

SoftwareTeamAwesome's Final Project: Appointment Scheduler

CSCI 3308 - Software
Development Methods and Tools
Group Project - Milestone 6
Spring 2018



Team Members

- Imran Dawud
- Andrew Gelfman
- Kyle Gorges
- Abby Johnson
- Yuxiang Wang
- Dongyu Wu



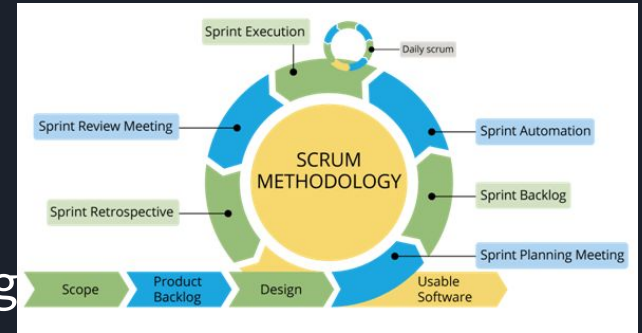
Tools We Utilized

- Agile/Scrum, Peer Code Reviews - Project management
- Trello - To manage what each person had to do
- Github - To compile all of the work and code we did
- Slack - To communicate with each other
- Heroku - Utilized for deployment and hosting
- MySQL - Relational database management
- MyTAP - Database testing
- NodeJS (+plugins) - Runtime environment, front/back interaction
- EJS - View rendering

Project Management Methods



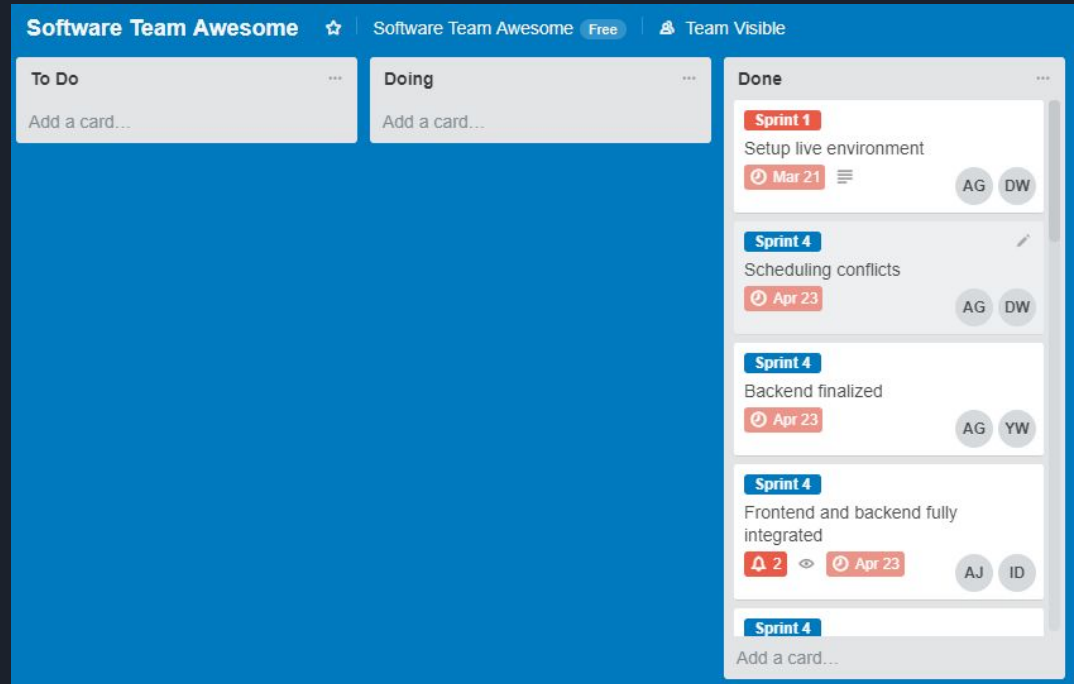
- Agile/Scrum Methodology
 - Continuous integration testing
 - Measured sprints
- Peer Code Reviews
 - Constant application feedback
 - Ensured universal understanding within group



Project Tracker



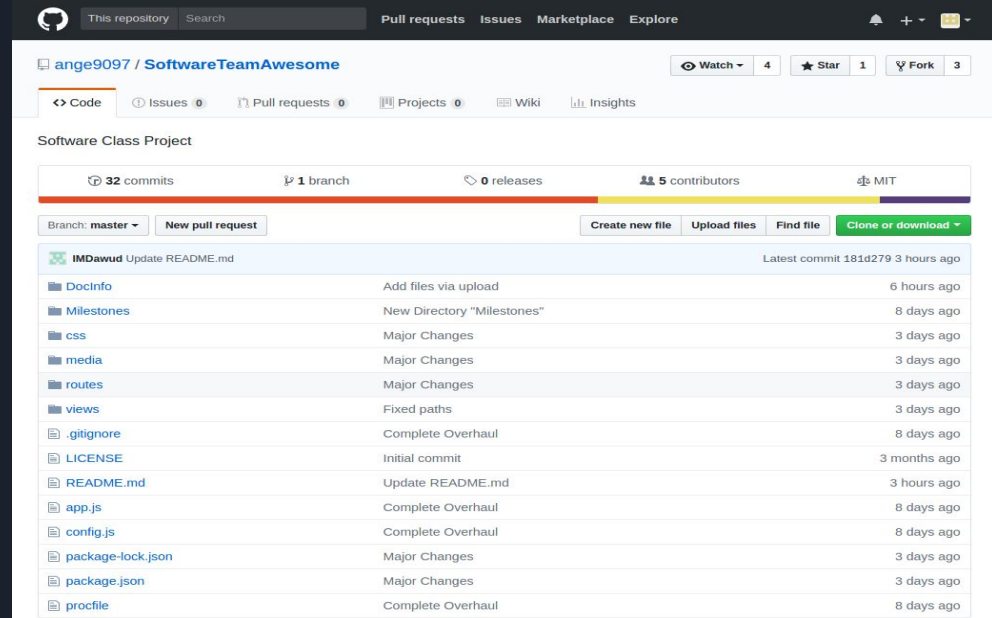
- Trello
 - Cards track task progress
 - Members assign themselves to tasks
 - Move cards as tasks are completed



VCS Repository



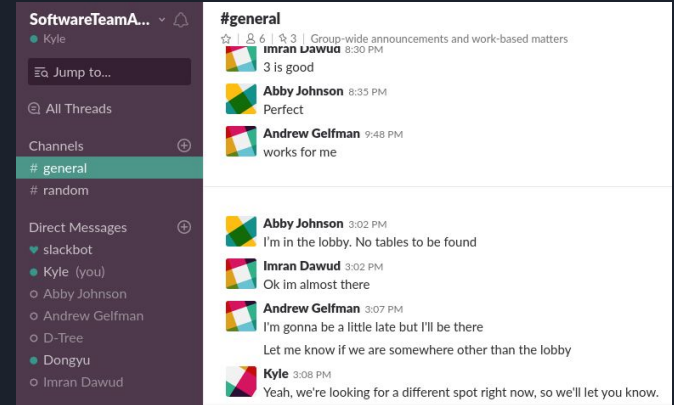
- Git/GitHub
 - Ensured team was in sync
 - Latest version of code (version control)



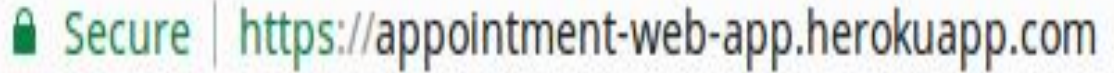
Communication Tool



- Slack
 - Used to facilitate communication between team members
 - Functions well on both computers and mobile devices



Deployment Environment



- Heroku
 - PaaS (Platform as a Service)
 - Used to deploy and host our web application
 - Cloud-based hosting
 - Free version is limited:
 - Sleeps after inactivity
 - RAM limited
 - Workers limited



Database



- MySQL
 - Relational database management system
 - Stores the data for:
 - User information
 - Available appointments
 - Scheduled appointments



Testing Tool



- MyTAP
 - “Test Anything Protocol”
 - MySQL testing suite
 - Used to verify existences and values of database tables

```
Terminal - user@cu-cs-vm: ~/SoftwareTeamAwesome/Testing/mytap-0.02
File Edit View Terminal Tabs Help
user@cu-cs-vm:~/SoftwareTeamAwesome/Testing/mytap-0.02$ mysql -u root -p --disable-pager --batch --raw --skip-column-names --unbuffered --database testfinal --execute 'source test.sql'
Enter password:
ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: YES)
user@cu-cs-vm:~/SoftwareTeamAwesome/Testing/mytap-0.02$ mysql -u root -p --disable-pager --batch --raw --skip-column-names --unbuffered --database testfinal --execute 'source test.sql'
Enter password:
1..7
ok 1 - This should always pass
ok 2 - This verifies APerkins in users
ok 3 - This verifies JLeslie in users
ok 4 - This verifies JSmith in users
ok 5 - This verifies APerkins in calendarEvents
ok 6 - This verifies JLeslie in calendarEvents
ok 7 - This verifies JSmith in calendarEvents
user@cu-cs-vm:~/SoftwareTeamAwesome/Testing/mytap-0.02$
```

```
1  -- Start a transaction.
2  BEGIN;
3
4  -- Plan the tests.
5  SELECT tap.plan(7);
6
7  SET @testuser1='APerkins';
8  SET @testuser2='JLeslie';
9  SET @testuser3='JSmith';
10
11 -- Run the tests.
12 SELECT tap.pass( 'This should always pass' );
13
14 SELECT tap.eq(username, @testuser1, 'This verifies APerkins in users') FROM users WHERE username='APerkins';
15
16 SELECT tap.eq(username, @testuser2, 'This verifies JLeslie in users') FROM users WHERE username='JLeslie';
17
```



Runtime Environment



- Node.js
 - JavaScript runtime environment
 - Used to allow for dynamic content on the web pages
 - Interaction between web pages and database

```
1 var express = require('express')
2 const PORT = process.env.PORT || 5000
3 var app = express()
4 var mysql = require('mysql')
5 var myConnection = require('express-myconnection')
6 var config = require('./config')
7 var dbOptions = {
8   host: config.database.host,
9   user: config.database.user,
10  password: config.database.password,
11  port: config.database.port,
12  database: config.database.db
13 }
14
15 app.use(myConnection(mysql, dbOptions, 'pool'))
16 app.set('view engine', 'ejs')
17
18 var index = require('./routes/index')
19
20 var expressValidator = require('express-validator')
21 app.use(expressValidator())
22
23 var bodyParser = require('body-parser')
24 app.use(bodyParser.urlencoded({ extended: true }))
25 app.use(bodyParser.json())
26
27 var methodOverride = require('method-override')
28 app.use(methodOverride(function(req, res){
29   if(req.body && typeof req.body === 'object' && '_method' in req.body){
30     var method = req.body._method
```

Javascript Templating Tool



`<%= EJS %>`

- EJS
 - “Embedded JavaScript”
 - Allows for HTML markup via JavaScript
 - Used to manage web page view rendering

```
1 <html>
2 <head>
3   <link href="../css/index.css" type="text/css" rel="stylesheet"/>
4   <title>Home Page</title>
5 </head>
6
7 <body>
8   <div class="menubar">
9     <%- include layouts/menubar_guest.ejs %>
10  </div>
11  <section class="panel">
12    <p class="avatar">
13      
14    </p>
15    <h3>Dr. Doctor Inc</h3>
16    <p class="lead">Welcome to Dr. Doctor's page! </p>
17    <p>
18      <a class="btn" href="../user/login">
19        <i class="fa fa-bars"></i> Make Appointment
20      </a>
21    </p>
22    <p class="tags">
23      303-333-3333 <br>
24      2055 Regent Drive, Rm., Boulder, CO 80309-0020
25    </p>
26  </section>
27 </body>
28
29 </html>
```



Challenges Encountered

- Calendar
 - Tried to create an interactive calendar
 - Interaction with database was difficult
- Communication
 - Initially difficult to set up meetings
 - Initially struggled to maintain consistent communication about project status
- Managing the form input and output
 - Front-end/back-end interaction



Overcoming the Challenges

- Calendar
 - Instead decided to utilize the built-in date/calendar functionality of HTML
- Communication
 - Utilized Slack, texting, and email more frequently
 - Scheduled more regular meetings, even if not everyone could attend
- Managing the form input and output
 - Practice with node.js was able to resolve this issue



Project Demonstration

- <https://software-team-awesome.herokuapp.com>



Image Sources

- https://assets-cdn.github.com/images/modules/logos_page/GitHub-Mark.png
- <https://www.mysql.com/common/logos/logo-mysql-170x115.png>
- <https://brand.heroku.com/static/media/heroku-logotype-horizontal.81c49462.svg>
- <https://d2k1ftgv7pobq7.cloudfront.net/meta/u/res/images/brand-assets/Logos/0099ec3754bf473d2bbf317204ab6fea/trello-logo-blue.png>
- <https://zaynabzahrablog.files.wordpress.com/2017/10/11-e1507403418292.png?w=588>
- <https://nodejs.org/static/images/logos/nodejs-new-pantone-black.png>
- <https://cdn.dealstreetasia.com/uploads/2016/04/Slack-e1459571703586.png?resize=750,417>
- <http://ejs.co/#promo>