SoCalendar

Nick Chonko

Advised By Nan Niu

Tim Walsh

Background

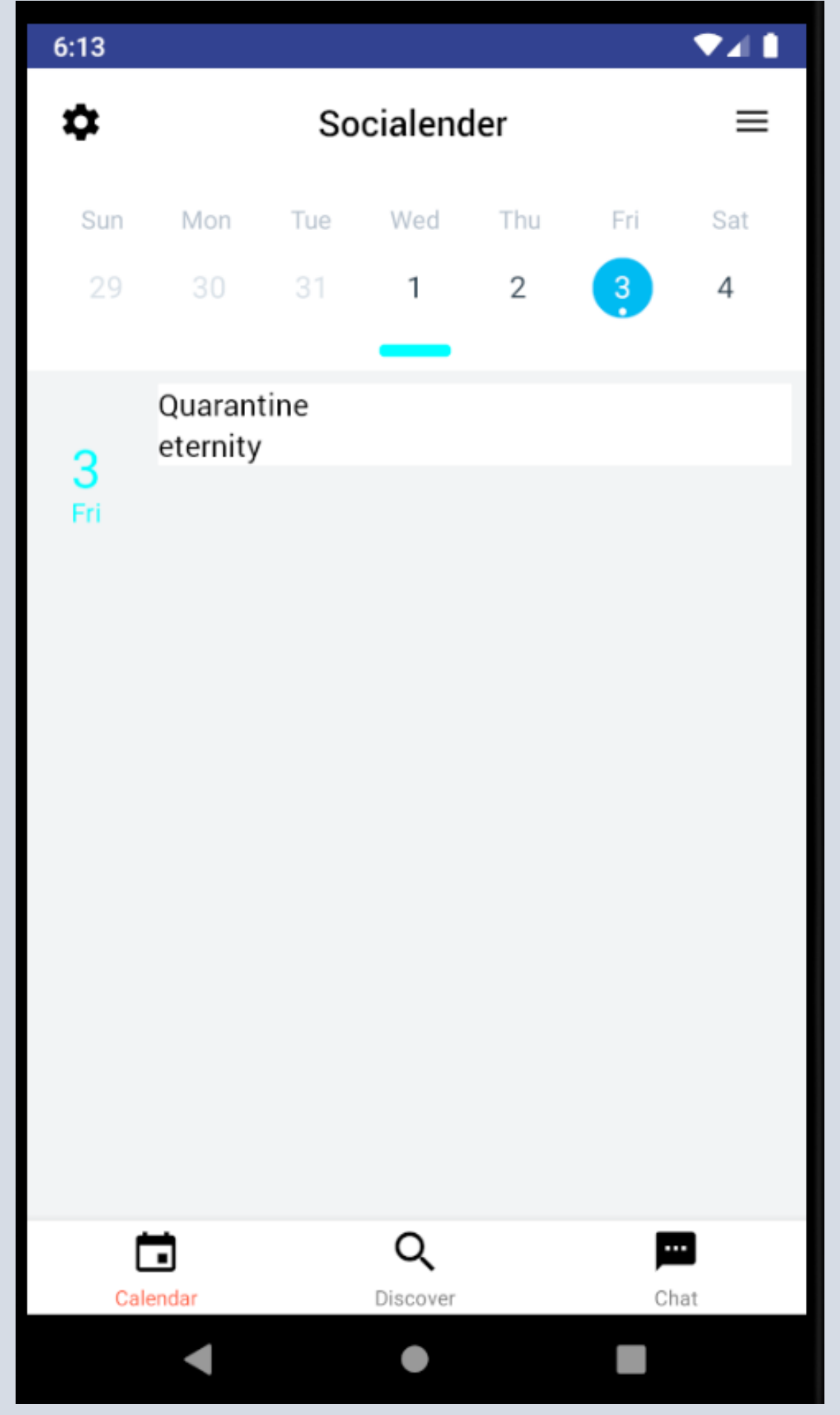
SoCalendar is a social media application based on a user's schedule. A user inputs his or her scheduled events and/or tasks. SoCalendar provides the user an interface to discover fun activities and add them to his or her schedule. The user can chat with friends within the application to easily make plans. SoCalendar will then schedule the user's upcoming tasks freeing the user's valuable time.

Goals

- Well maintained and tested code
- Intuitive UI and UX
- User sign up and login capabilities
- Working Calendar
- Automatic Event Scheduling
- Discover page with recommended events
- Messaging capabilities
- Cloud deployed backend
- Google Play Store front end application
- Improved planning, time management, architecture and development skills

Design

EASY USER-FRIENDLY DESIGN ALLOWS THE USER TO MAKE PLANS EASILY



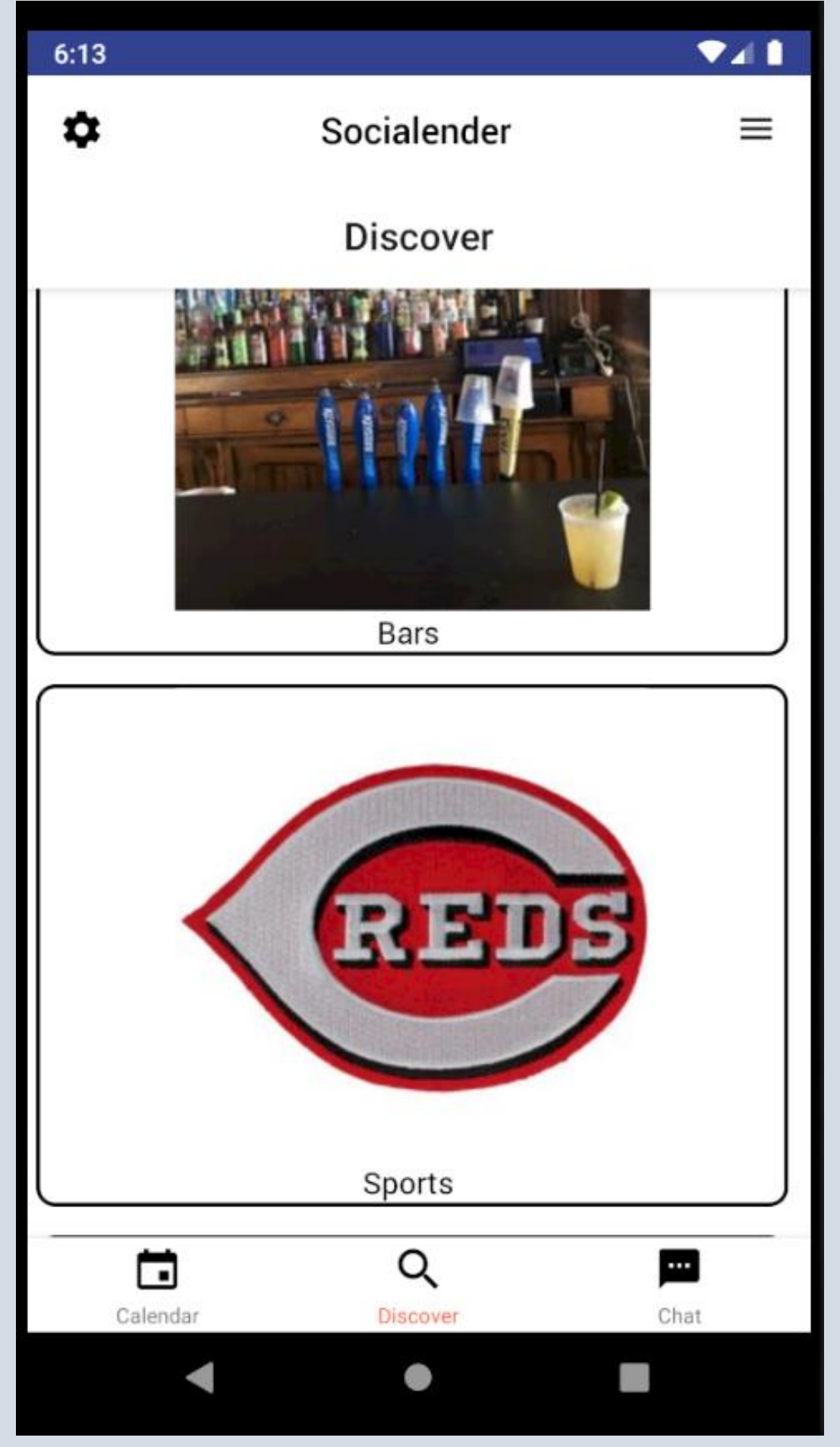
6:13

Socialender

Sun	Mon	Tue	Wed	Thu	Fri	Sat
29	30	31	1	2	3	4

3 Fri — Quarantine eternity

Plan and Schedule:
At its core, SoCalendar is designed to help a user organize his or her schedule. SoCalendar intuitively schedules events and tasks at the times a user needs to attend or complete them.

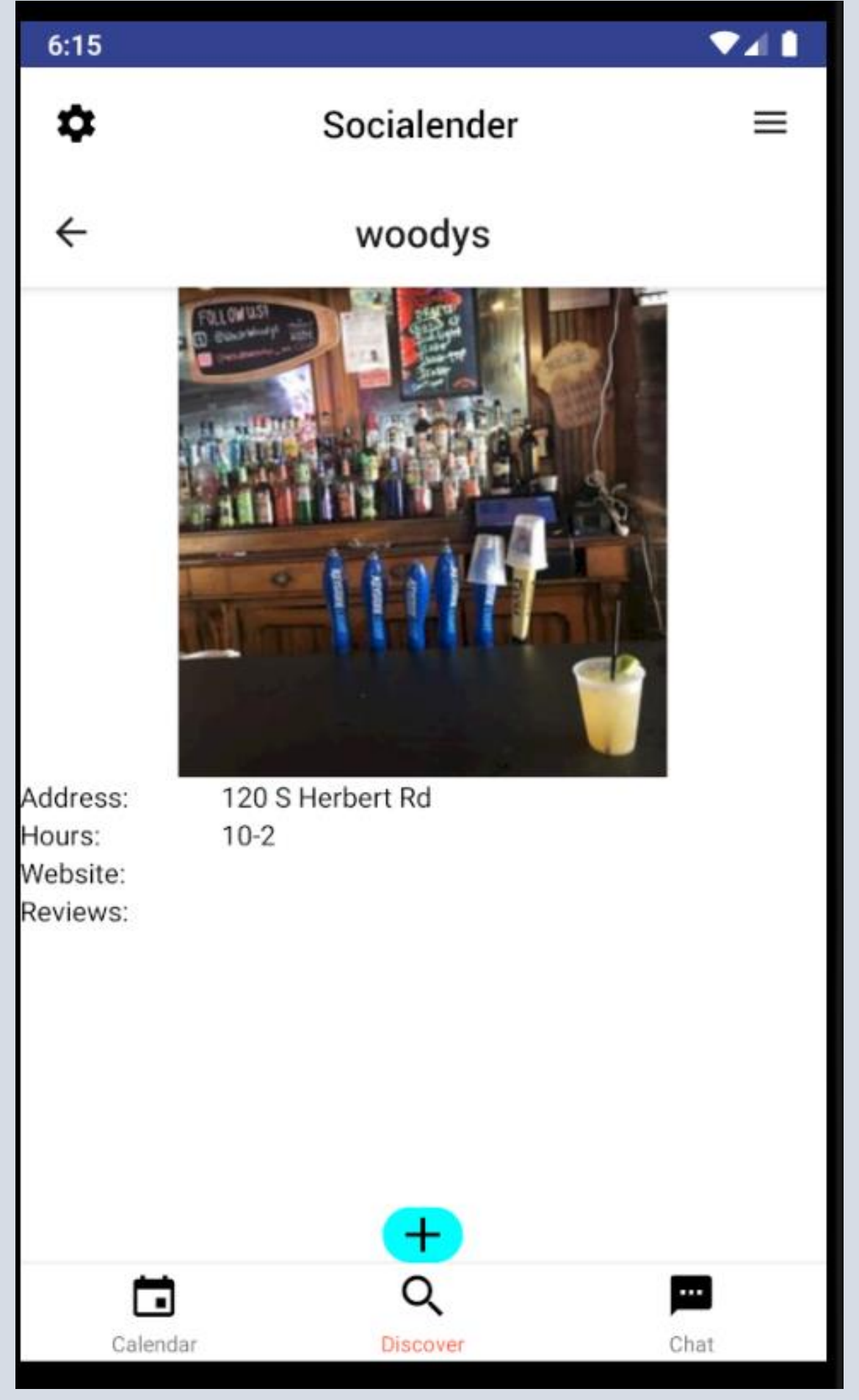


6:13

Socialender

Discover

Search and discover:
SoCalendar is designed to help a user maximize his or her schedule. Through the Discovery page, a user may simply browse things to do by category.



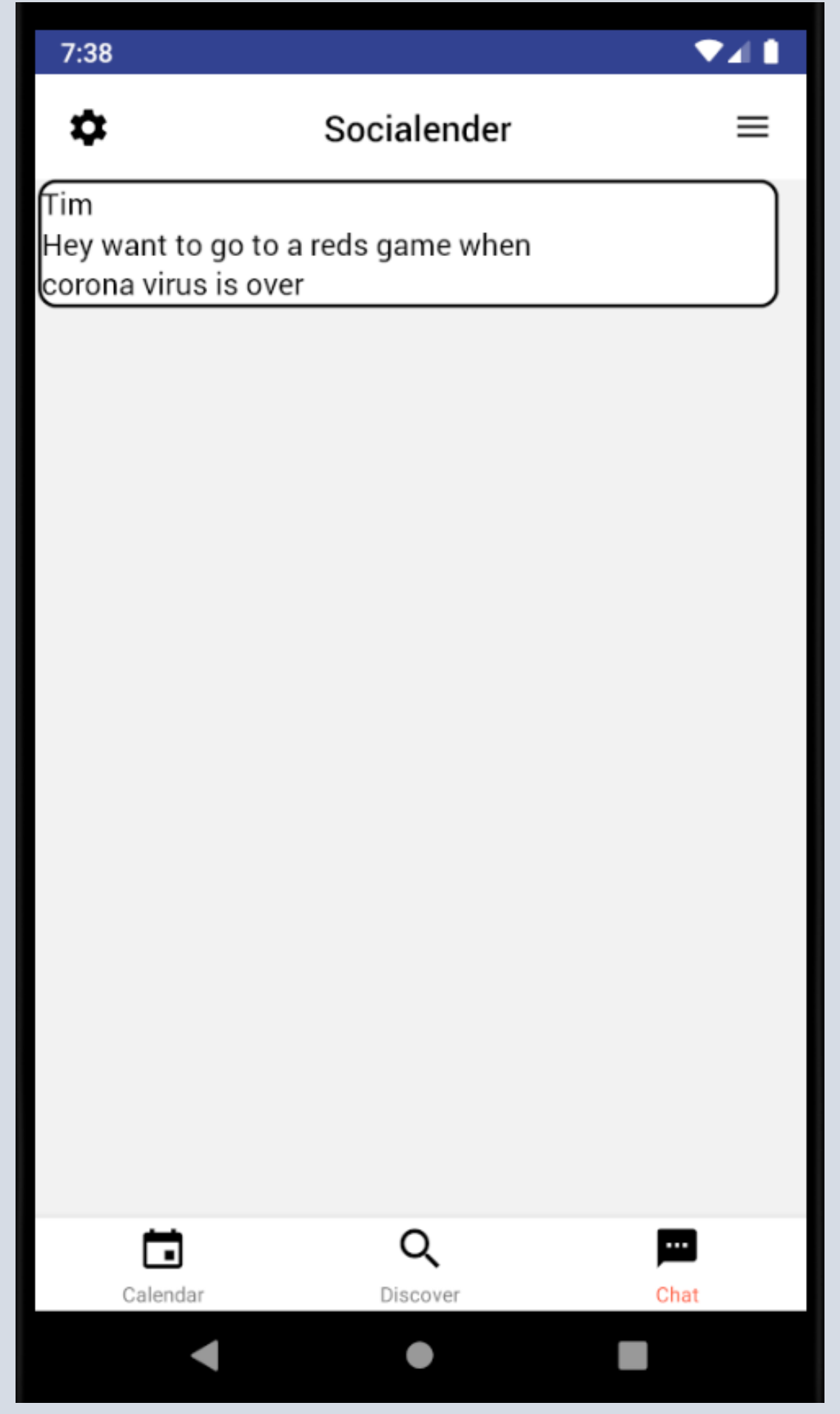
6:15

Socialender

woodys

Address: 120 S Herbert Rd
Hours: 10-2
Website:
Reviews:

Add it to the Calendar:
Worried there is no time to do something fun? Fear not! SoCalendar will rework a user's schedule to fit his or her needs.



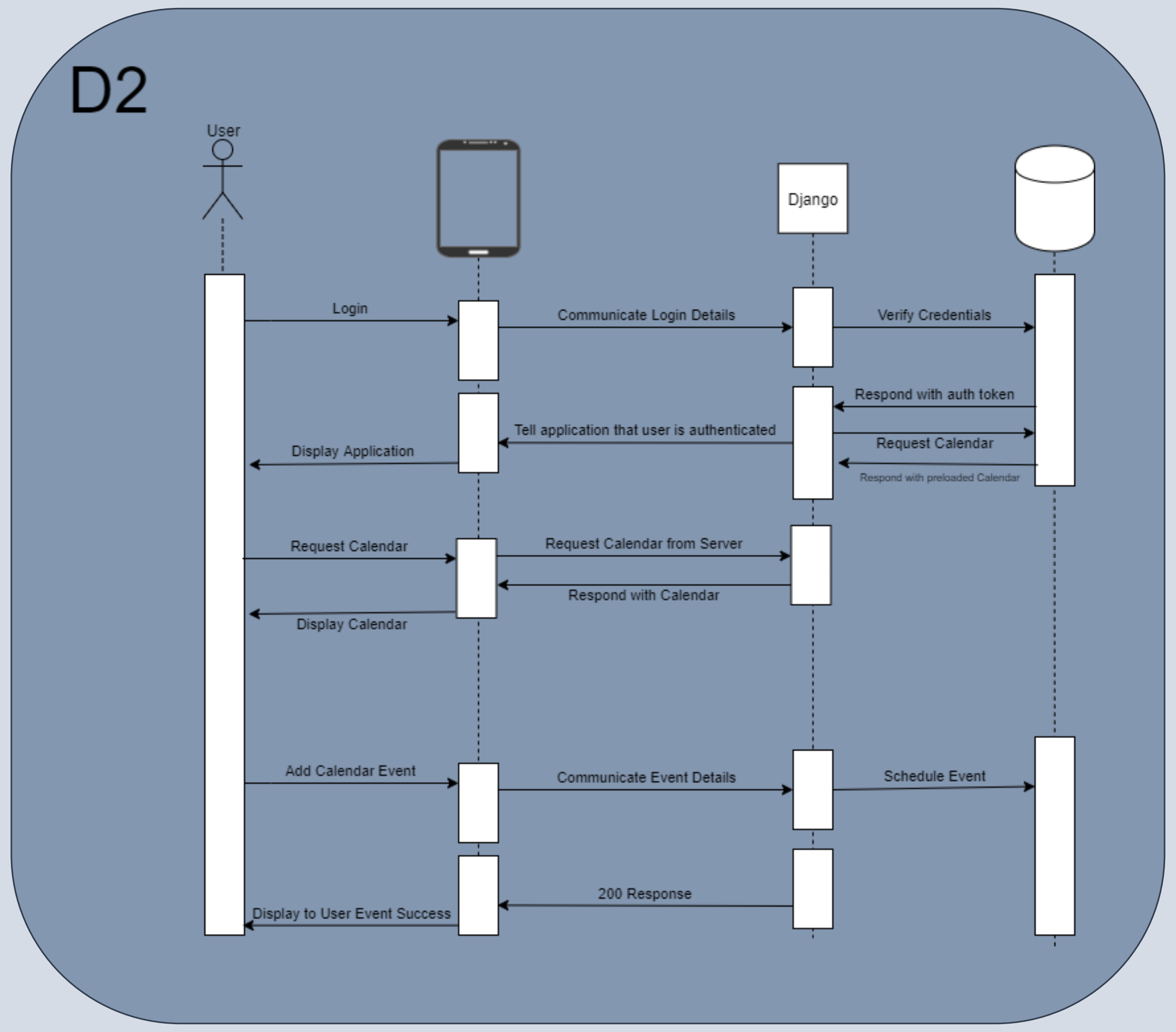
7:38

Socialender

Tim
Hey want to go to a reds game when corona virus is over

Squad Up:
Messaging capability within the app. Users have no excuses to miss out on fun with friends because SoCalendar will free up their calendars.

Our lowest level Design Diagram



Design Decisions

Frontend:
For the frontend we are using the React Native framework with several additional libraries in order to expand functionality. The main reason we are using React Native is that it supports Android and IOS platforms; this significantly reduces the development time necessary to support both platforms. We utilize react-native-calendars (an open source library developed by WIX) to complete the Calendar rendering quickly. React-navigation is the library available to support our navigation needs. We were able to quickly focus on the custom needs of our application, such as the discovery page, because of its pre-built functionality.

Backend:
For the backend, we are running a Django server with Django Rest Framework to implement our API. We use Celery to run some of our automated processes asynchronously. We use a PostgreSQL database to handle all of our data storage.
Our backend runs within Docker Containers which allow for easy deployment to AWS. We use pytest to test our backend to make sure that all of our functionality works properly. Finally, we use Flake8 to lint our code and to ensure that the entire backend follows the same quality standards.

Challenges

By far the biggest challenge for the developers is keeping testing up-to-date. Nothing is added into the code before thorough testing. Other commitments, such as school and homework, take time away from the project. Django Rest Framework makes writing the API easier, but it is a complex framework. Moving off campus created remote work across multiple time zones.

Achievements

- Developed a phone application and docker container
- Used React Native and Django to create a client-server architecture
- Utilized several open source libraries
- Researched many technologies and chose the most effective one to meet the constraints of our solution
- Worked as a team to effectively communicate and develop a solution on time