

# Project Plan

## sdmay23-06

<b>Project Management Style</b>	<b>1</b>
<b>Task Decomposition</b>	<b>2</b>
Front End	2
Backend	2
Milestones and Metrics	2
<b>Project Timeline</b>	<b>4</b>
Gantt chart	4
Personal Effort	5
<b>Risks</b>	<b>6</b>
Risk Management	6

# Project Management Style

Our group decided to go with the Agile development methodology. One of the main reasons we picked this is because we have a client with needs that could change. With Agile, we are able to adapt to changing needs and implement/change the code. Another reason is that agile is used by most software/tech companies. Using agile for this project will be beneficial because we are learning popular methods that are applicable to the real world. We are using GitLab for the repo and to track progress.

# Task Decomposition

Identify tasks like front end back end, and break stuff down

## Front End

- Create 5 pages that the user will interact with
  - (Log in, Request Data, Results, Make New Account, Reset Password)
  - Make all pages follow a similar format for ease of use and a uniform look
- Integrate REST API to make HTTP calls to backend
- Testing

## Backend

- Report generation
  - Creates report from template with data from database
- Database
  - Use Canary API to get data from database
- Email API
  - Use Email API to send report PDFs through email
- Log
  - Configure start/end times for report generation
- Config File
- Windows Server 2014
  - Configure the application to be hosted on Windows Server 2014

## Milestones and Metrics

Go over milestones for each task, testable and easy to say yes or no it works.

1. An accurate report is generated with 90% accuracy.
2. A report is generated within 15 seconds.
3. A report is emailed to specific emails using the frontend.
4. The log file is correctly displaying start/end times for report generation.
5. The frontend correctly displays a report 90% of the time.
6. The frontend allows new accounts with usernames and passwords 90% of the time.
7. The reset/forgot password page works 90% of the time.

8. The application works on 3 out of the 5 total browsers.

# Project Timeline

## Gantt chart

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Report Generation	Orange															
Query Database		Orange														
Log File					Orange											
Formatting Data			Orange	Orange												
Email API						Orange										
Login Page	Green															
Request Data Page (WO# and L#)		Green	Green													
Results Page				Green												
Make New Account Page					Green											
Reset/Forgot Password Page						Green										
Integration									Blue	Blue						
Testing							Blue				Blue	Blue				Blue
Optimization and Bug Fixing								Blue				Blue	Blue	Blue		

Top row = week  
 Column = Task  
 Orange = Back end  
 Green = Front end  
 Blue = Both

## Personal Effort

The following chart is a basis for the expected number of hours per task that is expected to be spent. These numbers were chosen based on the factors of: number of individuals working on a task, time to learn necessary skills per task, average time spent on similar tasks.

<b>Task</b>	<b>Hours</b>
Report Generation	10
Query Database	16
Log File	6
Formatting Data	40
Email API	20
Login Page	10
Request Data Page (WO# and L#)	16
Results Page	25
Make New Account Page	10
Reset/Forgot Password Page	8
Integration	20
Testing	45
Optimization and Bug Fixing	45

# Risks

			Impact		
Probability	Very Low	Low	Medium	High	Very High
Highly Probable	5 - Moderate	10 - Major	15 - Major	20 - Severe	25 - Severe
Probable	4 - Moderate	8 - Moderate	12 - Major	16 - Major	20 - Severe
Possible	3 - Minor	6 - Moderate	9 - Moderate	12 - Major	15 - Major
Unlikely	2 - Minor	3 - Moderate	6 - Moderate	8 - Moderate	10 - Major
Rare	1 - Minor	2 - Minor	3 - Minor	4 - Moderate	5 - Moderate

\*\* Chart values are referenced in the Risk Mitigation section when discussing risks and their solutions.

# Risk Management

Risk	Solution
Client backing out - 10	As a team we are developing an important tool for their business. Additionally we are keeping in close contact with them so that they have an understanding of the development process.
Losing access to the database that holds the machine data - 9	Ensuring that we constantly have the most up to date credentials to the

	machine's database.
Bad data injection from malicious users - <b>10</b>	Authenticating users to ensure only users with the correct credentials are accessing the system.
The web application does not load properly on different browsers - <b>3</b>	Check that the frontend is able to run and display correctly on multiple browsers.
Credentials server fails to authenticate users	Have thorough testing on the login system, as well as a flushed-out system to reset a users page.
PDF generation fails - <b>9</b>	Check connection to database, check if the queries are correct.
Queries return wrong results or NULL even when the correct data is there - <b>6</b>	Check the connection to the database, and make sure the queries are correct.
Report generates with incorrect data - <b>12</b>	Test to make sure that we are getting correct data from the database. Once the queries are working correctly, we can then inject data into the report.
Report is not emailed to the correct recipients - <b>6</b>	Test to make sure that emails are sending reports to the correct recipients.
Log file does not show correct PDF generation data - <b>6</b>	Check to make sure reports are being generated correctly by testing.
New user's account is not registered into the system - <b>12</b>	Check to make sure the APIs are being used correctly by testing.