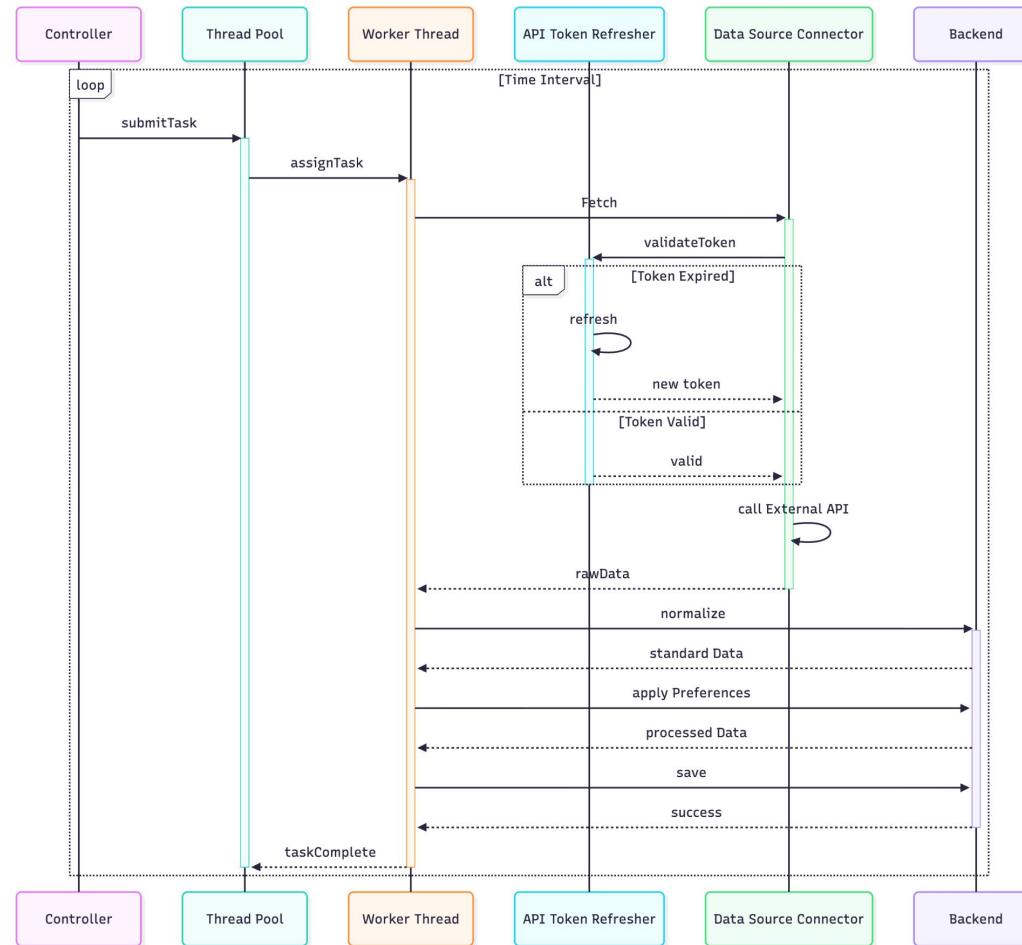
Architecture - Repository



Architecture - Pipe and Filter



Design Pattern - Thread Pool



Schedule and Task breakdown

3 ADD Iterations

1 - Establishing an Overall System Structure

2 - Refining Application Structure

3 - API Calling and Implementation

ADD 3.0 - Use Cases and QAs

Use Case	Description	
UC-1: Import Data from Multiple Sources	The user should be able to import data from multiple health services by connecting their account	
UC-2: Set preferences for data	The user should be able to set preferences on which data sources take priority during certain days and times	
UC-3: View aggregate step count in multiple ways	The user should be able to view their daily step count in the app, within a widget, and by notification	
UC-4: Data from sources should be pulled automatically	Data from the user's connected accounts should be pulled automatically at set intervals	
Quality Attributes	Scenario	Associated UC
QA-1	The User should have the ability to deleted any and all data as requested from our service	UC-1
QA-2	The user should have a log of all actions taken within the app to verify security	UC-2

ADD 3.0 - Constraints and Concerns

Constraints	Description
CON-1	API standards must be maintained
CON-2	User data must be secure
Concerns	Description
CRN-1	Some API's may require frequent updates to keep active key.
CRN-2	Some API's are only available on certain devices (Android, Apple)

Open Source & License

Github:

<https://github.com/aaron730/Coalesce>

Allow developers to add different health app data pipelines

MIT License

Research

- **Consumer API Convergence Across Industries and The Future of APIs in an AI Web**
 - Compare how similar external APIs are across industries
 - Explore structure and depth and see how each industry has converged
 - Also view the current state of APIs and how they must evolve to work with AI

Research Questions

- 1. What is the state of API Standardization across global industries?**
- 2. How do these industry-specific standardization drivers dictate the required strategic architectural approach for enterprise API integration**
- 3. What are the foundational technical protocols and tooling that form the technical bedrock for modern industry APIs, and what are the key emerging trends**

Research Methods

- **Comparative Analysis: The core method was to compare the findings across the ten sectors to draw universal conclusions.**
 - Count API Required fields
 - Look at max nesting dept
- **Connect Research to other results found to gain broader picture**

Research Conclusions

- **Industries that are more heavily regulated have more API structure.**
- **Some APIs forgo structure and depth for Performance.**
- **Commercial Industries rely on third parties for normalization.**
- **Industries are coalescing around a small group of API technologies (JSON, REST, OAuth)**

APIs with AI

- **Found some industries like Travel that are already looking to make APIs more AI friendly.**
 - Wrappers
 - Ability to have AIs ask API calls what they do in real time

Questions?



University of Colorado
Colorado Springs



University of Colorado
Boulder | Colorado Springs | Denver | Anschutz Medical Campus