Team Rana Scheduler Application Final Report

Nick Benoit - Eamon Oldridge - Nathan Rosenberg - Yil Verdeja

Purpose

Throughout this project our group tried to use the different skills of our team members as effectively as possible. In this section, we will cover the roles that each team member occupied.

Nick Benoit	Organizer, Stylist
Eamon Oldridge	Front End Developer
Nathan Rosenberg	Backend Lead
Yil Verdeja	Front End Coordinator

Introduction

The goal of this project was to produce a simple scheduler application that could enable an organizer to create a schedule that they could share with participants. During the creation of the schedule, organizers could set the parameters for the schedule, such as the start and end times, start and end dates, and duration of the meetings. Once the schedule was created, the organizer could open and close individual time slots, or change the availability of an entire day or specific time. Participants could then schedule meetings with the organizer in open time slots, or cancel a meeting they had already scheduled. A system administrator would be able to review recent activity in the system and delete old schedules.

Our solution is the Rana Scheduler Application. It contains all of the features required in the project description, and is simple and easy to use. Organizers can create schedules and share them using a URL, which allows participants to easily access the schedule to add meetings. Our clean interface allows organizers and participants alike to clearly see which time slots are available or unavailable, and scheduling or canceling meetings is intuitive. The organizer can modify the parameters of the schedule using the review mode, which provides flexibility if the schedule needs to change. Our solution represents the best effort of our team members to provide an elegant solution to the scheduler problem.

Team Organization, Members, and Responsibilities

Our team was successful because we were able to leverage each member's skills effectively. Part of our success can be attributed to our meetings, which were held frequently, and allowed everyone to stay on the same page. We scheduled two meetings a week for the first half of the project, and for the second half of the project we met on a more frequent, but ad hoc basis.

Throughout this project our team never designated a leader, but as challenges arose our team members rose to meet them. Early on Nick was useful as an organizer by making sure that meetings were scheduled, everyone attended, and they proceeded efficiently. Nathan volunteered to do the backend of the application on AWS, and Yil was the captain of the front end. Eamon was a developer for the front end and main tester of functionality. The main roles and responsibilities are summarized in the table below.

Team Member	Role	Responsibilities
Nick Benoit	Organizer, Stylist	● Organize and facilitate meetings ● Write group reports ● Use Case documentation ● UI Mockup ● Add styles (CSS) to webpages ● Method for parsing JSON objects for calendar data and creating a calendar table from it
Eamon Oldridge	Front End Developer	● Methods for searching timeslots and creating tables from them ● Finalizing Use Case design ● Methods for cancelling meetings for both organizer and participant ● Testing and identification of functionality errors
Nathan Rosenberg	Backend Lead	● Created Github repositories ● Everything with AWS ● Class diagrams ● All test cases ● All backend code
Yil Verdeja	Front End Coordinator	● UI Mockup ● HTML layout of web page ● Implementation of JavaScript methods to create, update, and modify schedules

		● Implementation of JavaScript methods to create and cancel meetings ● Implementation of JavaScript sysadmin functions

Process

Our original process was to hold regular group meetings, two times a week, where we would meet to brainstorm ideas and assign tasks for individual group members to complete. This was effective during the initial stage of the project when we were describing use cases and creating user stories. During our meetings we could collaborate on our ideas and then separately work on solutions before presenting them at the next group meeting.

This method was used for creating use cases, designing user stories, mocking up the interface, and creating UML diagrams. Our group would decide on a course of action and assign tasks to individual members to be completed by the next meeting. At the next meeting, individuals would present their work to the group for critique, which would allow us to improve our work, before finally submitting.

During the later weeks of the project, our team shifted from regularly scheduled meetings to more frequent, but ad hoc meetings. Our group became quite specialized, with one member in charge of the back end, one member in charge of the front end, and the other two in charge of assisting them. This meeting structure evolved as a response to the changing group dynamic, and was largely successful at keeping group members on the same page. We also made frequent use of a group chat, to allow members to communicate issues they were having or to announce progress.

Individual tasks were still assigned to different group members, but as issues were encountered tasks might be switched between members with different strengths. This flexible structure allowed our team to overcome obstacles that might have otherwise been debilitating, by utilizing the strengths of each of our team members where they could be most useful.

When our team was writing code for our application we used two Github repositories, one for the front end and one for the backend, to keep our code organized and manageable. For new features we would use branches to add and test functionality, and once the new feature was working, merge the branch with the master. When new functionality was added to the application, our group would attempt to break it in order to expose flaws or weaknesses in our application, which led to a more robust solution after the issues were resolved.

The front end of our application was written in HTML, JavaScript, and CSS; the backend of the application was written in Python. For testing our code we employed PyTest, a framework for testing Python applications. We were able to achieve 80% code coverage using PyTest.

Tools

Our team made use of multiple tools over the course of our project to increase our productivity. For developing Use Cases and group reports we used Google Docs to collaborate on writing and editing. The mockups for our user interface were done in Google Drawings and Draw.io. We found these applications useful because it allowed us to store our documents in the cloud where all team members could access them, as well as work together to create and edit material at the same time.

Our team used Github to as our central cloud repository for code. We created two repositories, one for the front end and one for the backend. Github gave us source and version control capabilities that were easy to use, which kept our code organized and up to date. For each new piece of functionality, we created a new branch, which kept our master code working at all times and allowed us to expand the application’s capabilities in manageable pieces.

Zappa, a serverless Python web service, was used to connect to the Amazon Web Service API Gateway and deploy our backend to AWS Lambda. It helped to reduce the time to deploy code updates.

To test the backend of our application we used the PyTest framework with Coverage.py for code coverage. Using the framework we were able to achieve 80% coverage of our code. To achieve this level of code overage we were able to simulate a client with the Flask testing framework and test each of the routes.

Accomplishments

The following table summarizes the Use Cases that were created for this project based on the project definition, and their status as of the final submission.

Use Case	**Status**
Create Meeting Schedule	Completed
Select Dates	Completed
Select Time Slot Duration	Completed
Set Start Time	Completed
Set End Time	Completed

Name Schedule	Completed
Toggle Individual Slot	Completed
Toggle Day's Slots	Completed
Toggle Time's Slots	Completed
Delete Schedule	Completed
Review Schedule	Completed
Change Review Week	Completed
Share Schedule	Completed
Cancel Meeting	Completed
Create Meeting	Completed
Search Timeslots	Completed
Extend Start Date	Completed
Extend End Date	Completed
Retrieve Schedule List	Completed
Delete List of Schedules	Completed

Deliverables

Rana Scheduler Application	Allow Organizers to create custom schedules that Participants can add meetings to. Organizers can manage created schedules or modify their parameters. System Administrators can review system usage and delete old schedules.
Rana Scheduler Application README	A User's Guide to inform individuals on the capabilities and correct usage of the application.

Reflection

Overall, our team worked well together and this project was successful. Our team had two team members that were more proficient than the other two, which led to an imbalance in the amount of code contributed by each team member to the final application. Better communication between group members could have helped rebalance the load, and enable everyone to contribute more to the project.

Our biggest mistake was not communicating as effectively as we could have to each other about our progress towards individual components of the group project. If we were to do this project again, we would have continued to hold regular meetings throughout the entirety of the project, and formally assign tasks to group members with definite dates that they were expected to be completed. This would have allowed everyone to work on different components of the system and be accountable for their portions of the project.

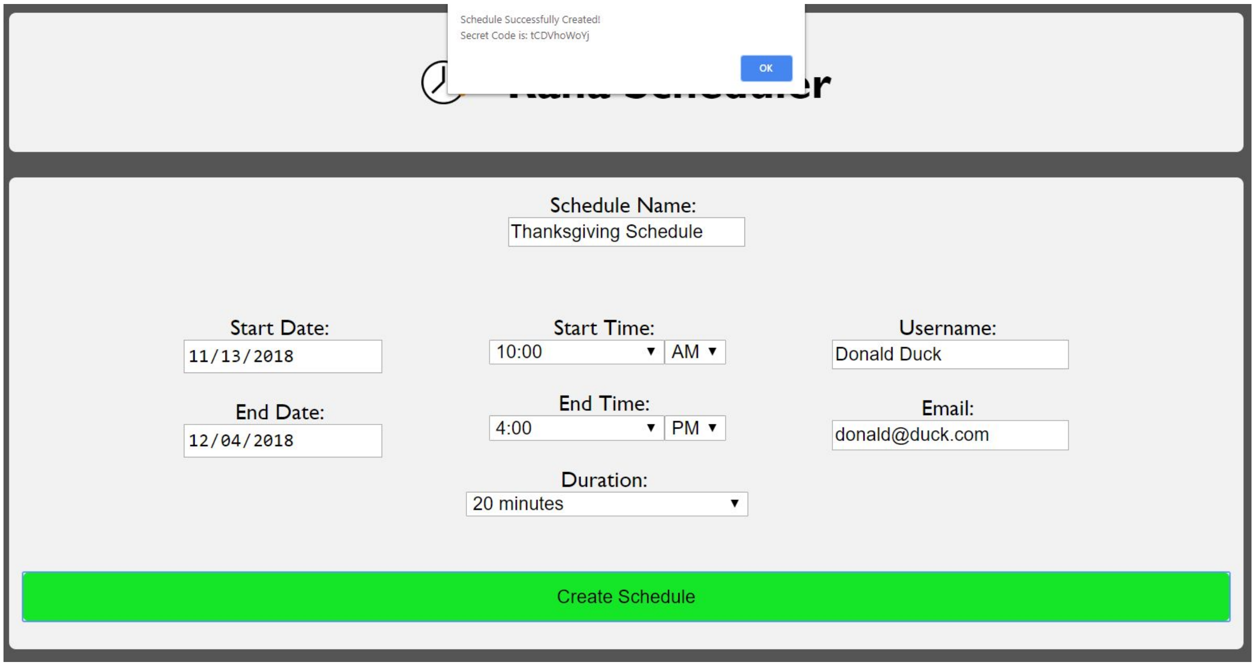
This project was a useful learning experience in developing a full stack application from start to finish. Planning the use cases and user stories at the beginning at the project and using this as a guide as the project went on made it easy to develop the application in small pieces. Mapping use case functionality to the UI allowed us to find missing or superfluous details of our design. By creating a solid plan it was easy to see where features would develop in the future, so there was minimal refactoring of old code to make it compatible with new functionality.

If we needed to give advice to a future team, we would want them to:

- Plan! Plan early and make sure everyone understands the plan. This will make your life a lot easier at the end of the project.
- Find the strengths of each group member and use them as effectively as possible. Share the work between group members with dissimilar skills so they can grow and develop as software engineers.
- Create a schedule and stick to it. If you are having issues, talk to your group members or the course staff.

Screenshots

Create Schedule



Schedule Successfully Created!
Secret Code is: tCDVhoWoYj

Schedule Name: Thanksgiving Schedule

Start Date: 11/13/2018

End Date: 12/04/2018

Start Time: 10:00 ▾ AM ▾

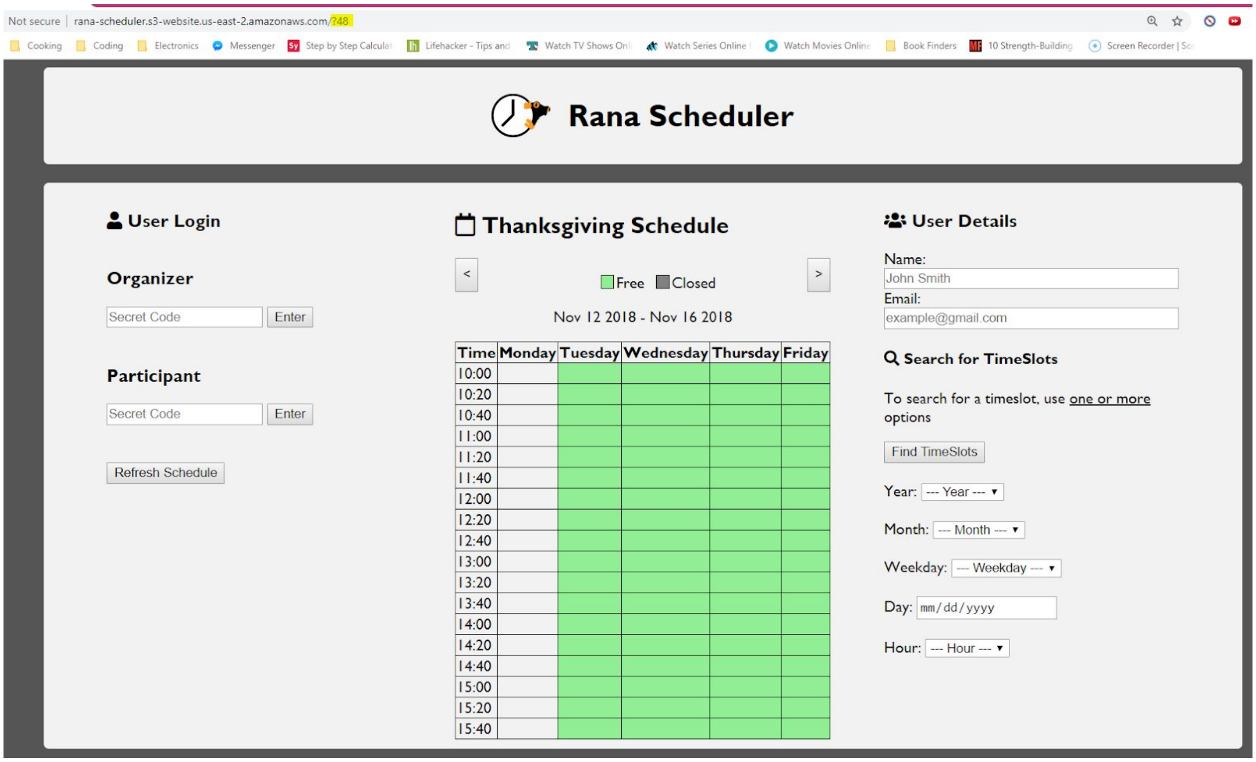
End Time: 4:00 ▾ PM ▾

Duration: 20 minutes ▾

Username: Donald Duck

Email: donald@duck.com

Review Schedule



Not secure | rana-scheduler.s3-website.us-east-2.amazonaws.com/?48

User Login

Organizer

Secret Code [Enter]

Participant

Secret Code [Enter]

Thanksgiving Schedule

< 🟩Free ▨Closed >

Nov 12 2018 - Nov 16 2018

Time	Monday	Tuesday	Wednesday	Thursday	Friday
10:00		🟩	🟩	🟩	🟩
10:20		🟩	🟩	🟩	🟩
10:40		🟩	🟩	🟩	🟩
11:00		🟩	🟩	🟩	🟩
11:20		🟩	🟩	🟩	🟩
11:40		🟩	🟩	🟩	🟩
12:00		🟩	🟩	🟩	🟩
12:20		🟩	🟩	🟩	🟩
12:40		🟩	🟩	🟩	🟩
13:00		🟩	🟩	🟩	🟩
13:20		🟩	🟩	🟩	🟩
13:40		🟩	🟩	🟩	🟩
14:00		🟩	🟩	🟩	🟩
14:20		🟩	🟩	🟩	🟩
14:40		🟩	🟩	🟩	🟩
15:00		🟩	🟩	🟩	🟩
15:20		🟩	🟩	🟩	🟩
15:40		🟩	🟩	🟩	🟩

User Details

Name: John Smith

Email: example@gmail.com

Search for TimeSlots

To search for a timeslot, use one or more options

Year: --- Year --- ▾

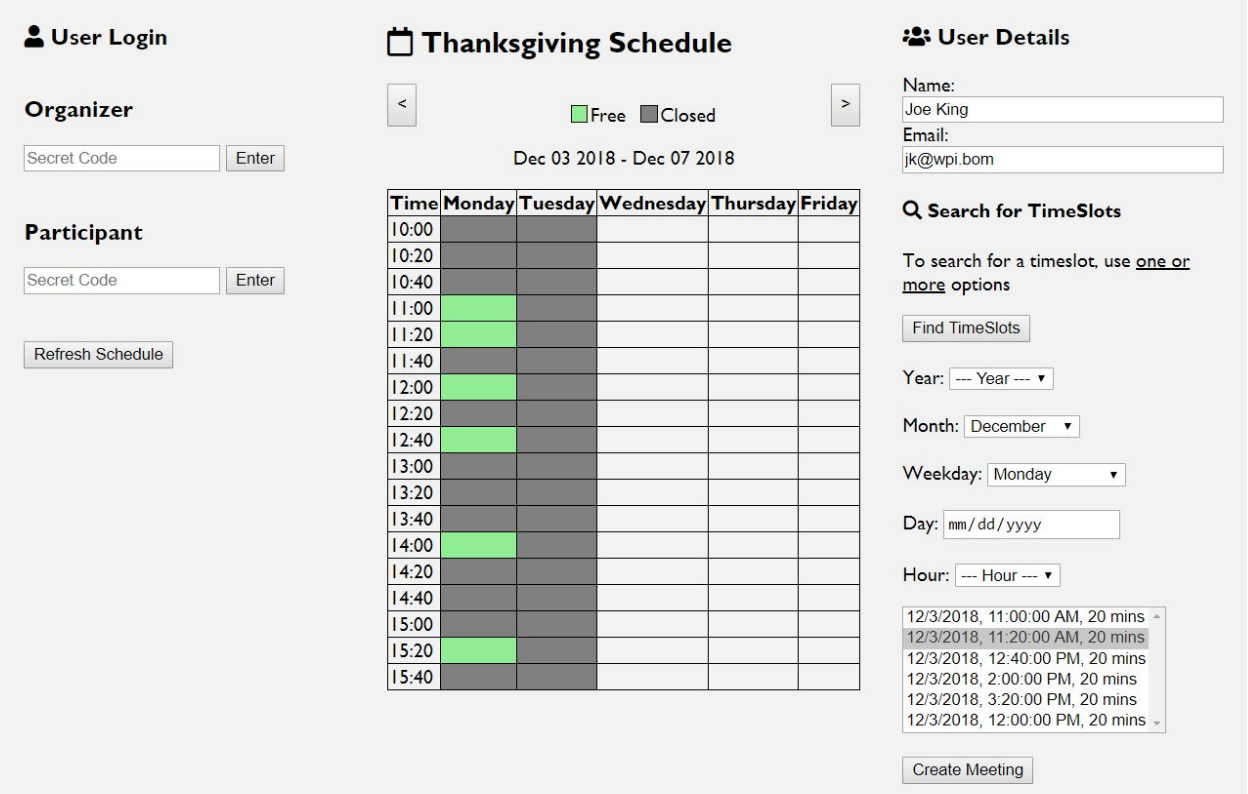
Month: --- Month --- ▾

Weekday: --- Weekday --- ▾

Day: mm/dd/yyyy

Hour: --- Hour --- ▾

Participant Meeting



User Login

Organizer

Secret Code [Enter]

Participant

Secret Code [Enter]

[Refresh Schedule]

Thanksgiving Schedule

[<] 🟩Free ⬛Closed [>]

Dec 03 2018 - Dec 07 2018

Time	Monday	Tuesday	Wednesday	Thursday	Friday
10:00	⬛	⬛			
10:20	⬛	⬛			
10:40	⬛	⬛			
11:00	🟩	⬛			
11:20	🟩	⬛			
11:40	⬛	⬛			
12:00	🟩	⬛			
12:20	⬛	⬛			
12:40	🟩	⬛			
13:00	⬛	⬛			
13:20	⬛	⬛			
13:40	⬛	⬛			
14:00	🟩	⬛			
14:20	⬛	⬛			
14:40	⬛	⬛			
15:00	⬛	⬛			
15:20	🟩	⬛			
15:40	⬛	⬛			

User Details

Name: Joe King

Email: jk@wpi.bom

Search for TimeSlots

To search for a timeslot, use one or more options

[Find TimeSlots]

Year: --- Year ---

Month: December

Weekday: Monday

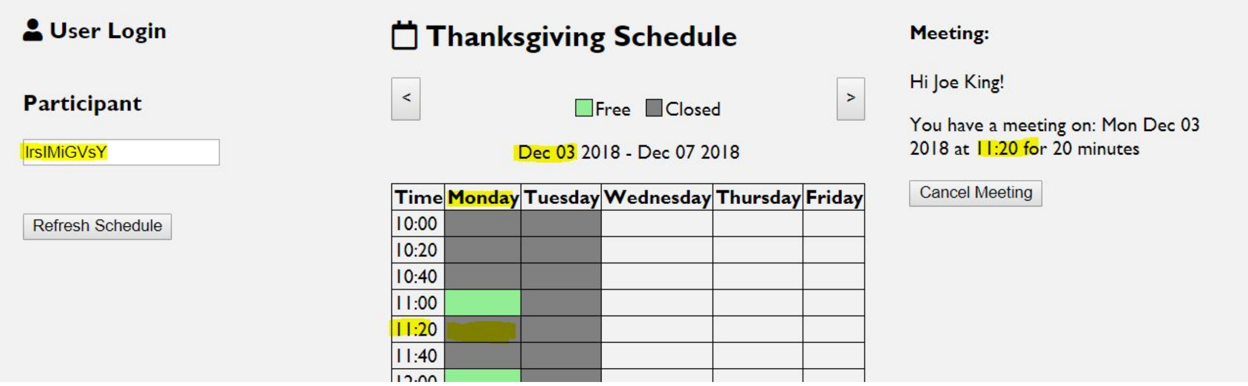
Day: mm/dd/yyyy

Hour: --- Hour ---

12/3/2018, 11:00:00 AM, 20 mins
12/3/2018, 11:20:00 AM, 20 mins
12/3/2018, 12:40:00 PM, 20 mins
12/3/2018, 2:00:00 PM, 20 mins
12/3/2018, 3:20:00 PM, 20 mins
12/3/2018, 12:00:00 PM, 20 mins

[Create Meeting]

Meeting Created



User Login

Participant

IrsIMiGVsY

[Refresh Schedule]

Thanksgiving Schedule

[<] 🟩Free ⬛Closed [>]

Dec 03 2018 - Dec 07 2018

Time	Monday	Tuesday	Wednesday	Thursday	Friday
10:00	⬛	⬛			
10:20	⬛	⬛			
10:40	⬛	⬛			
11:00	🟩	⬛			
11:20	🟩	⬛			
11:40	⬛	⬛			
12:00	🟩	⬛			

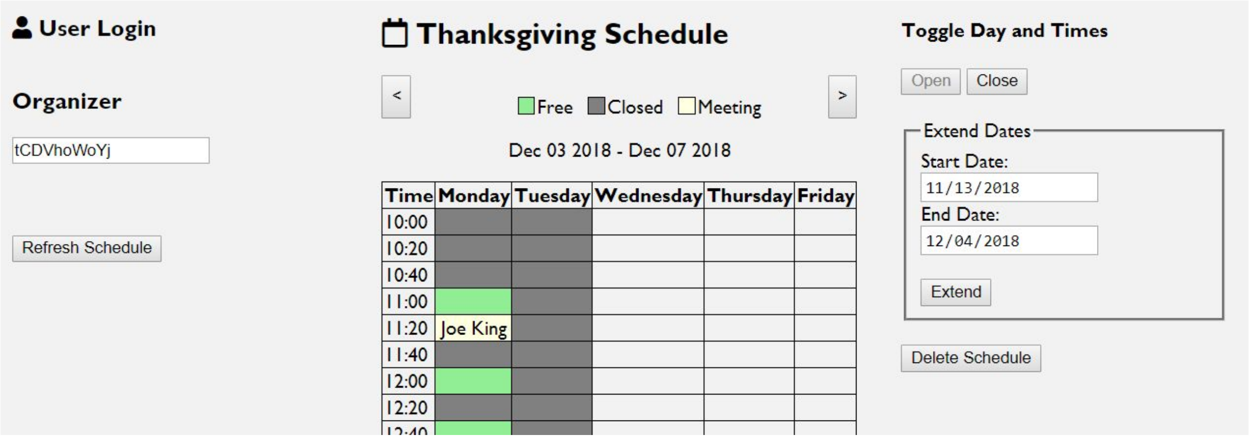
Meeting:

Hi Joe King!

You have a meeting on: Mon Dec 03 2018 at 11:20 for 20 minutes

[Cancel Meeting]

Organizer View



User Login

Organizer

tCDVhoWoYj

Refresh Schedule

Thanksgiving Schedule

< 🟩Free ⬛Closed □Meeting >

Dec 03 2018 - Dec 07 2018

Time	Monday	Tuesday	Wednesday	Thursday	Friday
10:00	⬛	⬛			
10:20	⬛	⬛			
10:40	⬛	⬛			
11:00	🟩	⬛			
11:20	Joe King	⬛			
11:40	⬛	⬛			
12:00	🟩	⬛			
12:20	⬛	⬛			
12:40	🟩	⬛			

Toggle Day and Times

Open Close

Extend Dates

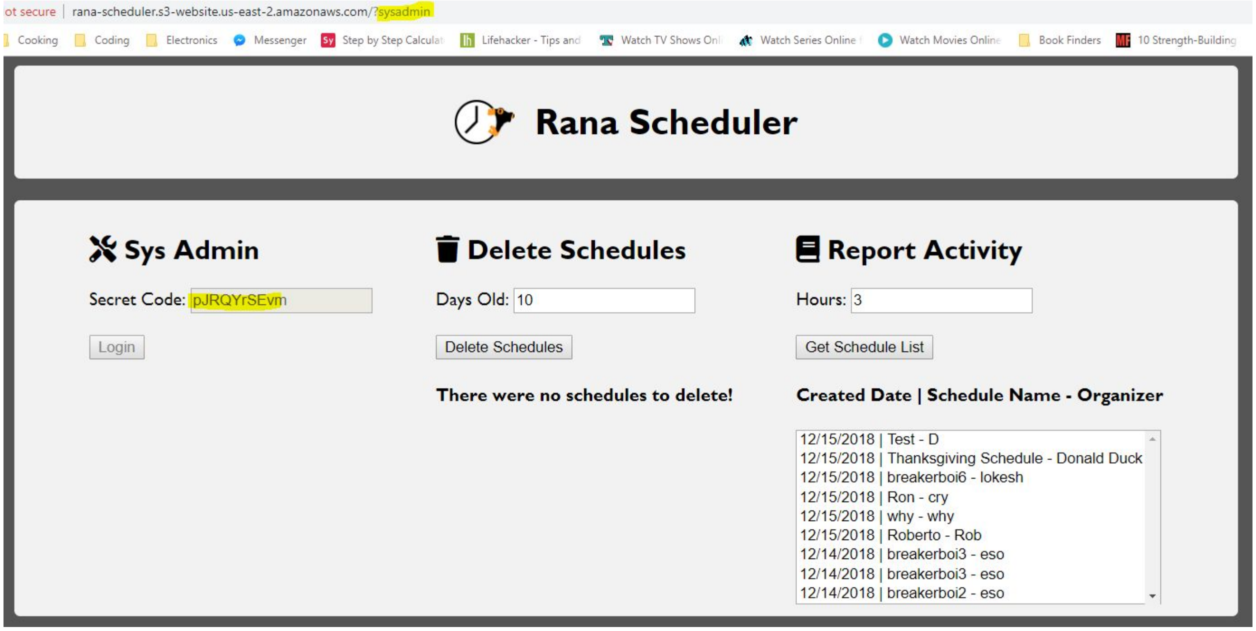
Start Date: 11/13/2018

End Date: 12/04/2018

Extend

Delete Schedule

Sys Admin View



ot secure | rana-scheduler.s3-website.us-east-2.amazonaws.com/?sysadmin

Rana Scheduler

Sys Admin

Secret Code: pJRQYrSEvm

Login

Delete Schedules

Days Old: 10

Delete Schedules

There were no schedules to delete!

Report Activity

Hours: 3

Get Schedule List

Created Date | Schedule Name - Organizer

12/15/2018 | Test - D
12/15/2018 | Thanksgiving Schedule - Donald Duck
12/15/2018 | breakerboi6 - lokesh
12/15/2018 | Ron - cry
12/15/2018 | why - why
12/15/2018 | Roberto - Rob
12/14/2018 | breakerboi3 - eso
12/14/2018 | breakerboi3 - eso
12/14/2018 | breakerboi2 - eso