BAMZON
Sprint 2 Review

CSC 308
Fall 2018

Daniel Heyde | Brandon Ryan | Kevin Krein | Kellie Banzon | Kyle Vu

BZN

Table of Contents

BZN

User Requirements

- Members will be able to join teams via an invite sent by an Admin.
- Members will be notified of a change in the schedule via push notification.
- Members will be able to view a team roster, filterable team events calendar, team practice schedule, and personal attendance.
- Members can RSVP to events listed on their team's events calendar.
- Members and admins can view team records.
- Members and admins can view other members' statistics and biographies.
- Members can update their own biographies.
- Admins can update team records.
- Admins can update members' statistics and biographiess.
- Admins can create teams and invite members.
- Admins will be able to send out push notifications to their team(s).
- Admins can add events to their team's events calendar.
- Admins can adjust their team's practice schedule.
- Admins can view and update attendance.
- Admins can view members' RSVPs (accept/decline) on team events.
- Admins can remove members.
- Admins can make other members admins.

Functional & Non-Functional Requirements

- Functional
    - Admins will be able to create and edit descriptions, times, and locations.
    - Admins will be able to invite members by link or email.
    - Admins will be able to invite other members to be admins.
    - Admins will be able to manage rosters by adding or removing members.
    - Admins will be able to view and take attendance for all members.
    - Admins will be able to view everyone's RSVPs for all events.
    - Members will receive an individual push notification for each update to the calendar/schedule. The push notification will name which event has been updated, and clicking on it will launch the screen to that calendar/schedule item.
    - Members will have access to their attendance for the team they are registered for.
    - Members will have access to view the team roster.
    - Members will have access to view the team's practice schedule and events.
    - Members will be able to register/RSVP for events through the calendar and a linked registration page.

BZN

- Non-Functional
    - Member shall obtain an invite from an admin to join the team.
    - The calendar will update and display new information within 1 minute of changes.
    - Members will receive push notifications about calendar and schedule updates within 1 minute of the change.
    - Members will be able to register/RSVP for events before the registration deadline.

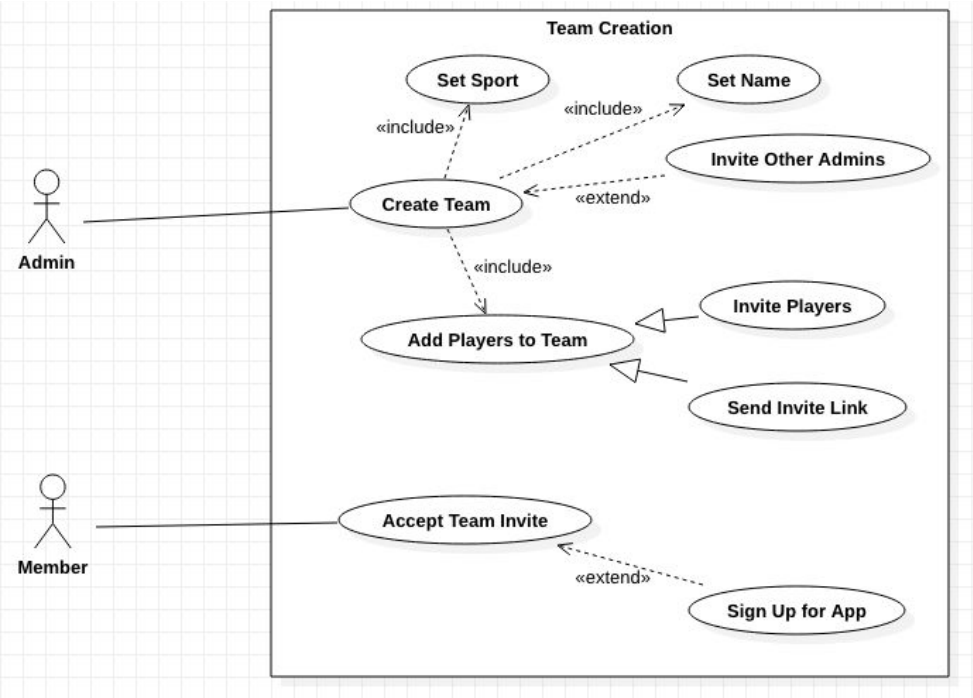
User Stories & Acceptance Criteria

1) As an admin, I want to be able to create teams so that I can add members later
    a) I can start a team by naming it
    b) I can specify a sport or activity for my team
    c) I can invite other users to be admins
2) As an admin, I want to be able to manage rosters in order to keep track of everyone
    a) I can view the members on my team
    b) I can invite members to join an existing team via email invitation or expiring link
    c) I can remove members from the team
3) As an admin, I want to be able to push notifications to team members so that team members stay up to date with new changes
    a) I can send a notification to my team members
    b) A clear and concise notification is displayed for team members
    c) Notifications can be clicked to view the event in question
4) As an admin, I want to be able to schedule regular practices for my members to see so that they can stay updated about practice times and locations
    a) I can create a recurring practice or meeting that my team can see
    b) I can set a time and location for recurring practices or meetings
    c) I can modify times and locations for recurring practices or meetings
5) As an admin, I want to be able to schedule events outside of regular practices so that my members RSVP and stay up to date with changes
    a) I can create a new event with a description, time, and date
    b) I can change the description, time, and date of existing events
    c) I can see which members have RSVP'd to events
6) As an admin, I want to be able to change stats and records for my team so that my members can keep track of their growth
    a) I can add and update statistics and records for a member
    b) I can add and update statistics and records for my entire team
7) As a member, I want to be able to see my team's roster
    a) I can see a list of who is currently on my team
    b) I can see what position each member on my team is playing

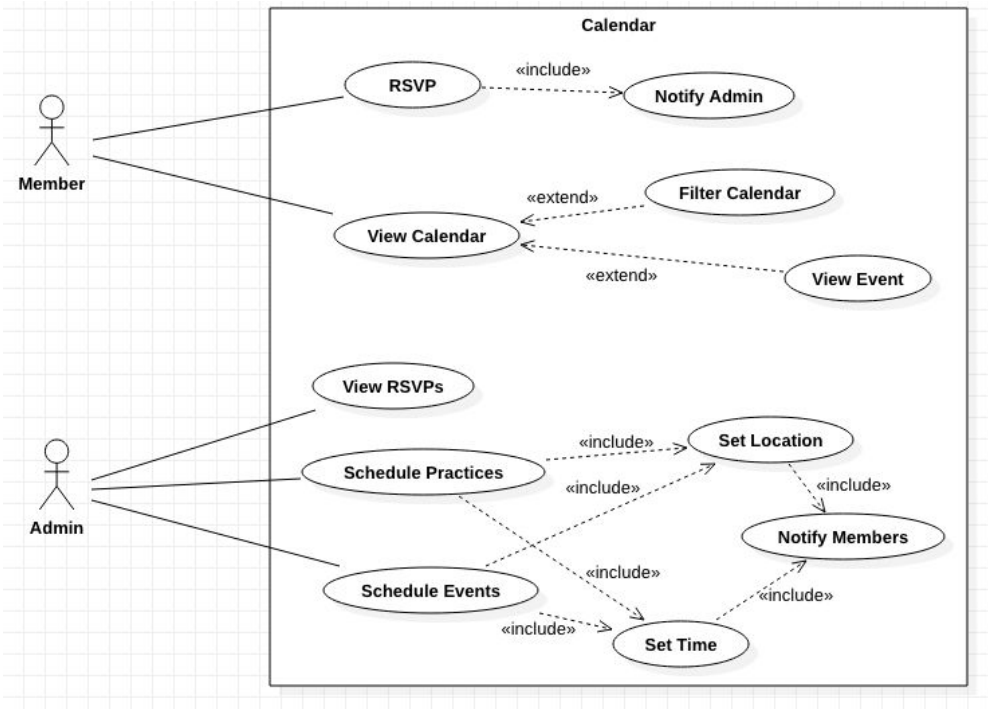
8) As a member, I want to be able to view and update my schedule so that I can stay up to date with my team
    a) I can see the schedule in a month, week, or day view
    b) I can RSVP to special events
    c) I can change my RSVP later if I choose to
    d) Team admins are notified of my RSVP
9) As a member, I want to be able to check myself into events and practices so that my admins can keep track of attendance
    a) I can select a current event or practice and mark myself as present or not present
    b) I can select a previous event or practice and mark myself as present or not present

Use Case Diagrams

Use Case Diagram 1: Create a New Team



Use Case Diagram 2: Calendar



Risk-Driven Specification

1. Push notification malfunction
    a. Risk Identification
        i. Event notification is not sent to member.
    b. Risk Analysis and Classification
        i. Depending on the event, this could be a high risk scenario.
    c. Risk Decomposition
        i. Risk: Notification not received
        ii. Causes: Many possible causes. Possibly not connected to internet, or a code bug.
    d. Risk Reduction and Assessment
        i. If a user does not receive a notification then they may miss important events like form due dates or a race/game time change.
        ii. Thoroughly test notification pushing in different scenarios
        iii. Add functionality for offline notifications for events that are RSVP'd.
2. Non-admin with ability to add people
    a. Risk Identification
        i. Someone without admin permissions is able to add people to the team roster.
    b. Risk Analysis and Classification
        i. If the person added is malicious then it is high risk.

c. Risk Decomposition
    i. Risk: Malicious entity could gain access to members' profile information or other sensitive information about events in the team
    ii. Causes: Failed permission check on who can add members to the team, or an admin account may have been hacked
d. Risk Reduction and Assessment
    i. Check permissions when adding users, and securely store user passwords, and ensure admins protect their passwords.

Security & Safety Requirements

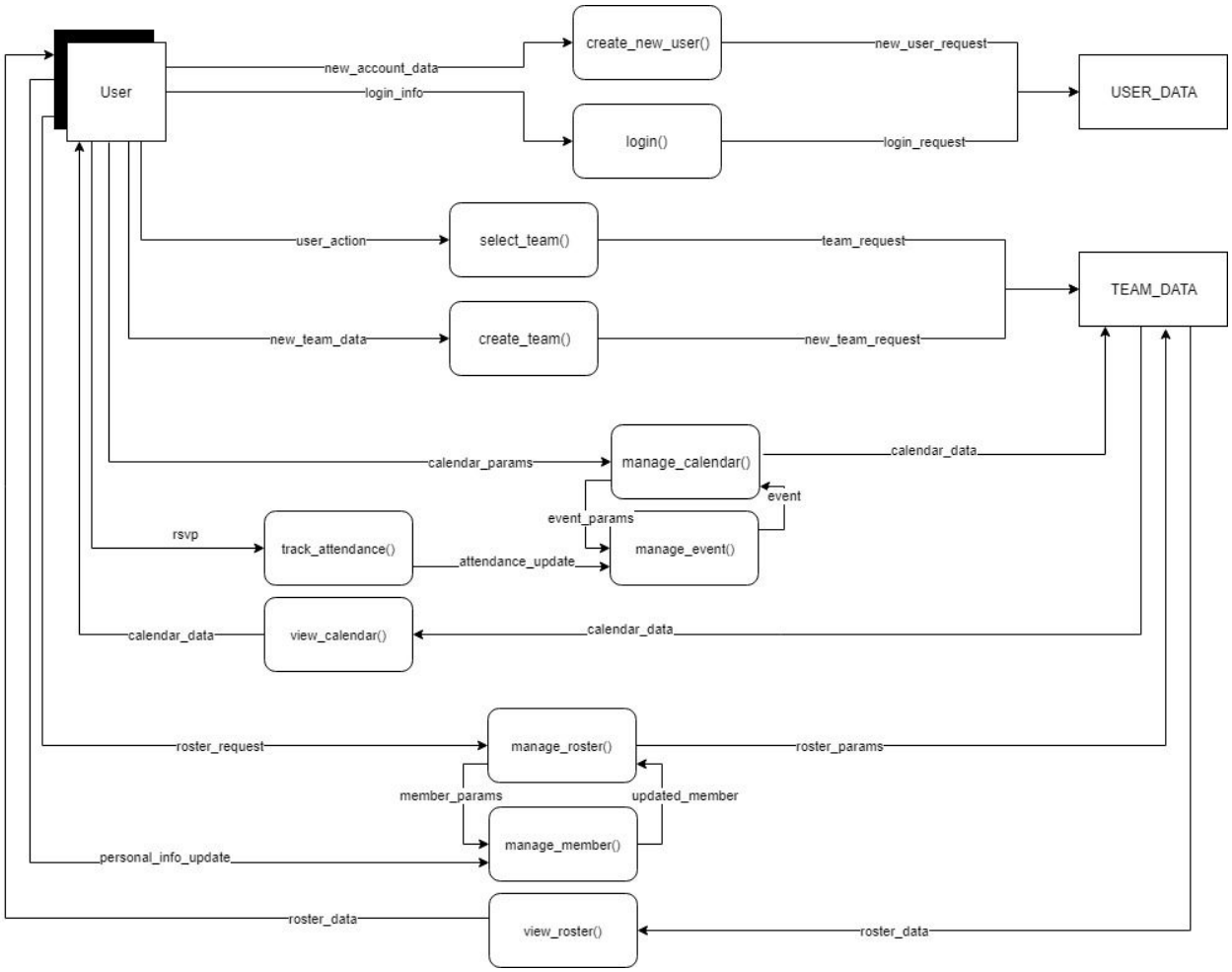
Security

- Only Admins shall add/remove people to/from the group
- Only Admins shall modify calendar events
- Only Admins shall have the ability to push out notifications
- Only Members and Admins can view roster and contact info for the team
- Passwords shall be encrypted

Safety

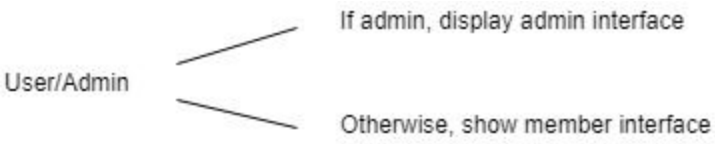
- Location of an event is a safe place
- If there is alcohol at an event, verify user is 21
- Users shall have the ability to report other users for harassment, bullying, or other inappropriate behavior
- Admins shall have the ability to remove users from the team if they are behaving inappropriately

Data Flow Diagram

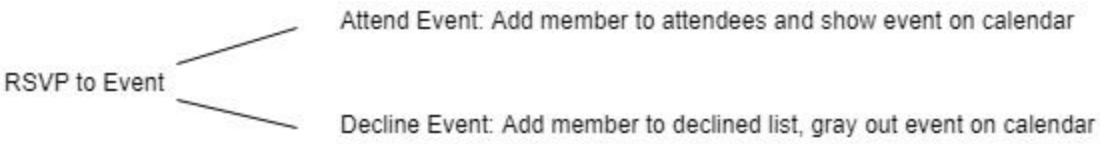


Decision Trees

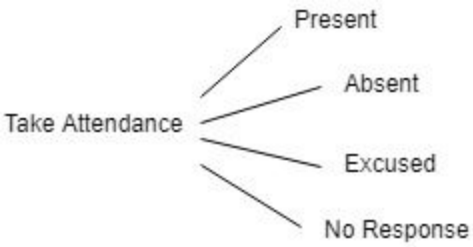
1. Permission Level



2. RSVPing to an event

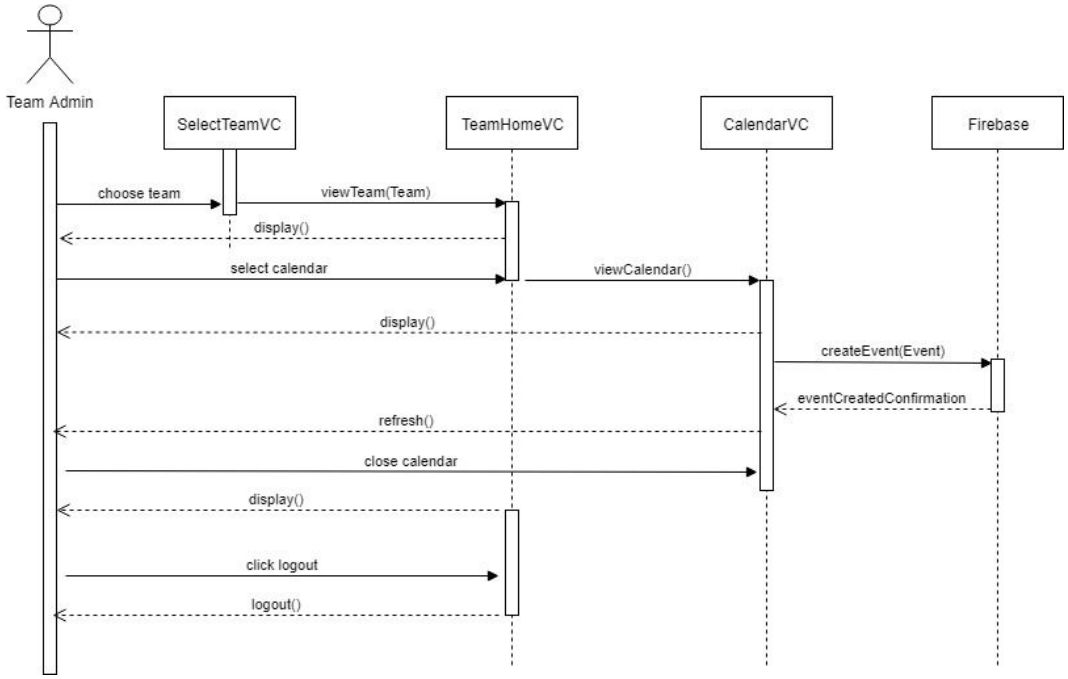


3. Taking attendance

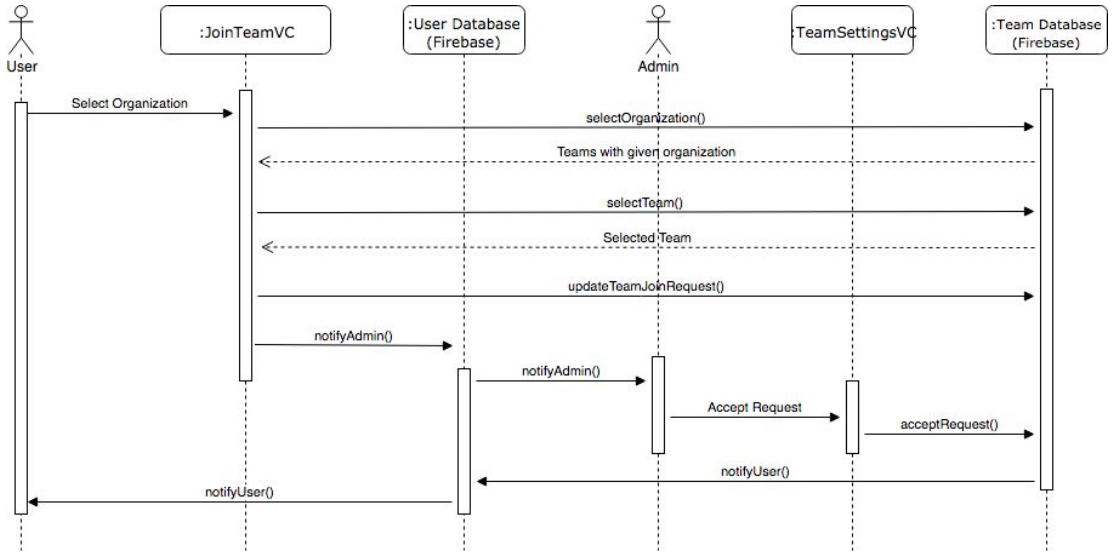


Sequence Diagrams

This sequence diagram shows how a user with admin permissions will add an event. Most interactions with the app follow this structure; the user will use SelectTeamVC to access their team, and they will use TeamHomeVC as a launcher to view the various team pages in the app.



This sequence diagram shows the process of adding a user to a team. First the user interacts with the view controllers and requests to be added to a team, then Firebase notifies the admin(s) of the team, and the request is accepted or denied.



User Interface

User Interface Design Criteria

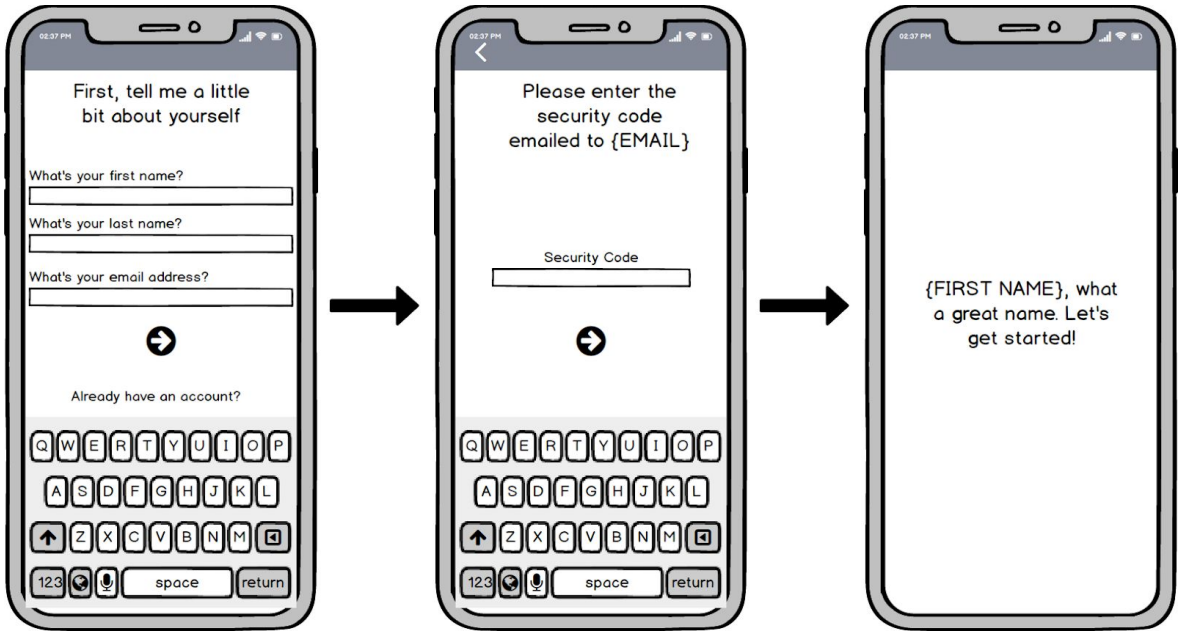
We used navigation to keep steps consistent throughout the interface. A nav bar is present when not in a wizard.

We also used a online help and additional documentation by allowing a user to look at FAQ's through the "Help" section or contacting us directly through the "Contact Bamzon" section.

Recognition not recall is used for our navigation icons, allowing users to navigate easily throughout the app

We also will keep our interface consistent to keep the "look and feel" the same throughout the app. Though it is not reflected in these wireframes, all screens will share consistent backgrounds, fonts, and color schemes.

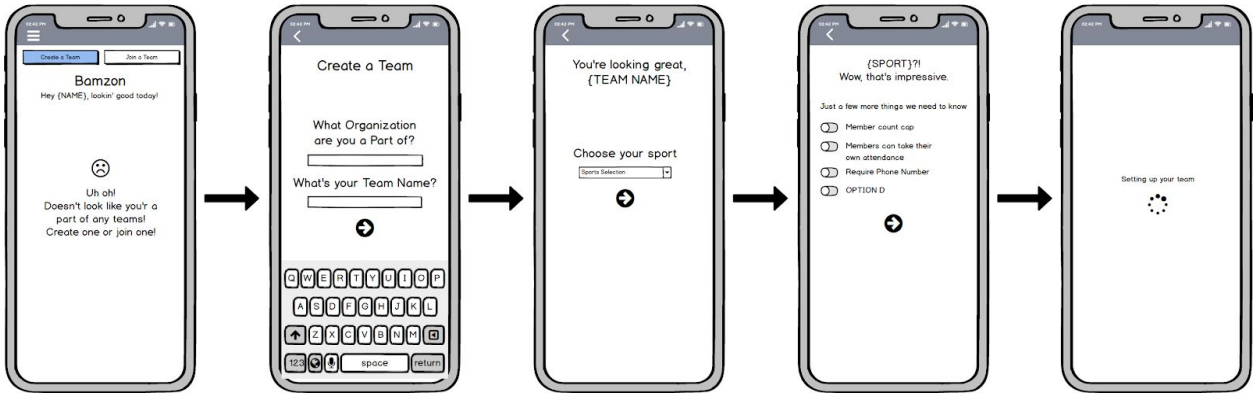
Create Account



Login

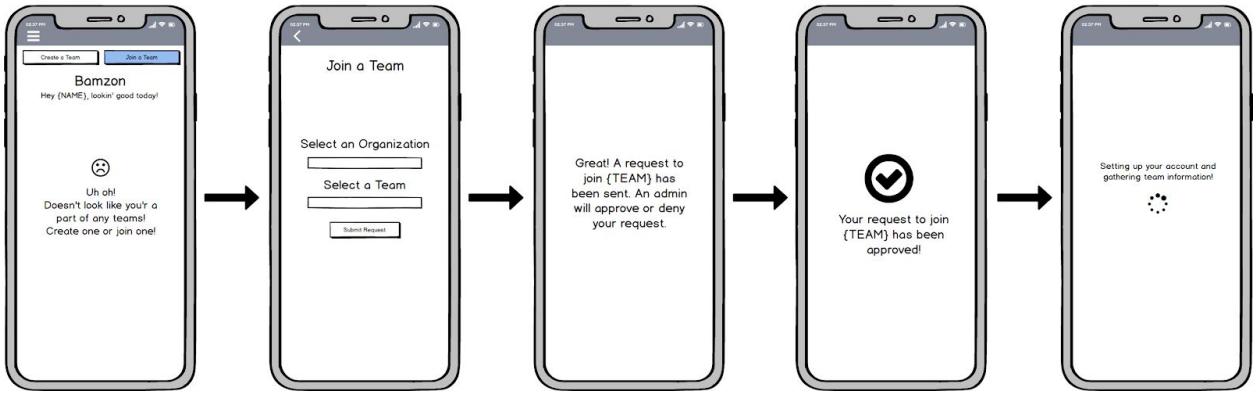
Create Team

Design Pattern: Autocomplete

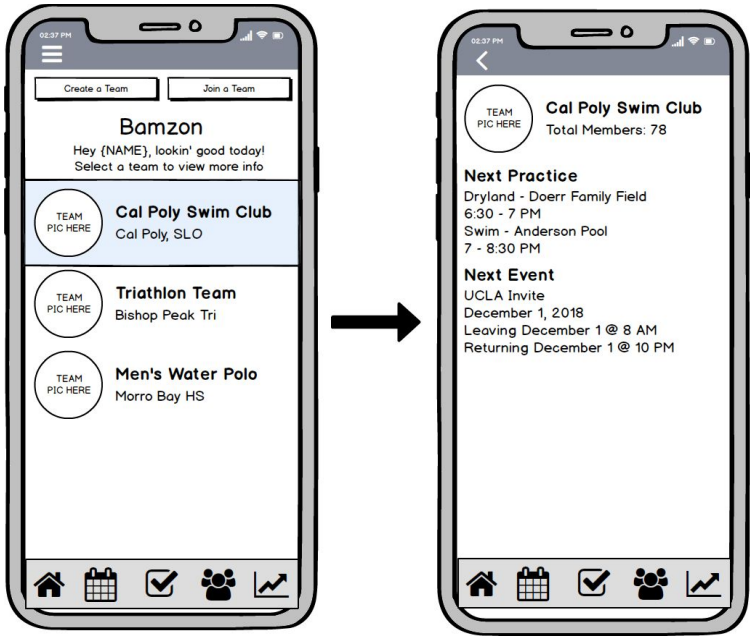


Join Team

Design Pattern: Autocomplete



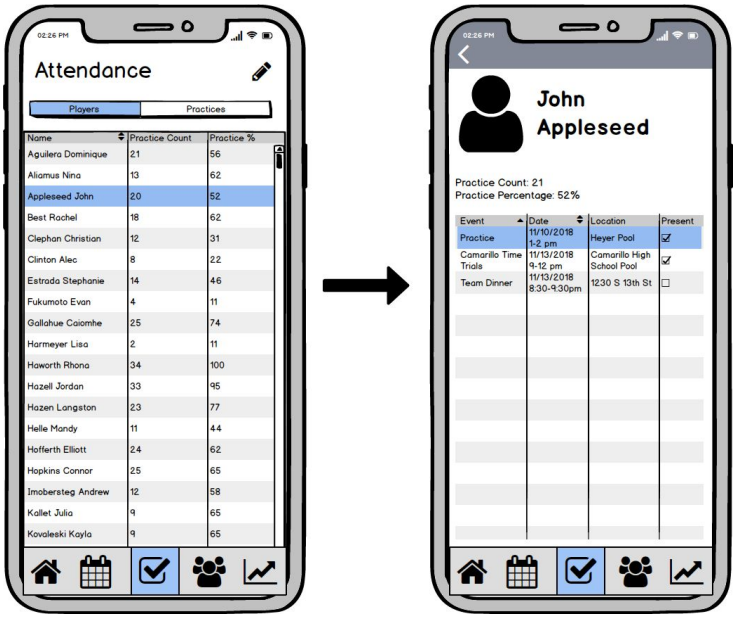
Home & Team View



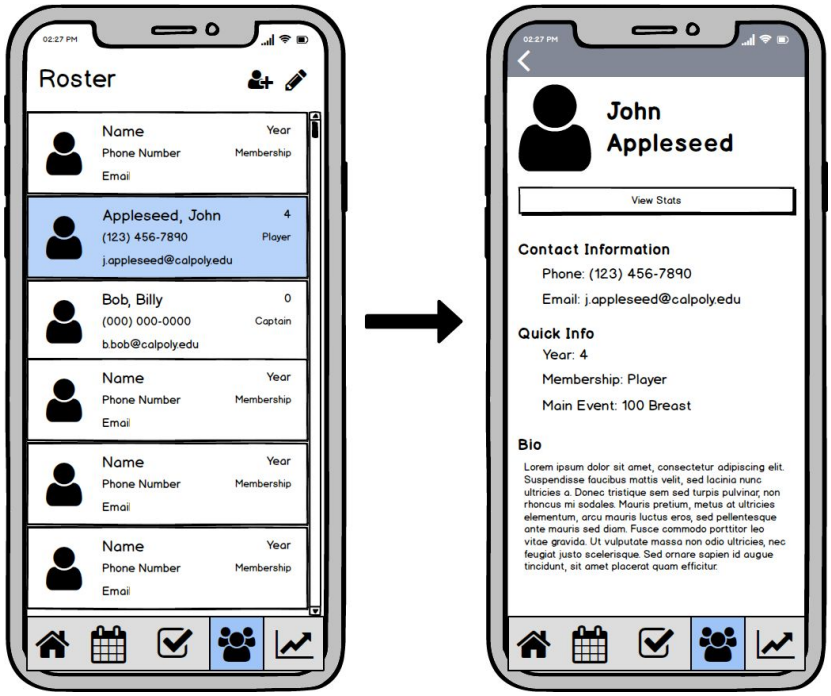
Calendar

Design Patterns: Event Calendar, Search Filters

Attendance



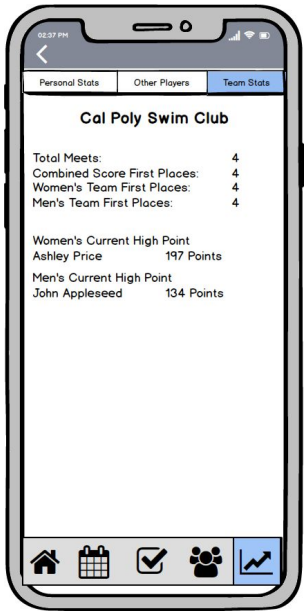
Roster



Personal Stats

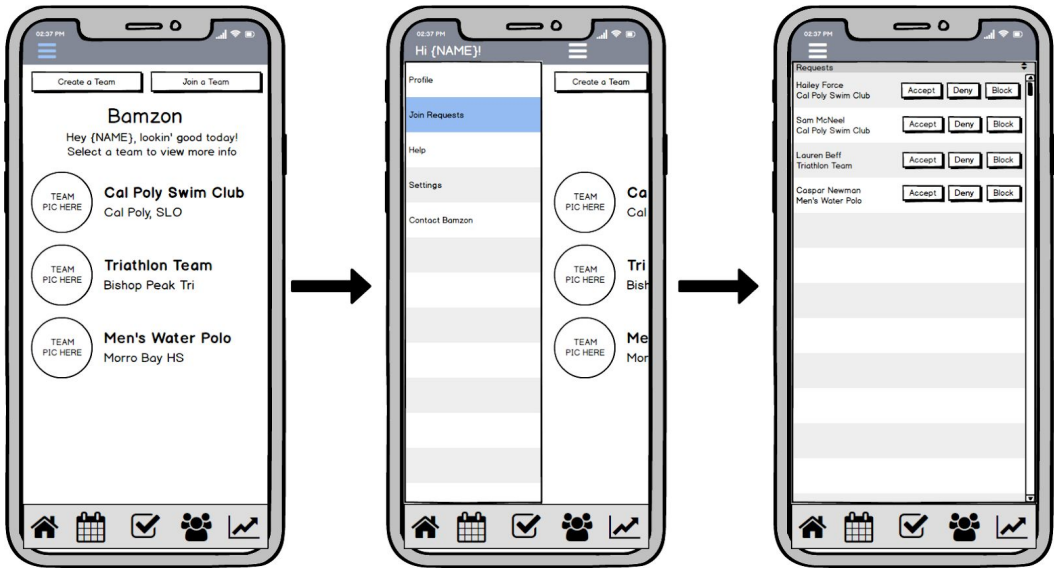
Other Player Stats

Team Stats



Settings & Join Team Requests

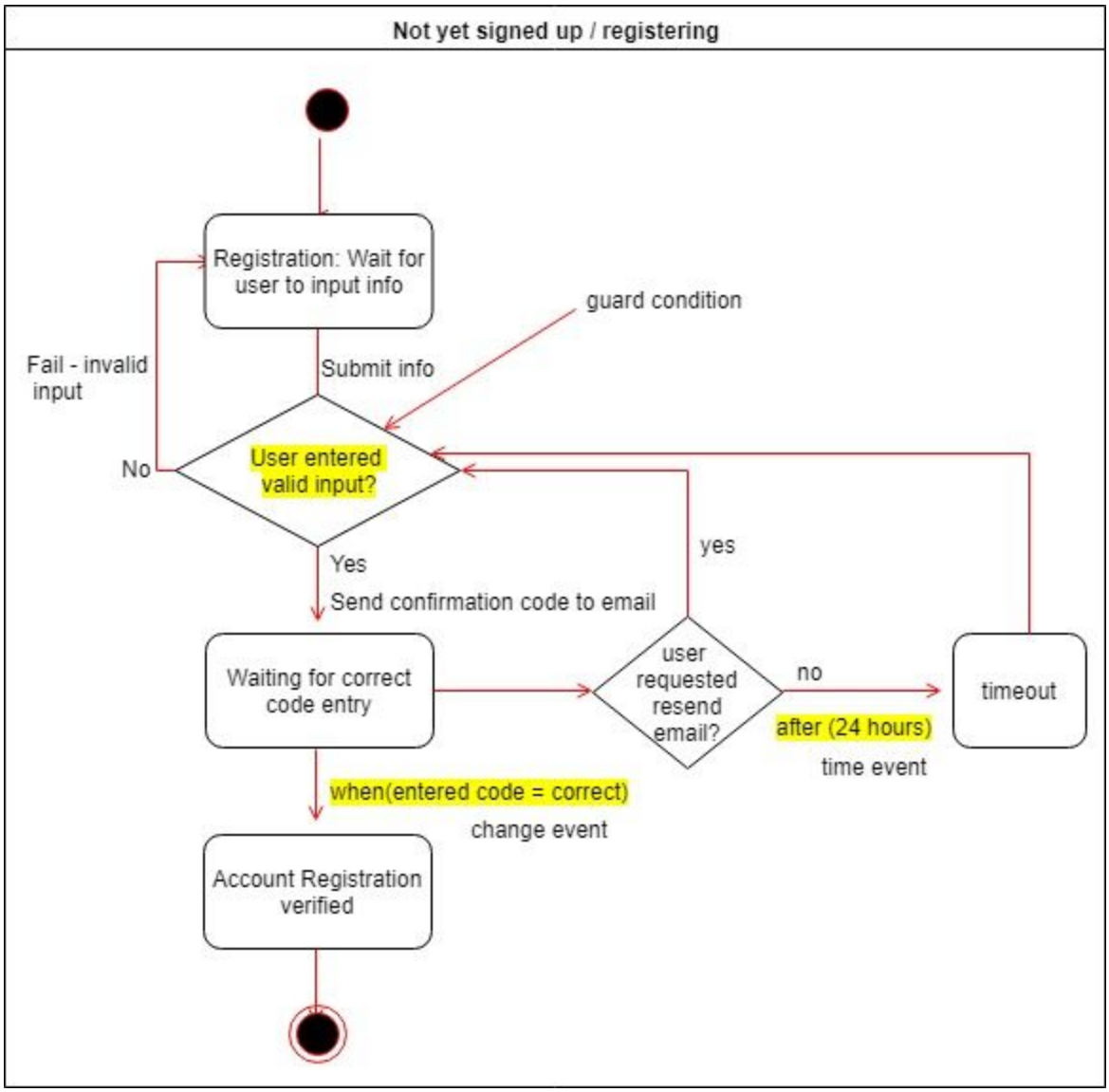
Design Pattern: Settings



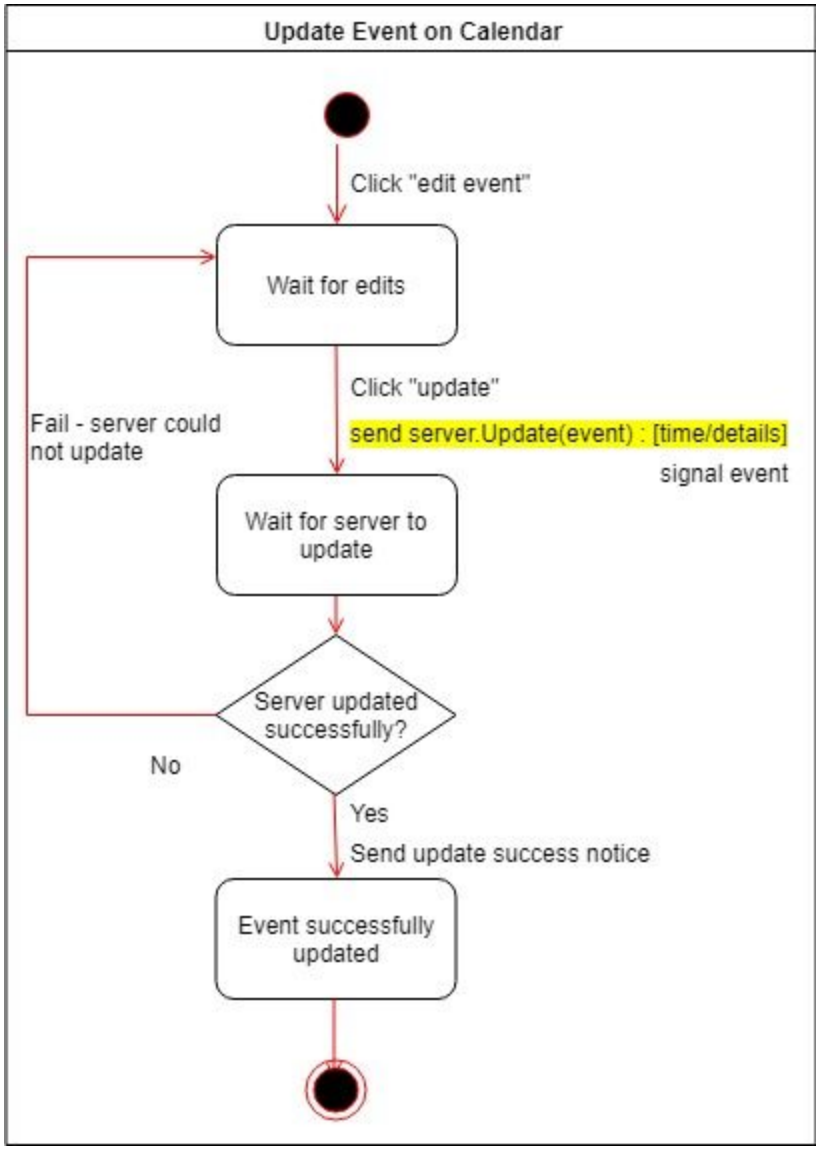
State Chart Diagrams

On the following state chart diagrams, highlights indicate different events and guard conditions.

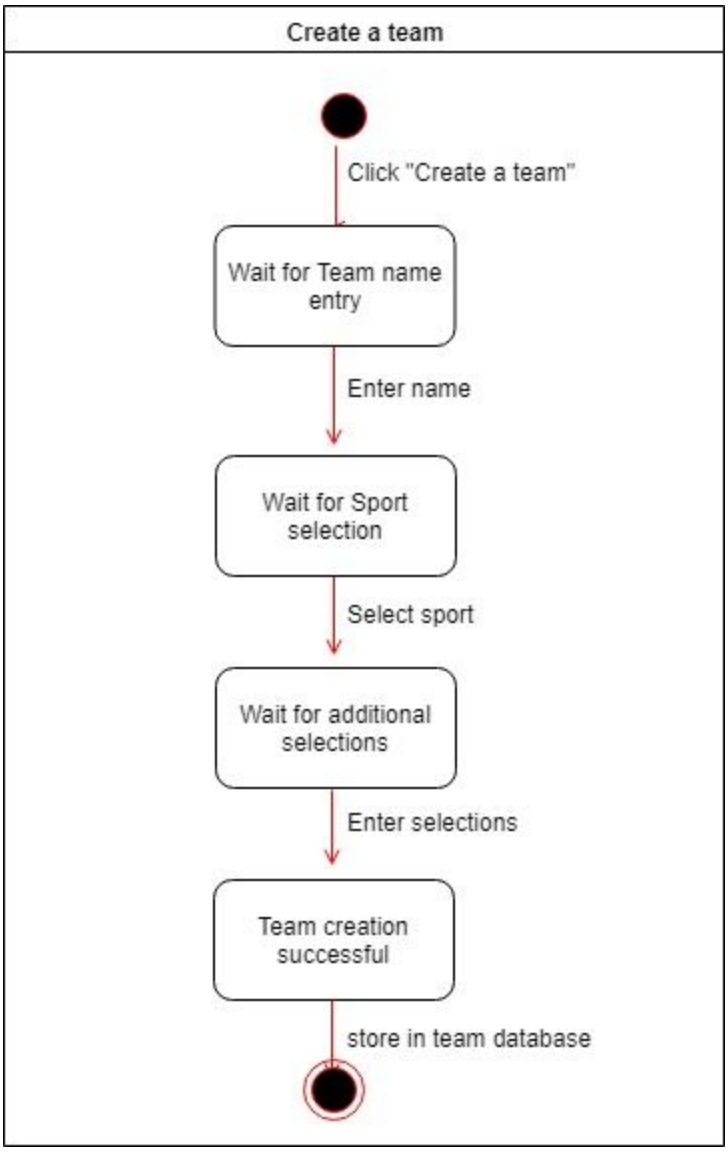
Registration



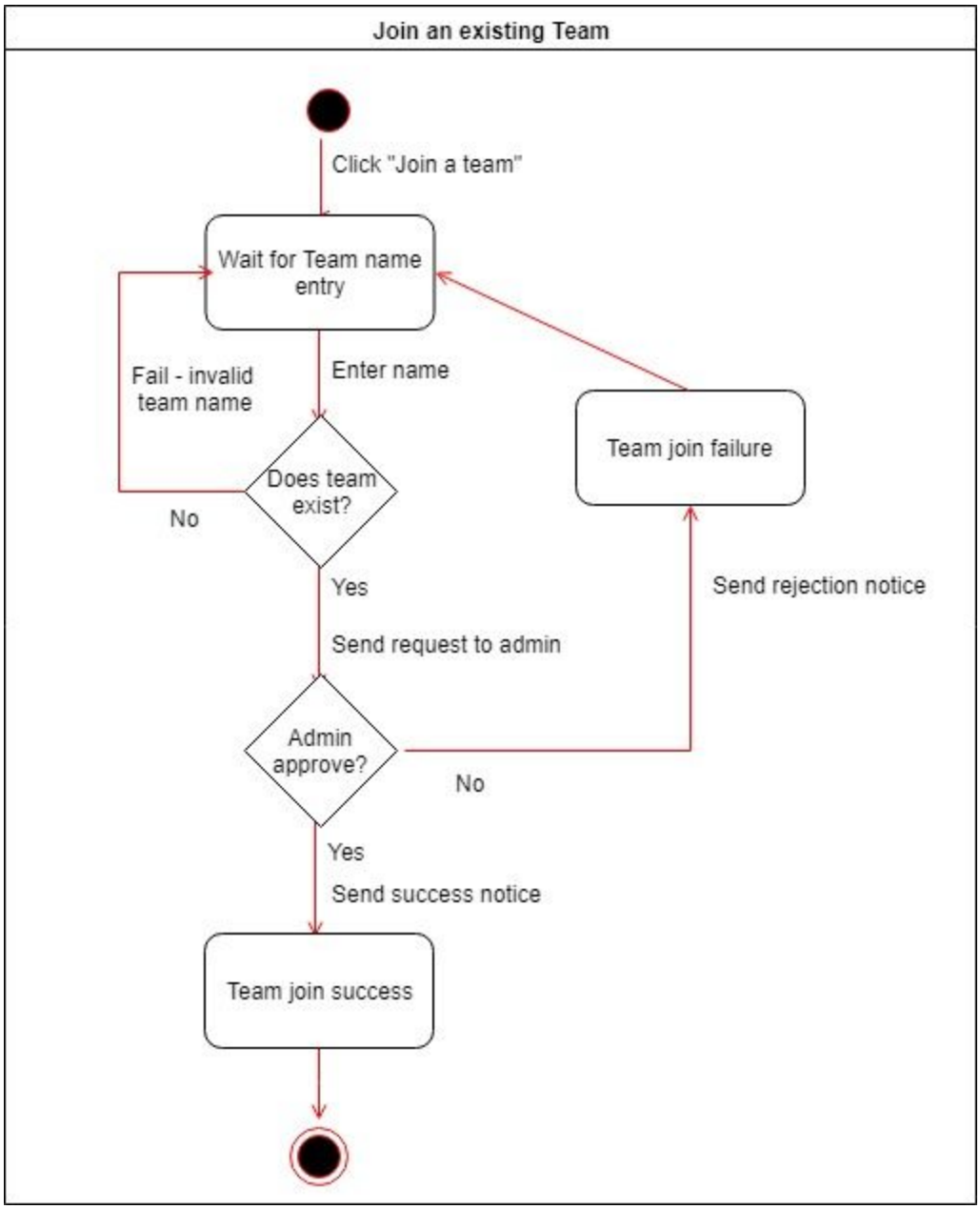
Updating a Calendar Event



Creating a Team



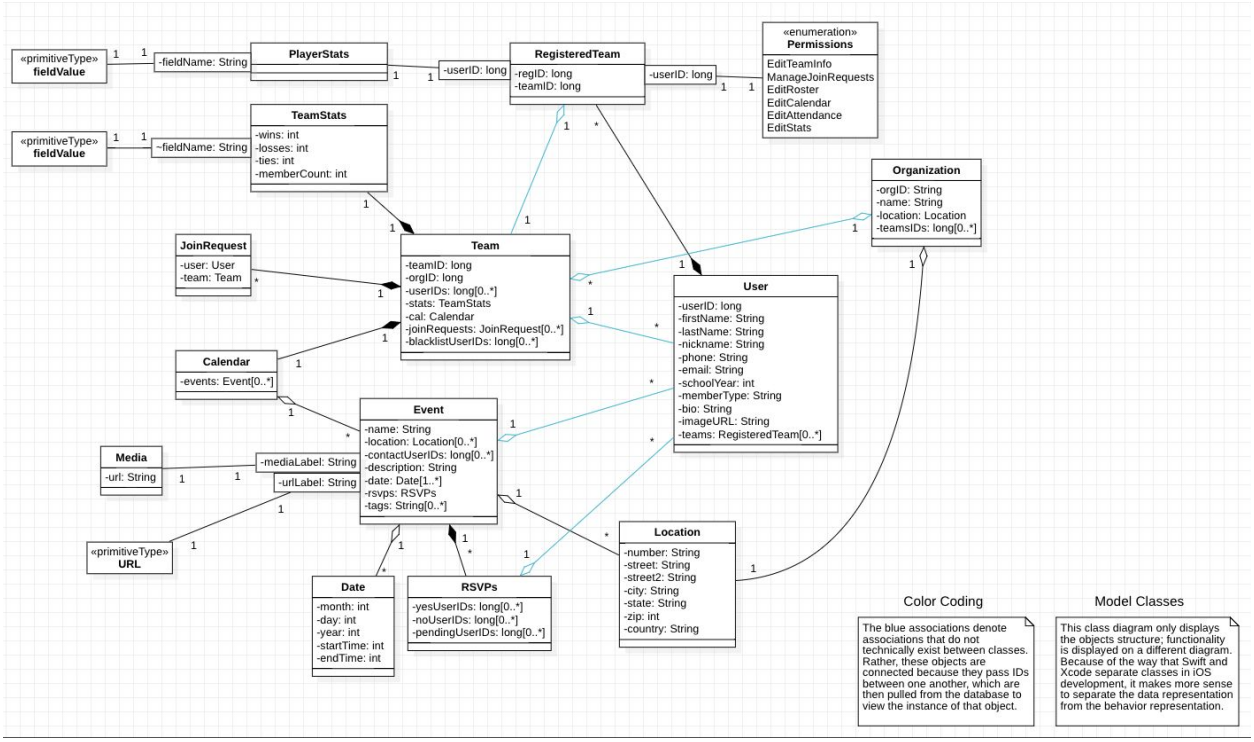
Joining a Team



Class Diagrams

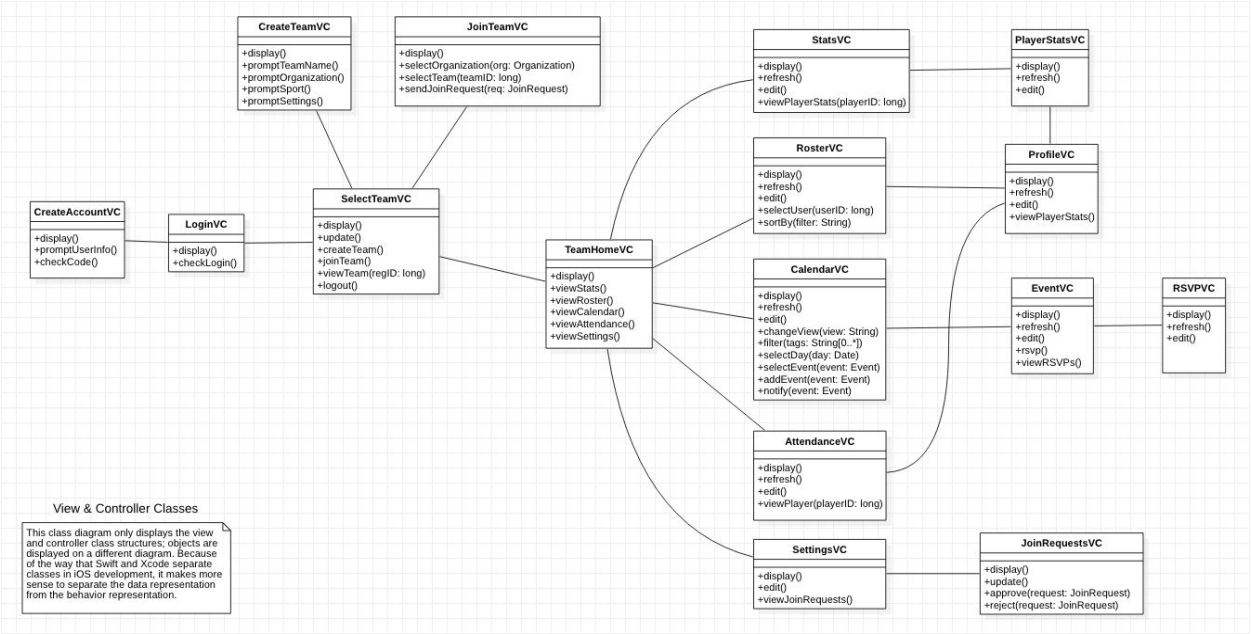
Class Diagram 1: Object Classes

This class diagram only displays the objects structure; functionality is displayed on a different diagram. Because of the way that Swift and Xcode separate classes in iOS development, it makes more sense to separate the data representation from the behavior representation.



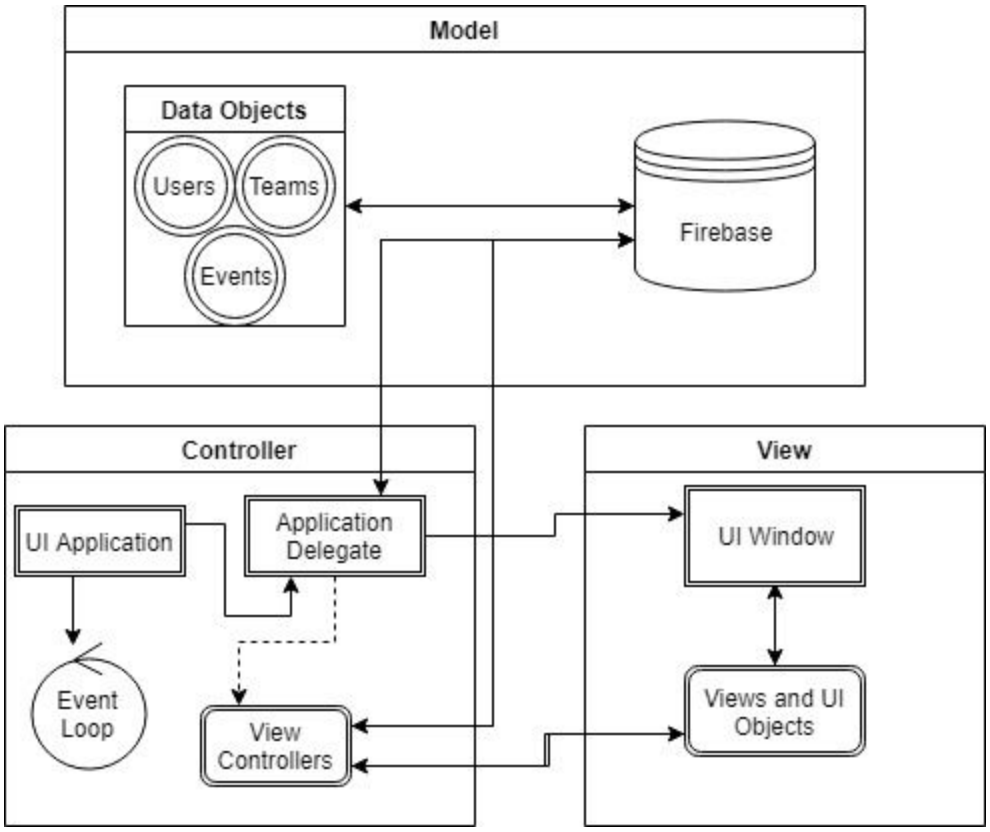
Class Diagram 2: View & Controller Classes

This class diagram only displays the view and controller class structures; objects are displayed on a different diagram. Because of the way that Swift and Xcode separate classes in iOS development, it makes more sense to separate the data representation from the behavior representation.



BZN

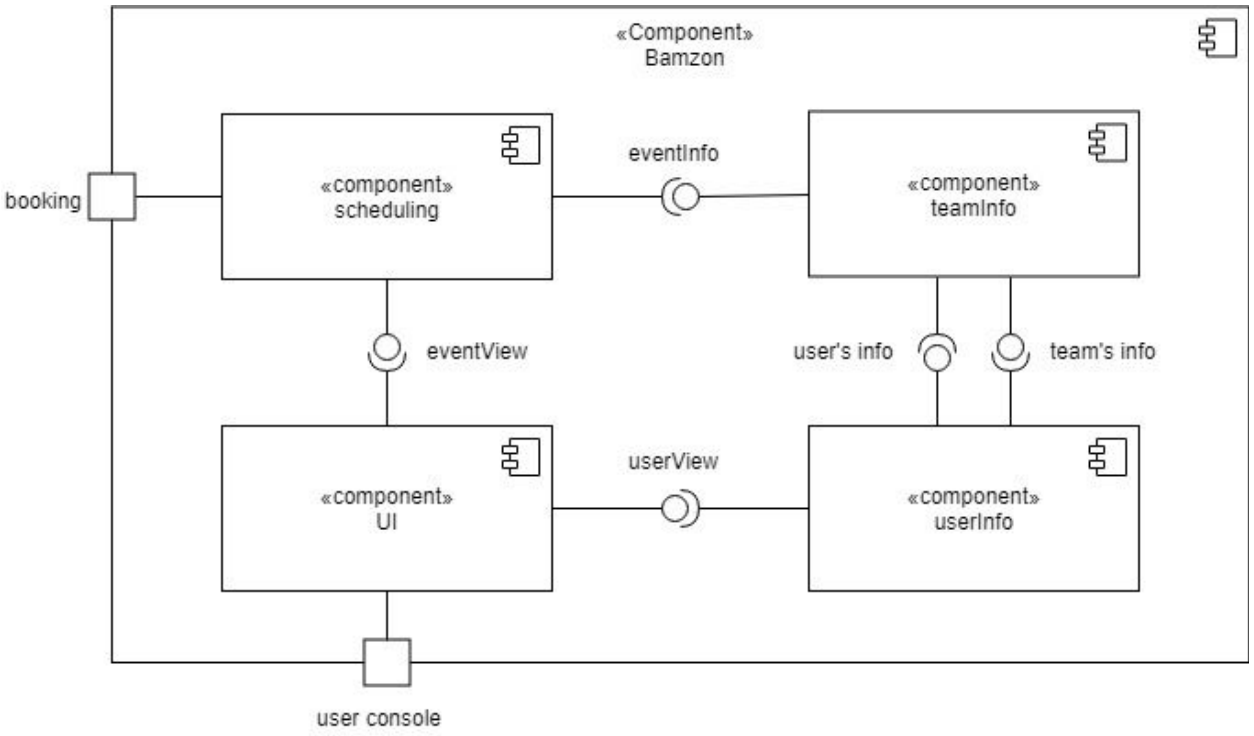
Software Architecture Diagram



We will be using several external frameworks in our application in addition to the standard iOS frameworks. These frameworks include JTAppleCalendar for all of our calendar and scheduling UI pieces and XLPagerTabStrip for our tab UI. We will be using the MVC framework as well. This organizational pattern is best for helping us efficiently update and query for all of our view controllers and works very well with how iOS applications are constructed in the back-end.

BZN

Component Diagram



BZN

Glossary

- Admin: User with moderator rights to the team. This could be a member of the team's board of directors (President, Vice President, Treasurer, Advisor, etc.) or a team captain.
- Member: User that does not have any moderator rights.
- Event: An event is a scheduled activity that is NOT a normal practice (e.g. race, game, party, trip, due date for registration forms).
- Practice: A normal, recurring workout session, NOT a special event
- Calendar: A calendar that lists all events and practices that the team hosts/participates in (e.g. games, tournaments, socials, etc.).
- Roster: A list of team members, their contact information, and their positions on the team.