

A3 README - Beta Phase

Contributions:

Eshitha Bangray (Release 25%, Overall 25%):

- Made the admin view controller and all UI functionality
 - Ability to add a new competition logo through the photo album
 - Competition name, location, judges text boxes
 - Scroll feature for date selection of the comp
 - New VC with a custom check box feature to select multiple teams for a competition
 - Horizontal scrollable table view under each comp to show competing teams
- Made the horizontal view controller on the competitions page
 - Includes the competition image, and some text fields
 - Added an edit button on the top right for admin control to edit a competition
- Added the swipe to delete and edit features on the teams tab for all the teams and on the competitions tab for the previous competitions
 - Added Firestore integrations to take care of deleted teams and competitions

Aryan Samal (Release 25%, Overall 25%):

- Competitions VC
 - Added Firestore + Storage integrations for both the previous comps table as well as the upcoming comps collections view
 - Touched up the UI components for both the table view and the collection view
- Create New Competition VC
 - Added Firestore + Storage integrations to update the database with the new competition data
- Elo VC
 - Fixed the UI for the Elo VC so that it displays rank, logo, name, and score
 - Added Firestore + Storage integrations as well as a sorting function to update the ELO ranks based on ELO score and display them
- Database Schema
 - Designed DB schema for the users, teams, and comps database tables
 - Set up Firebase Storage to store pictures + file directory

Akhilesh Bitla (Release 25%, Overall 25%):

- Competitions Tab:
 - Implemented Media Gallery Subview: Implemented a grid structure for rendering images for competition
 - Set up Firestore + Storage integrations to upload image and video data using the upload button

- ELO Tab:
 - Set up the initial table view and table view cell for the ELO tab and cleaned up additional constraints when viewing ELO information.
- Create New Team VC: I created a view controller for our create new team logic. This is one of our Admin pages, and I set up the necessary buttons and text fields required to enter the attributes for a new team.
- Auth Screens:
 - Refined auth navigation from login to sign up and vice versa.
 - Ensured password text fields are masked, and user authentication works without any errors.
- Constraints/UI Refinements (on all screens): Reviewed all screens for any UI refinements, added proper constraints, improved the UI for a smoother user experience, and improved overall application workflow.

Rhea Shah (Release 25%, Overall 25%):

- Competitions Tab
 - Upcoming competition section: horizontal scroll view with cards per comp
 - Segmented picker to filter upcoming comps by all, live now, month, week
 - Previous competitions: scrollable table view displaying comp name, logo, date
- Competition Details Page – launched when any comp card is clicked
 - Top portion stays the same: comp logo banner, name, location (links to google maps), date (add to calendar feature), chat button
 - Bottom content changes according to segmented picker: Lineup, Judging, and Media subviews
 - Lineup: competing teams (with ranks if comp is over)
 - Judging: judges names and links to scoring spreadsheets
 - Media: buttons to view official media and upload your own media
- Database Setup and Integration
 - Set up the users, comps, and teams tables and did the read and write connections
- Admin Views and Features
 - Created View Controllers to create new competitions and teams as well to edit them. Connected it to Database to display pre populated fields and save edits to the db.
 - Other admin features like selecting a lineup, editing judging info, and deleting comps and teams
- Add to calendar feature, instagram button linking, uploading images

Deviations:

In the Beta phase of our project, we successfully implemented the majority of our planned features, with the exception of the ELO algorithm. We integrated real-world competition and team data into the mobile app and presented it through a clean, user-friendly, view-only

interface. Additionally, we went beyond our original scope by implementing an "admin side" of the application. This feature enhances user engagement by allowing admins to interact with and modify app content, rather than simply viewing static data from Firestore.

Our user roles now include both admins and general viewers. Admins have the ability to create, edit, and delete team and competition data, while regular users can browse and view existing information. Note: Right now all users have admin privileges for testing purposes, we'll hide admin functionality in the final phase.

Much of our development time during this phase was devoted to building out the admin functionality, which involved several subcomponents and extensive CRUD operations (create, read, update, delete). As a result, and with Dr. Bulko's approval, we chose to omit the chat feature from our app.

While we were able to design the UI for ELO rankings, we have not yet implemented the full simulation feature, as it requires integrating a more complex formula based on current team standings.

Other key accomplishments include the addition of sorting for upcoming competitions, a search feature for teams, and enhancements that integrate with native iOS apps, such as adding competitions to a user's iPhone calendar, linking to social media like Instagram, and uploading/storing media from the Photo Library.

One known issue we plan to address in the next phase involves the rendering of competition cells on a team's details page. If a team is associated with two or more competitions, all are displayed correctly. However, when a team is linked to only one competition, that competition does not render as expected.

Last but not least, our keyboard dismissal works for all of the text fields. However, we used a search bar for our Teams VC to search for any particular team. However, we couldn't figure out how to dismiss the keyboard from the search bar. We thought we could just change the widget from UITextField to UISearchBar, but that did not work. We aim to fix this in the final phase and ensure that it works properly as well.