

EE / CprE / SE 492 - sdmay23-06

Production Run Reports

March 4th - March 24th

Client: JEDA Polymers

Faculty Advisor: Dr. Stoytchev

Team Members:

Colton Carlson	-	Team Lead, Backend, Email API
Hayden Havelka	-	Backend, User Authentication
Jay Arnold	-	Backend, User Authentication
Connor Linn	-	Frontend, Login / Sign Up Pages
Noah Meyer	-	Frontend, Request Report / Display Report Pages

Past Week Accomplishments

- **Colton:**
 - Worked on implementing and optimizing canary labs web api calls.
 - Created http client factory
 - No longer need to manually dispose of clients as factory will dispose of them.
 - Moves construction of all http clients to one method call, making modifications easier.
 - Uses a custom handler as CanaryServer uses self signed certificates, which if left as default would throw runtime exceptions.
 - Used C#'s native json serializer/deserializer
 - Created custom deserializer for data points, which can take an optional parameter if the value being deserialized is not equal to the optional parameter, then no value is deserialized.
 - This makes it so when we look for what line a work order is in, instead of manually checking all data points, we only need to check what web api call has values after being deserialized.
 - Reintroduced and reintegrated logging into the project.
 - Took previously written logger and migrated to using Serilog
 - Things logged manually by Production Run Reports is no longer written to console, instead written to log file in format:
 - “Date/Time [Error Level] Message”
 - File operations are now handled by serilog which is thread safe.
 - Settings File:
 - Added settings file to program

- Following are variables read at run time, which can be changed from file:
 - Canary Server IP
 - Canary Server Port
 - SMTP Server IP
 - SMTP Server Port
 - Max concurrent api calls
 - Log File location
 - Report Email sender address
 - Report Email sender name
 - If the settings file is corrupted, missing, or unreadable, a new file will be created with all settings set to default.
 - Authentication Controller
 - Created a basic authentication controller that can be used.
 - Reduces front end authentication calls to posts and gets.
 - JWT Tokens can be created and verified, with a 7 day lifespan.
 - A new secret key is created every time the server starts.
 - Password hashing is handled by Microsoft.Identity, a state of the art authentication library produced by Microsoft.
 - Everything is stored locally as a sqlite database, in event database becomes corrupted or unreadable, a new empty database is created and formatted to be used instead.
 - Email Service
 - Integrated basic emailing into pdf report generation, but as authentication was not done it was sent to a static pre-determined email address.
 - This however did confirm that the local smtp server the client uses can send outbound mail to external emails.
- **Hayden:**
 - Backend and Frontend Authentication (Alternative method that did not end up being used)
 - Figured out how to connect razor web pages to react web pages
 - Overrode default web razor web pages from ASP.NET Core Auth library with new pages that better match react pages
 - Connected backend authentication to razor web pages
 - Worked on and recorded progress for project overview video.
- **Jay:**
 - Login Authentication method
 - Added logic that checks to see if a user has correct credentials, confirmed or unconfirmed email, or incorrect credentials
 - Generates a logon token and sends to frontend
 - Sends new email confirmation to user

- Made slides and talked over some slides for project overview video
- **Connor:**
 - Frontend Integration
 - Post and Gets for work order resolution
 - Passing work order to display page using blobs
 - Edge Case protection
 - Checking work order values for legal inputs
 -
- **Noah:** Front end development
 - Began front and backend integration
 - User credential POSTs to back end
 - Assisted Connor with work order & line number POSTs to back end
 - Assisted with accurate report PDF rendering on front end
 - Added new webpages for user management
 - Includes ability for users to reset password and confirm their email.
 - Edge case protection
 - Cancel work order requests, email and password regex, disabling buttons until requirements are met, and error handling.

Pending Issues

- **Team Member(s):** None

Individual Contributions

Team Member	Contribution	This Report Hours	Total Hours (492)
Colton Carlson	Web API implementation, Settings Parsing, Serilog implementation, Emailing	50	145
Hayden Havelka	Backend and Frontend Alternate Authentication Method	40	92
Jay Arnold	Login authentication method	18	54
Connor Linn	Frontend Integration	30	59
Noah Meyer	Frontend Integration & UI improvements	26	73

Plans for Coming Two Weeks

- **All:**
 - Start client beta test.
 - Create Readme for project
- **Colton:**
 - Help finish authentication
 - Start work on WIX installer
- **Hayden:**
 - Help write test cases for backend and troubleshoot any issues for the client
 - Help finish up any additional work for the backend
- **Jay:**
 - Help finish up user authentication on the backend
 - Email confirmation, generate token, password reset
- **Connor:**
 - Bundling and cookies
 - Find how to send authentication tokens through cookies and using bundling with webpack if applicable
 - Document Frontend
- **Noah:**
 - Finish user authentication on the frontend
 - Make POSTs to backend with user credentials on the Login, Signup, Reset Password, and Email Confirmation webpages
 - Document frontend