

Superfrog Scheduler

Designing a Simpler TCU Spirit Scheduling Experience

Anh Vu, Sarah Allen, Duncan Scott-Martinson, John Payton, Jose Lazalde

Faculty Advisors: Bingyang Wei, Ph.D (COSC)

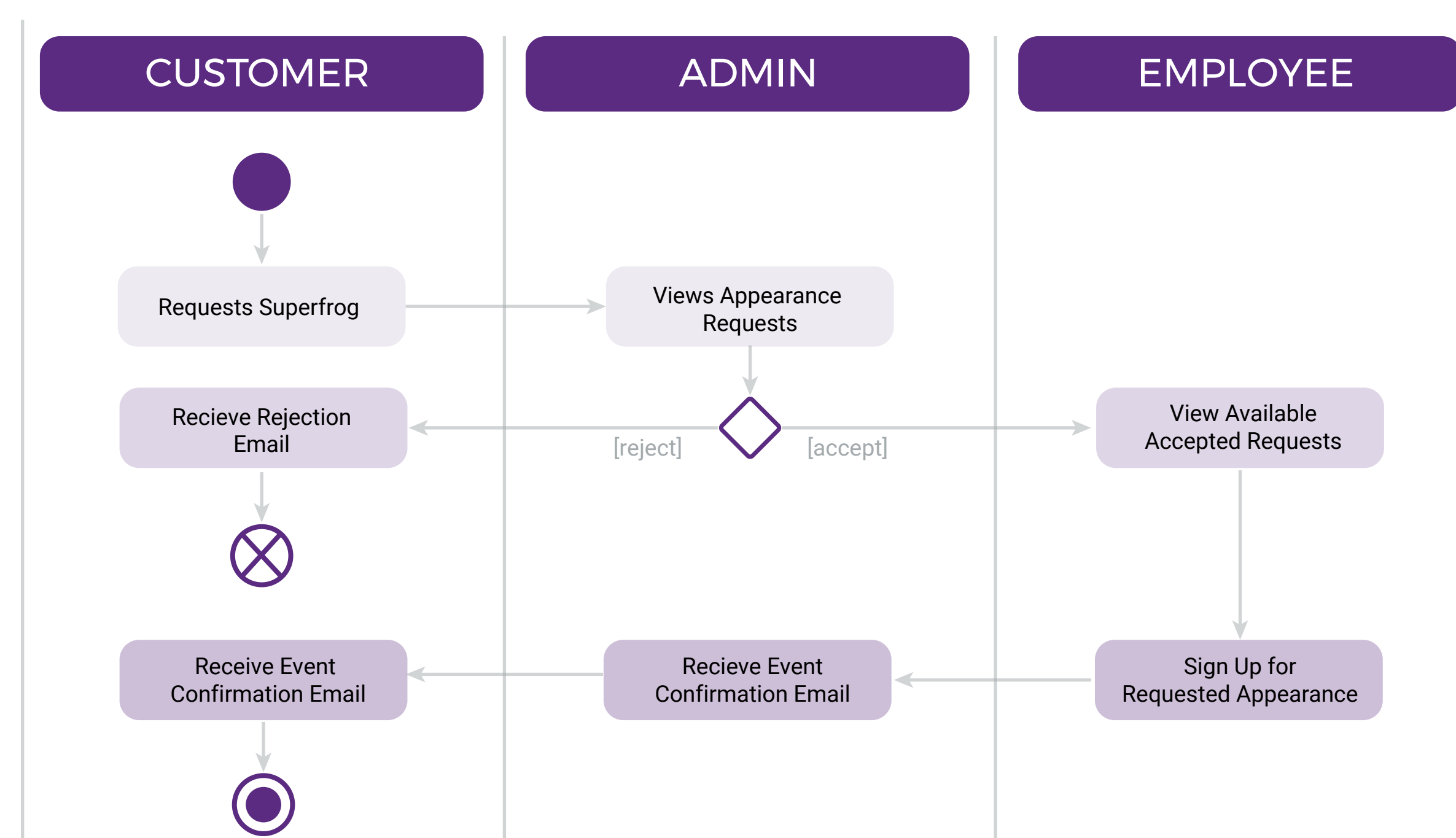
Background

TCU's mascot, Superfrog, is in high demand. There are over 100 requests for Superfrog to appear at an event every semester. All of these events are handled by one person, the TCU Spirit director, and involve a lot of paper, emails, and phone calls. The process of booking and scheduling an appearance at an event is difficult for the customer, the director, and the student employees who appear as Superfrog. Members from the TCU Spirit team approached the TCU Computer Science department in hopes of developing a simplified and automated process for scheduling and booking Superfrog appearances at requested events. Development of the Superfrog Scheduling web application began in Fall of 2018.

Goals

- 1 Create a website where customers can easily request a Superfrog appearance at their event
- 2 Give the TCU Spirit Director one place where she can accept or reject customer appearance requests
- 3 Allow the TCU Spirit Director the ability to generate payroll
- 4 Give Superfrog employees one place where they can sign up to appear at events and check their event schedules
- 5 Give Superfrog employees necessary event information at a glance

Work Flow



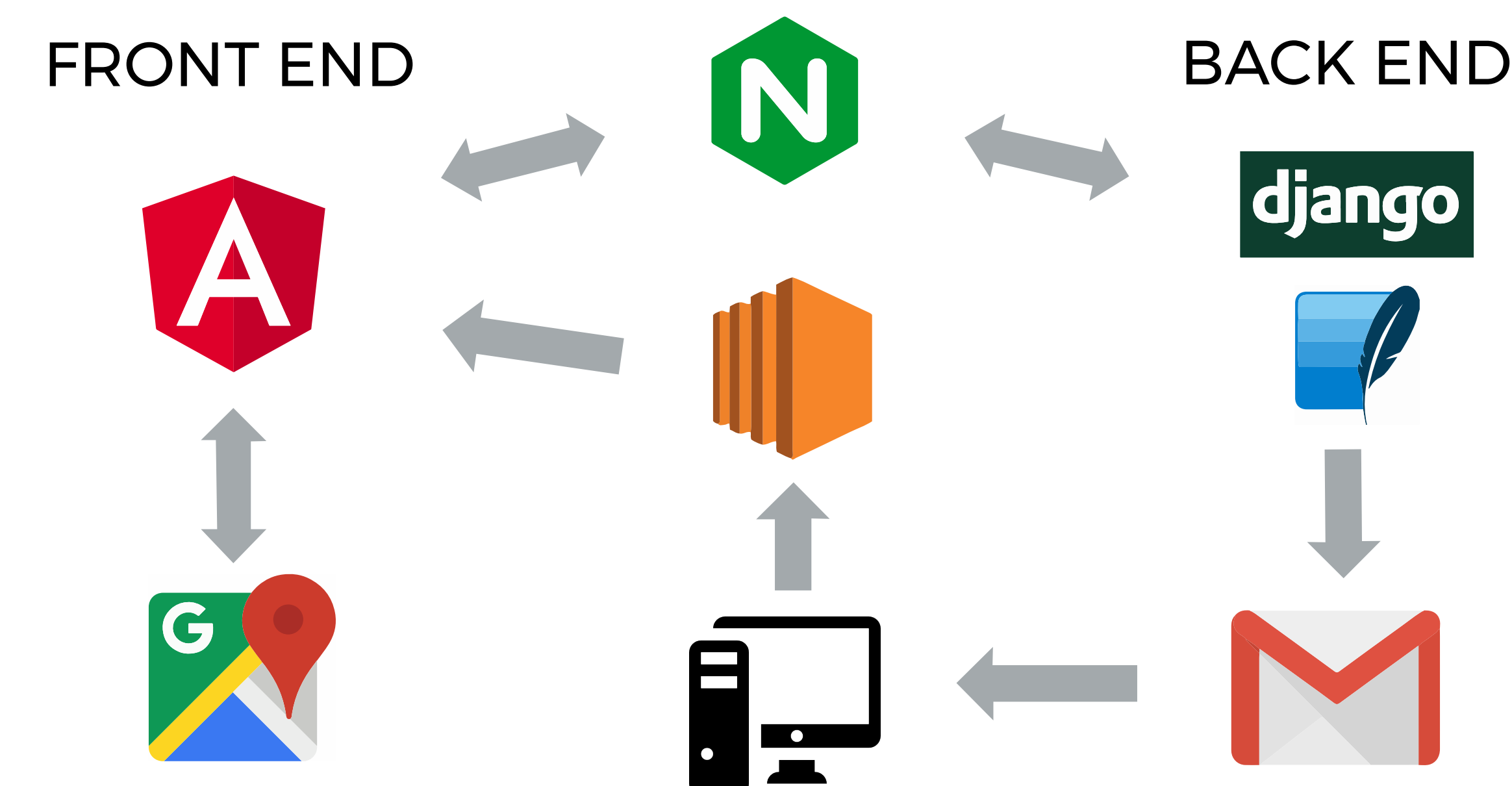
Once a Superfrog has agreed to attend an event and the customer has been notified that their event is booked, the Superfrog will contact the customer to discuss details about their appearance.

System Architecture

The website is hosted on a cloud server provided by Amazon Web Services. This server runs the front-end and back-end servers.

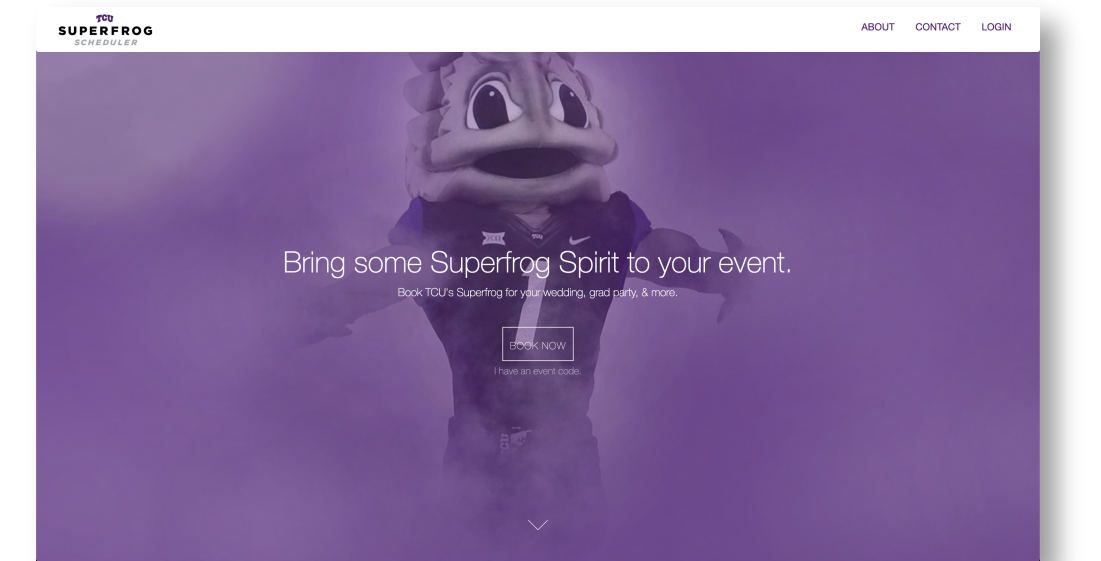
The front-end server provides interfaces for the user to interact with, and communicates with the back-end server to manipulate and display useful data. It also communicates with Google Maps to get information about appearance locations.

The backend server handles data requests from the front-end server and maintains the database. It also communicates with Gmail in order to send emails to users.

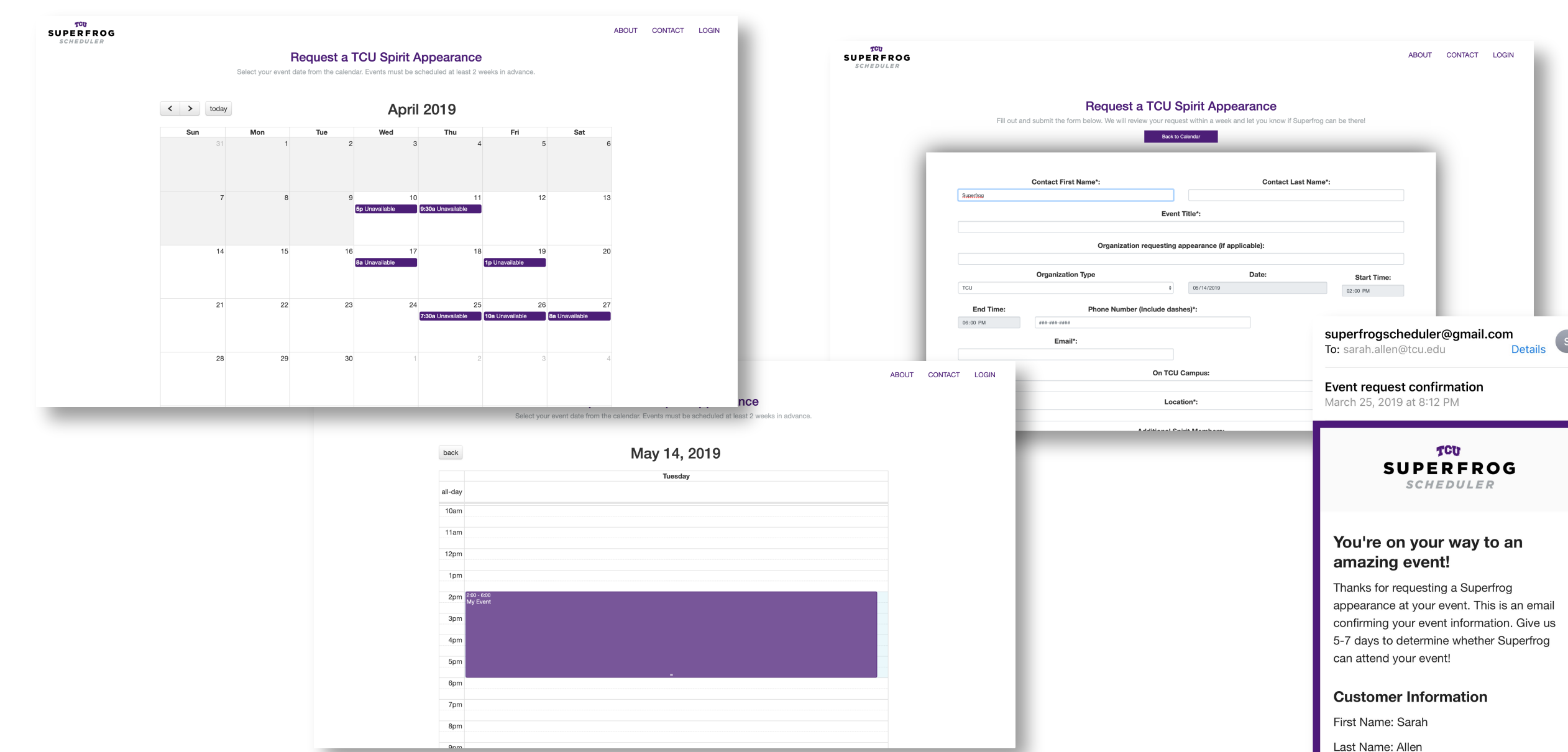


Solution

Our solution is a single web application that serves the needs of the customer, admin, and employee. The customer uses the website to request a Superfrog appearance by submitting an electronic form. The administrator can log into the same application to view customer requests and either approve these requests or reject them. Employees can also login to the application to view and sign up for new events that have been approved by the admin, or to view the details of events that they have already signed up for. Emails are sent to customers, employees, and admin throughout the process as clear and consistent communication.



Customer View



User interface is straightforward and easy to navigate.

Customer can request an event with little confusion and time.

Availability and terms of appearances are communicated clearly.

Request is confirmed through email.

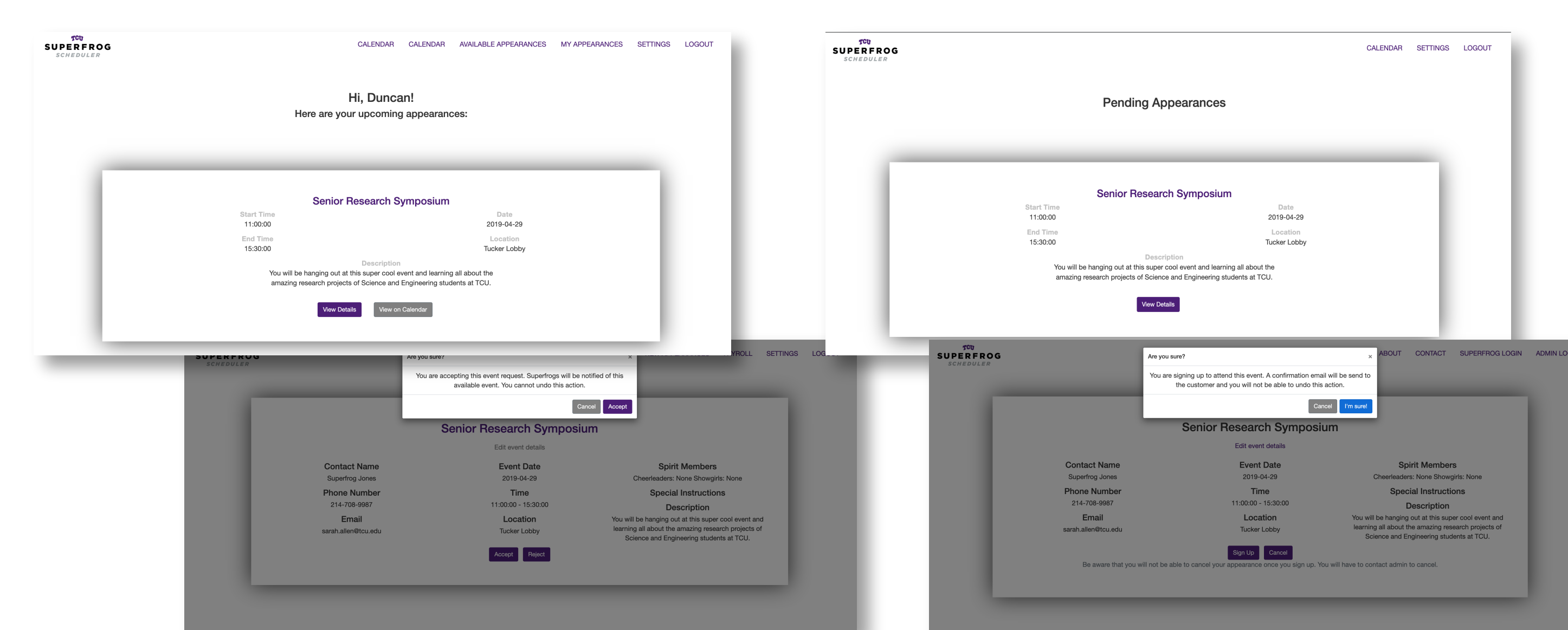
Application Reach

The purpose of this application was to make the lives of customers and TCU employees easier. By removing nearly all of the paper and various technologies that were originally weighing down the process of booking and scheduling a TCU Spirit appearance, the Superfrog Scheduler will hopefully increase community engagement with Superfrog they feel more comfortable navigating an appearance request, and allow the spirit program to allocate resources to more meaningful and productive areas.

References

Bootstrap Documentation: <https://getbootstrap.com/>
Angular Documentation: <https://angular.io/docs>
Django Documentation: <https://www.djangoproject.com/>
FullCalendar Library: <https://fullcalendar.io/docs#toc>
Google Maps API: <https://developers.google.com/maps/documentation/>
Gmail API: <https://developers.google.com/gmail/api/>
SQL Lite Documentation: <https://www.sqlite.org/index.html>
Nginx: <https://www.nginx.com/>

Admin & Employee View



Event requests are clear to the admin and include all necessary details.

Admin can approve or reject request with one click.

Upcoming events and relevant information clear to both admin and employees.

Communication of schedules is clear, not confusing and scattered.

Acknowledgements

Thanks to TCU Spirit for entrusting us with this task and believing in the end product. We would like to thank Dr. Bingyang Wei for being such an encouraging professor throughout this year and for teaching many valuable lessons about what it really means to be on a software engineering team. We would also like to thank the CS faculty as a whole for supporting us and guiding us as we learned through these past four years.