

Sko Play

Prancing Purple Ponies 201-1

Hyden Polikoff - Sung Kim - Nathan Straub - Randy Perecman - Paolo Castro - Thomas Deaner

Project Overview

A real-time, location-based application for creating pick-up games among CU students. Need a few more players to complete your game? Invite others on the app. Want to play but don't have enough people to join? Check the app to see what pickup games are scheduled.

Any CU student wanting to participate has only to sign up creating an account through their school email (to validate that they are a CU student) and can from there either host games at predetermined locations on campus or join game invites that have been pinged by other students.

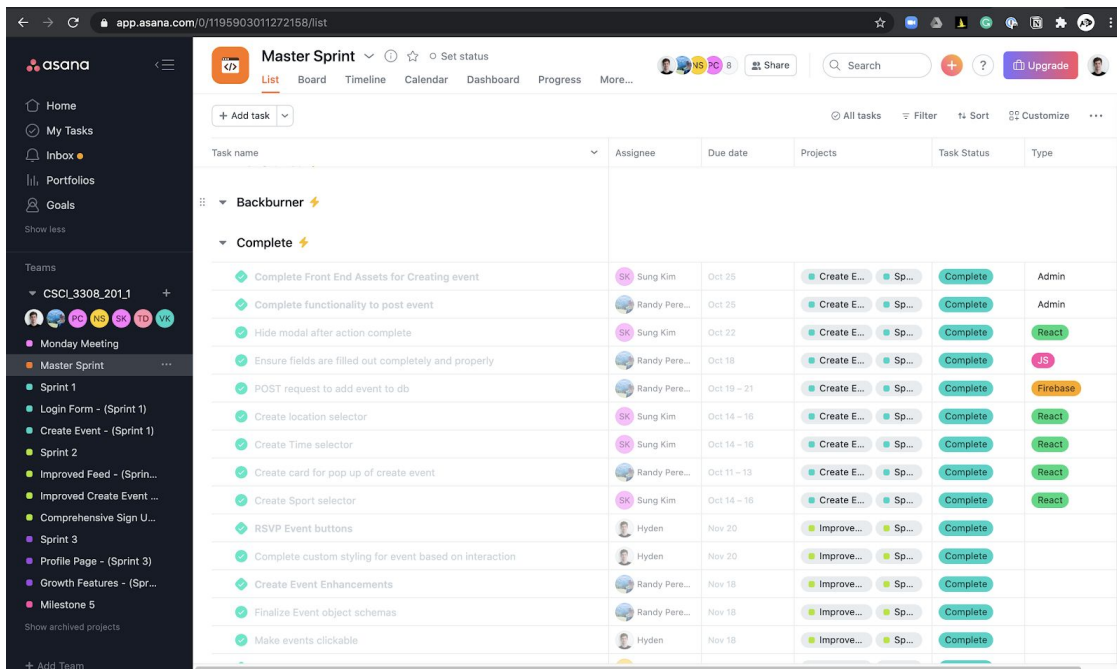
SKO Play also has students covered when it comes to socializing through sports. Once logged in, any student can update their profile page and update their name, major, favorite sport, and profile image. By doing so, students can socialize with other SKO Play users and make connections.

This application would benefit students wanting to play intramural sports that don't have the time to commit to weekly practices. Instead, SKO Play would open up the option of being able to play with other CU students at whatever time of day they choose. The app also works great for social networking and relationship building as sports lovers from all across CU will be able to interact and enjoy time together both on and off the field.

Project Tracker - Asana

[Here is a link to our Asana Project Tracker!](#)

Various projects correspond to different sprints and within each sprint our programming pairs had projects for just their specific tasks. You can also view a gantt chart under the timeline as well as a kanban style under the board view.



The screenshot displays the Asana Project Tracker interface for a project named 'Master Sprint'. The interface includes a sidebar with navigation options like Home, My Tasks, Inbox, Portfolios, Goals, and Teams. The main content area shows a list of tasks organized into sections: 'Backburner' and 'Complete'. Each task entry includes a task name, an assignee, a due date, and a status. The tasks are color-coded by status: green for 'Complete', yellow for 'In Progress', and red for 'Overdue'. The tasks are also categorized by type, such as 'Admin', 'React', 'JS', and 'Firebase'.

Task name	Assignee	Due date	Projects	Task Status	Type
Backburner					
Complete					
Complete Front End Assets for Creating event	Sung Kim	Oct 25	Create E... Sp...	Complete	Admin
Complete functionality to post event	Randy Pere...	Oct 25	Create E... Sp...	Complete	Admin
Hide modal after action complete	Sung Kim	Oct 22	Create E... Sp...	Complete	React
Ensure fields are filled out completely and properly	Randy Pere...	Oct 18	Create E... Sp...	Complete	JS
POST request to add event to db	Randy Pere...	Oct 19 - 21	Create E... Sp...	Complete	Firebase
Create location selector	Sung Kim	Oct 14 - 16	Create E... Sp...	Complete	React
Create Time selector	Sung Kim	Oct 14 - 16	Create E... Sp...	Complete	React
Create card for pop up of create event	Randy Pere...	Oct 11 - 13	Create E... Sp...	Complete	React
Create Sport selector	Sung Kim	Oct 14 - 16	Create E... Sp...	Complete	React
RSVP Event buttons	Hyden	Nov 20	Improve... Sp...	Complete	
Complete custom styling for event based on interaction	Hyden	Nov 20	Improve... Sp...	Complete	
Create Event Enhancements	Randy Pere...	Nov 18	Improve... Sp...	Complete	
Finalize Event object schemas	Randy Pere...	Nov 18	Improve... Sp...	Complete	
Make events clickable	Hyden	Nov 18	Improve... Sp...	Complete	

Video (Optional)

VCS - Github

[CSCI-3308-CU-Boulder/Prancing-Purple-Ponies \(github.com\)](https://github.com/CSCI-3308-CU-Boulder/Prancing-Purple-Ponies)

Individual Contributions

Hyden Polikoff

- [Hyden's Commits](#)
- I was tasked with developing the front end assets for the main feed of our app. I worked on displaying the list of cards containing events as well as developing the buttons for the various RSVP options for events. I also implemented dynamic styling attributes for the buttons. Lastly on the development front, I built out the rough framework for our event details page. For the PM side, I got our team set with Asana and Slack integrations.

Sung Kim

- [Sung's Commits](#)
- Sung Contributions

Nathan Straub

- [Nathan's Commits](#)
- Nathan Contributions
- I predominantly worked on the front-end design of this project. The starting home page, sign up, login, profile, and edit profile pages are all designed and executed by myself, including various style and add-on features from react native, react native elements, and various other libraries. Alongside my frontend contributions, I created our team's architectural diagram and managed the folder holding all of our projects documentation.

Randy Perecman

- [Randy's Commits](#)
- Randy Contributions

Paolo Castro

- [Paolo's Commits](#)
- I worked primarily on the backend aspect of the app. Specifically, I worked on creating a wrapper for the Firebase API and writing the backend for the main feed page and one of the profile pages. This involved connecting to the Firestore database and using Firebase Authentication. I also often worked on fixing merge conflicts and merging everyone's code into a single working version of the app. Since I was one of the more experienced among the group, I also assisted teammates on several occasions on their tasks.

Thomas Deaner

- [Thomas' Commits](#)
- Thomas' contributions

Deployment

Visit [SkoPlay on Expo](#) to view the project. Scanning the code with an Android phone using the Expo app is ideal. If using an Android phone is not an option, then click "Open project in the browser" to connect to an online Android emulator. Once the emulator has loaded, you will be sent to a browser page inside the phone. On that page, scroll down (by dragging the mouse) and click on "Open with Expo". This online emulator is extremely slow, which is why using an actual android phone is preferred.