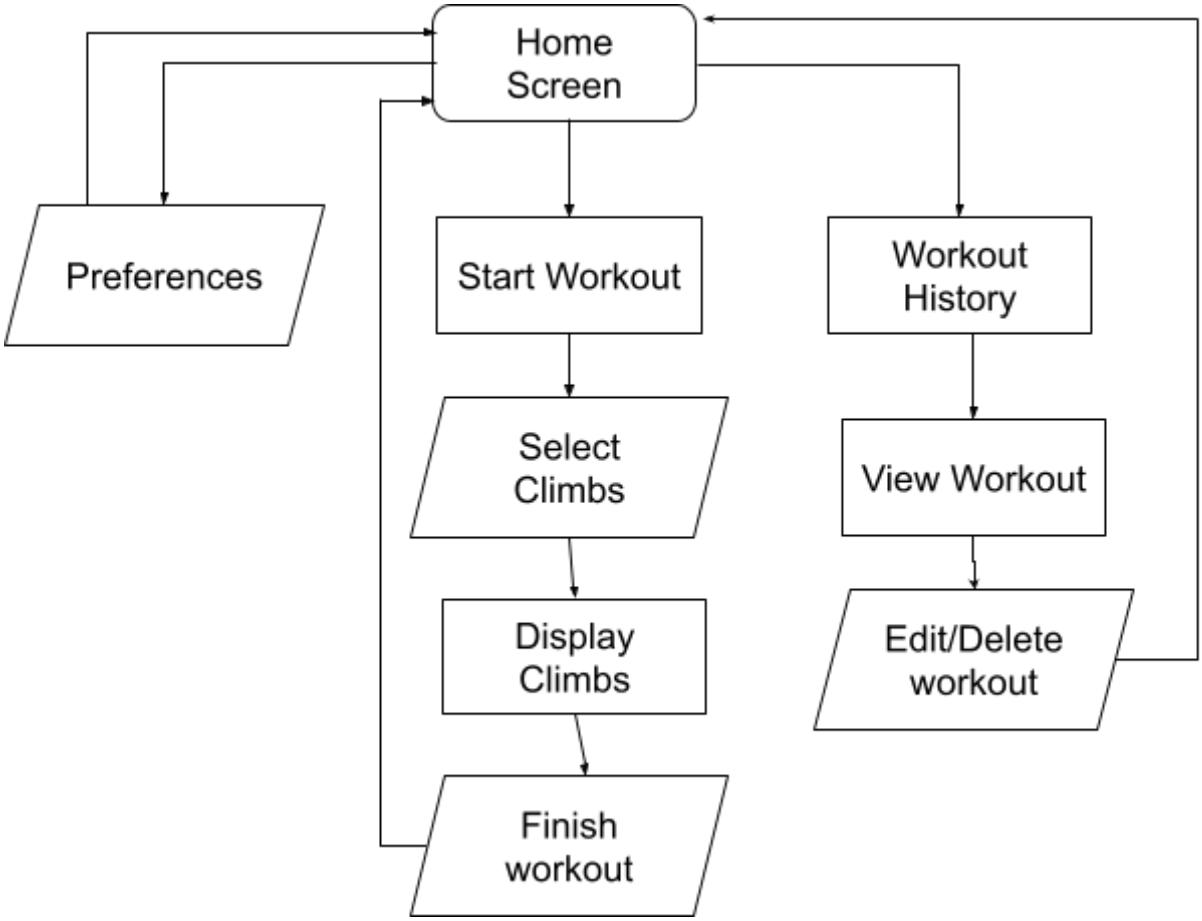
Criterion B: Design

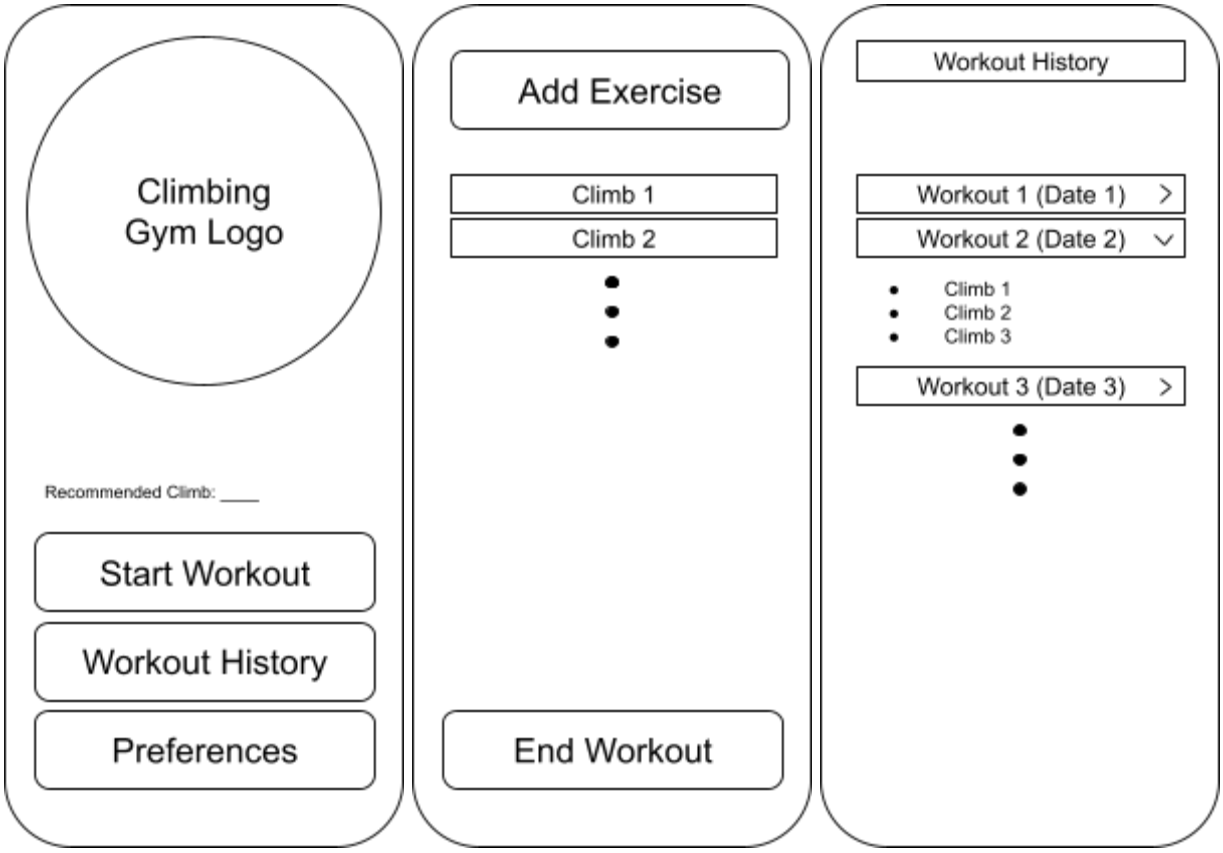
Introduction:

The planned architecture for this will involve a database, server, and user interface (referred to as the application). This criterion will cover a flowchart of the user interface, a flowchart of information between the three platforms, a sketch of the database structure, and finally, an algorithm (in pseudocode) to recommend a user a climb.

Flowchart of the User Interface:

The flowchart below represents a user's possible routes while using the application.



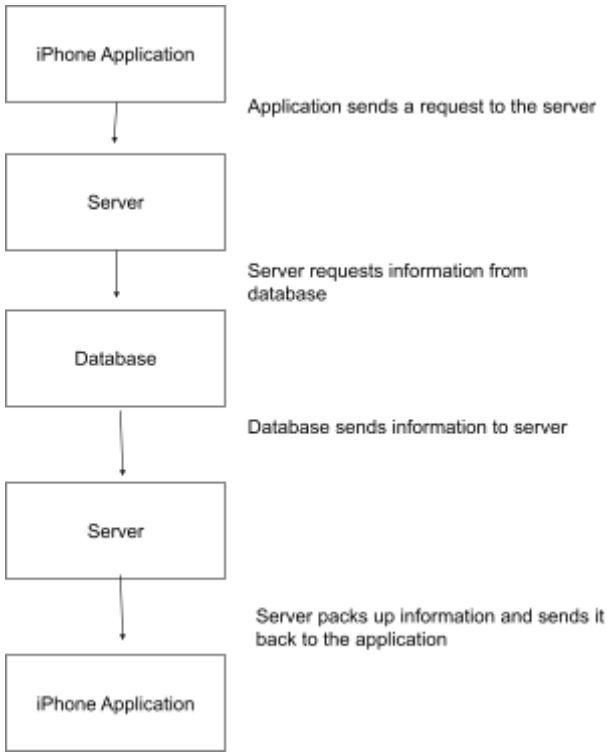


The application will start on the home page, where three buttons will allow the user to access the “Start Workout” or “Workout History” pages, as well as access the user’s “Preferences”.

When the user starts a workout, they will be able to add exercises by pressing the “Add Exercise” button, bringing them to a temporary search menu where all climbing routes present in the gym are organized to be selected. When the user is finished, they will press the “End Workout” button, bringing them to a summary page where they can share their workout with others.

Clicking the “Workout History” button will allow users to see their workout history over time. When a user taps a workout, they can see the specific climbs that they have done on that specific session.

Flowchart of Information



My application will require multiple layers to achieve a proper flow of information. The flowchart above shows a rough outline of how this will be accomplished.

Database

The climbs will be accessed and managed by the gym's staff using a database. Each climb has a climbID, name, color, location, grade, and status of whether they are active or not. The colors and locations' names will be called upon from the climb table. The user's workouts, workout durations, and workout dates will be stored locally on the phone and cannot be accessed by gym staff.

Climbs	Colors
ClimbID	ColorID
ClimbName	ColorName
ColorID	
LocationID	Locations
Grade	LocationID
IsActive	LocationName

Recommended Climb – Algorithm

This will recommend a climb to the user based on their current level (indicated in the preferences page) and their recent workout history. This algorithm will also order the “Add Exercises” page.

```
input USERLEVEL, CLIMBSLIST
DIFFERENCES = []
SORTEDCLIMBS = []

// Calculate the differences and store them
for each CLIMB in CLIMBSLIST:
    DIFF = |CLIMB.GRADE - USERLEVEL|
    Append (DIFF, CLIMB) to DIFFERENCES

// Sort DIFFERENCES based on the DIFF value
Sort DIFFERENCES by each entry's DIFF value

// Reorder CLIMBSLIST based on the sorted DIFFERENCES
for each (DIFF, CLIMB) in DIFFERENCES:
    Append CLIMB to SORTEDCLIMBS

output SORTEDCLIMBS
```

TestPlan:

Success Criteria	Method of Testing
1. Program will allow staff and administrators to create ‘climbs’ and label them with their important criteria (grade, type, color, etc.) 2. Program will update with new climbs added by administrators	Add a climb using MySql workbench. Verify that the climb information is visible in the ‘add Climbs section
3. Program allows a user to ‘start a workout’, starting a timer and allowing users to pick exercises and climbs 4. User is able to delete climbs which were accidentally selected	Pick a climb from the ‘Add Exercise’ section. Ensure it appears in “Today’s Session” after choosing. Delete a climb after selecting it.
5. Program allows the user to end workout, saving all of the climbs and exercises done are saved to today’s workout. If the workout is empty, the workout will be canceled with no data saved. 6. User is directed to a ‘workout summary’ page where the user is able to share the workout information.	Ensure that workout can be canceled before exercises are added. End workout and ensure all workout details are visible in the “Workout History” page. User sees the workout summary page and can successfully share workout information.
7. User is able to adjust preferences	Users can access the preferences page by tapping the ‘Preferences’ button on the home page. Users can change their name and current level of climbing and the main page and algorithm updates with this information.
8. User receives suggested workout on login	User receives a climb suggestion based on user preferences and workout history. Climb search bar is sorted by this algorithm.

Word Count: 322