Title:

ScheduleMe Launch Report

Contributors:

- Yuxiang Wang
  - GitHub: yuwa1155 | Email: yuwa1155@colorado.edu
- Nasen Alm
  - GitHub: nasenalmCUBoulder | Email: naal6332@colorado.edu
- Prajinkumar Suresh
  - GitHub: psureshkannan2 | Email: prsu4616@colorado.edu
- Alexander Schwarz
  - GitHub: A1exanderSchwarz | Email: alsc3632@colorado.edu
- Elkan Bruha
  - GitHub: elkanbruha | Email: elbr5153@colorado.edu
- Lucas Williamson
  - GitHub: luwi1899 | Email: luwi1899@colorado.edu

Project Description:

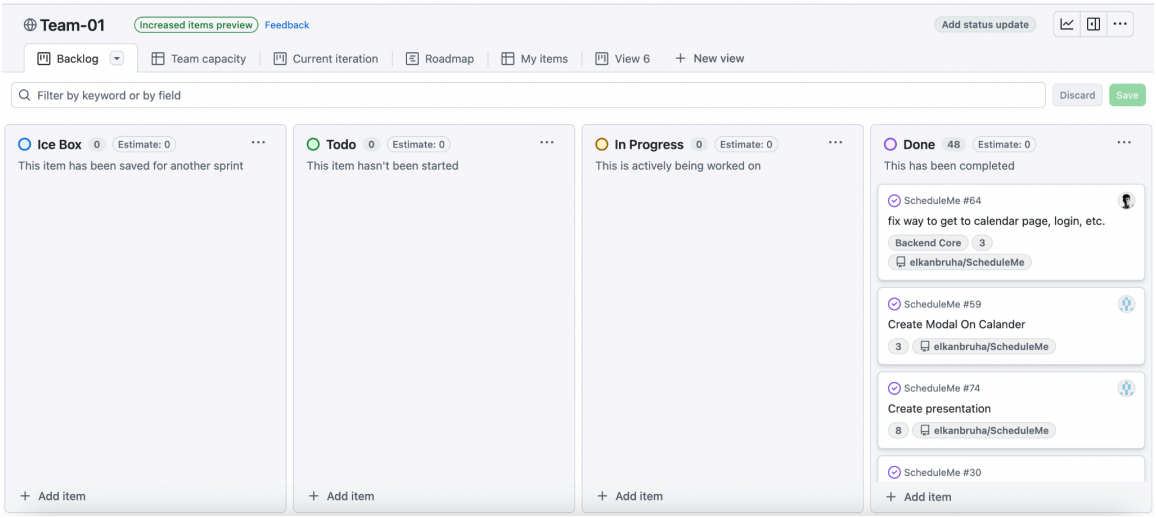
ScheduleMe is a smart calendar platform designed to connect customers with service providers, eliminating the usual back-and-forth of appointment scheduling. While our primary target market is landscaping businesses, we can help any service-based company. ScheduleMe makes it simple to secure mutually convenient time slots — without the hassle of overbooking or missed connections.

Our platform offers several key features to enhance the scheduling experience:

- **Calendar Integration:** Easily export appointments to iCalendar, so you can stay organized using tools you're already comfortable with.
- **Built-in Video Calls:** Schedule and conduct consultation appointments directly within ScheduleMe powered by Jitsi, helping customers and service providers connect face-to-face before services are confirmed.
- **Time Zone Support:** Book and manage appointments confidently across different time zones, ensuring smooth communication for both local and remote customers.

These features are built on top of a robust core platform that allows users to securely log in, schedule new appointments, edit existing ones, and cancel if needed. ScheduleMe transforms scheduling from a frustrating chore into a streamlined, intuitive experience for both businesses and their customers.

Project Tracker - GitHub project board:
https://github.com/users/elkanbruha/projects/2



Video:
https://drive.google.com/file/d/11FgboCYX06WfwWwHwPJinBl0bSJnPJLK/view?usp=sharing

VCS: https://github.com/elkanbruha/ScheduleMe.git

Contributions:

Nasen Alm: I started working on the backend, helping form the overall project structure, setting up express.js, handlebars, and some of the very basic endpoints. I worked on routing/rendering the pages that the people working on the frontend had created. I then created the first docker-compose file, and setup .gitignore and a basic .env file. Most of my contribution was later, individually building the iCalendar download feature, that let people save events created in the app to their apple calendars.

Prajinkumar Suresh: I contributed to the initial developments of the initial API endpoints to create basic functionality for our website. Additionally, I helped

create the original file structure with some of the basic html. I then started working on the implementation of our video call feature with the Jitsi API. I had to edit both the front end and the back end to implement our video call feature. Finally I was in charge of creating the bureaucratic items for our project. This includes creating the user acceptance testing plan, weekly team meeting notes, and the weekly release notes.

Alexander Schwarz: I initially created the file structure for the repository. I focused on user experience while using our website through frontend programming. I created the navigation bar and body outline. I designed and programmed our login/register pages form passing for backend to be able to input/pull user details into/from our database and made them simple/ sleek using CSS and javascript I created in our styles file to make sure it was perfect. I continued to modify, refine, and generally fix the CSS file as I worked on all other pages from the calendar to the about us pages.

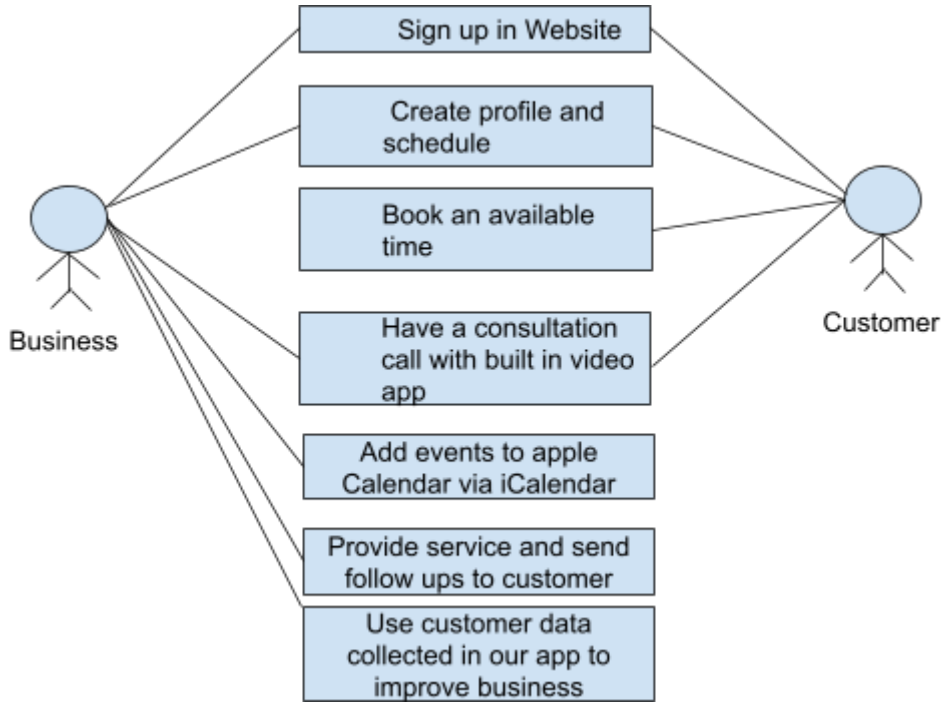
Elkan Bruha: I focused on both the user-facing experience and the underlying infrastructure. On the front end, I designed and refined our global CSS system to ensure visual consistency, responsiveness, and accessibility across the application. I built the Home Page from wireframe to production, then engineered the Calendar Page, integrating event rendering, and real-time refreshes. To support international users, I added a Timezone Manager module that automatically converts and displays events in each user's local time. On the back end, I established reliable connections between the React interface and our RESTful services, created secure login and sign-out APIs with token-based authentication, and fine-tuned database indexes and queries to reduce latency under load.

Lucas Williamson: For this project I worked on the front-end development as well as a bit of the back end. For the front end I created the entire help page with FAQ's and just general information for the user if they ran into issues using our website. I also worked on helped work on setting up the calendar page and the modal for creating an appointment. I also helped troubleshoot some of our problems with the database connection on the backend that definitely set us back. Finally I did the presentation that we presented in class and I also did the video showcasing our final product and how it works.

Yuxiang Wang: As a team member I have done more of the backend code and testing, I have done the creation of the database where users can store their account information and all the changes made after logging in, i.e. creating

accounts and creating calendars, I have also helped the team with the testing side of things, where the team has encountered things such as: the server is not working, the page is not displaying after logging in. I also used render to help the group set up the platform on the web to run the code and project better.

Use Case Diagram:



Wireframes:

ScheduleMe Wireframe

https://docs.google.com/presentation/d/1x00q0VinHrkross5NUkZbjtLCMbQxFvlzCj3xkUpXbo/edit#slide=id.g33894b9b325_0_78

Test results:

We Tested our code through a variety of methods including regression testing, automated testing through mocha, and unit testing. We also used survey testing to test our features.

Test through Unit Testing:

As a team we conducted unit testing whenever we introduced new features. We made sure to do this by making sure that everyone who contributed to the

development of that feature tested and made sure that it worked. After that we made sure to make the entire team test out the feature and give feedback.

Testing through Regression Testing:
We used regression testing whenever we built upon a feature so that nothing that was working before broke. By doing this we were able to ensure that we never committed a broken feature.

Testing through Surveys:
We worked with many people in our community to get their input on the app, so that we make changes to features to make them better and fix any bugs that they accidentally ran into while trying to learn the product.

test through server.spec.js and mocha

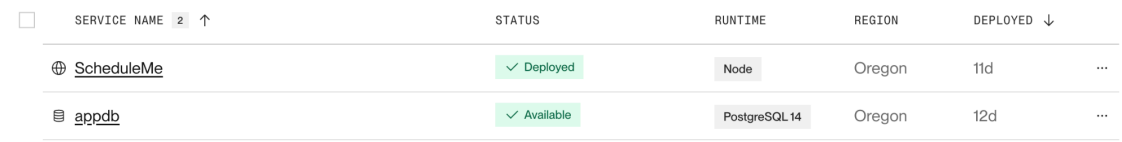
Testing Register API
- ✓ Positive: should register a new user
- ✓ Negative: should return 400 when required fields are missing
- ✓ Negative: should return 500 when email already exists

Testing Login API
- ✓ Positive: should login successfully with correct credentials
- ✓ Negative: should fail login with incorrect password

Testing Appointments API
- ✓ Positive: should create a new appointment
- ✓ Negative: should fail to create appointment with invalid credentials
- ✓ Positive: should get user appointments
- ✓ Positive: should get business appointments
- ✓ Negative: should fail to get appointments with invalid type

10 passing (5s)

Deployment: use docker for the deployment, within docker open cd into ScheduleMe and ProjectSourceCode then use docker compose up,then access http://localhost:3000/register to register account.



SERVICE NAME 2	STATUS	RUNTIME	REGION	DEPLOYED
ScheduleMe	Deployed	Node	Oregon	11d
appdb	Available	PostgreSQL 14	Oregon	12d

https://scheduleme-v2c3.onrender.com

And we can also use render to directly sign up and login and use our app!