
SPRINT FIVE

CS2450-002, Team 2

Cody Strange-*Scribe and Information Manager*

Ethan Taylor-*GUI Developer*

Jaden Albrecht-*Team Manager*

Tyler Deschamps-*Chart and Milestone document builder*

Jordan Van Patten-*V&V and Tester*

Kole Davis- *QA Manager*

Craig Sharp-*Stakeholder*

Table of contents

<i>Management.....</i>	<i>3</i>
<i>Procedures.....</i>	<i>3</i>
<i>Plans.....</i>	<i>16</i>
<i>Previous Sprint Items.....</i>	<i>17</i>
<i>Metrics.....</i>	<i>19</i>
<i>Defect Analysis.....</i>	<i>19</i>
<i>Ishikawa Diagram.....</i>	<i>22</i>
<i>Software Metrics.....</i>	<i>22</i>
<i>Function Point Analysis.....</i>	<i>35</i>
<i>Alpha / User Acceptance Test.....</i>	<i>36</i>
<i>Charts/Templates.....</i>	<i>38</i>
<i>Work breakdown structure.....</i>	<i>38</i>
<i>Pert Chart.....</i>	<i>39</i>
<i>Gantt Chart.....</i>	<i>40</i>
<i>Burndown Chart.....</i>	<i>40</i>
<i>Meeting Logs.....</i>	<i>41</i>
<i>Meeting Log#17.....</i>	<i>41</i>
<i>Meeting Log#18.....</i>	<i>42</i>
<i>Meeting Log#19.....</i>	<i>43</i>
<i>Meeting Log#20.....</i>	<i>44</i>
<i>Meeting Log#21.....</i>	<i>46</i>
<i>Meeting Log#22.....</i>	<i>47</i>
<i>Meeting Log#23.....</i>	<i>48</i>
<i>Meeting Log#24.....</i>	<i>50</i>
<i>Meeting Log#25.....</i>	<i>51</i>

Management

Procedures

Verification & Validation

Summary: V&V documents for sprint five specifically.

APPLICATION QUALITY ASSURANCE CHECKLIST

Project Name	EmpDat	Project #	SPR-5
Application Name		Application #	
Project Manager		Delivery Manager	
Date Completed	4/15/2022		

IMPORTANT NOTES FOR COMPLETING THIS DOCUMENT

1. Development Framework

Validation Questions	Yes	No	NA	Comments
1. Has the application been developed with the most recent OCIO-sanctioned version of the framework for the chosen technology?	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

1. Development IDE

Validation Questions	Yes	No	NA	Comments
1. Has the application been developed using an Integrated Development Environment that was approved by the OCIO?	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

1. Decoupling Business Logic From The Presentation Layer

Validation Questions	Yes	No	NA	Comments
1. Is the presentation layer of the application free from business logic?	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
1. Has the presentation layer of the application been developed in accordance with prevailing industry standards?	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

1. Record Locking / Concurrent Users

Validation Questions	Yes	No	NA	Comments
1. Have precautions been taken to avoid data clashes?	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

1. Passwords				
Validation Questions	Yes	No	NA	Comments
1. Does the system have functionality to allow the user to revise their password and force user account expiry?	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	The user is able to change their password, but does not force user account expiry, because user accounts would be deactivated or expired by an admin
1. Does the system support protected storage of passwords with privileged user access? The system should not support passwords in clear text embedded either in the application code, automated scripts, or the database?	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
1. Does the system meet the standard password requirements?	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Password requirements will be added in at a later execution of our program
1. Are the passwords in the production environment different than those in a non-production environment?	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	For now, passwords in the production environment do not currently have any requirements, but that will change later
1. Are all vendor supplied default passwords revised prior to placing the application in a production environment?	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
1. Are passwords for privileged accounts different than passwords for non-privileged accounts?	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	They are different passwords if the user chooses, but the user is the one that chooses passwords

1. Logging and Auditing				
Validation Questions	Yes	No	NA	Comments
1. Based on the application's Information Security Classification, does the application meet the logging functional control requirements?	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
1. Based on the application's Information Security Classification, does the application meet the auditing functional control requirements?	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	

1. Modularized Code With No Duplication				
Validation Questions	Yes	No	NA	Comments
1. Is the application modularized?	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
1. Has code duplication been avoided?	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

1. Consistency of Code				
Validation Questions	Yes	No	NA	Comments
1. Is the code written in a consistent manner throughout the application?	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
1. Have all developers followed the same coding style and naming conventions?	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
1. Have all developers followed the coding best practices as set out by the organization which owns the technology?	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

1. Code Comments				
Validation Questions	Yes	No	NA	Comments
1. Does all application code include sufficient comments for support personnel?	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

1. Code Comments				
Validation Questions	Yes	No	NA	Comments
1. Does each code unit have its own brief and accurate description?	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

1. Error Handling – End User				
Validation Questions	Yes	No	NA	Comments
1. Does the application handle all the errors that could reasonably be expected to occur?	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
1. Do the error messages contain minimal but meaningful information?	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
1. Does the application avoid displaying system information in error messages?	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
1. Are the error messages kept in a single location?	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Error messages are kept in separate locations in the code, and in the application it is shown below the effected area, I.E. if date is incorrect, it will display the error message below the date entry box

Error Logging				
Validation Questions	Yes	No	NA	Comments
1. Are all errors for the application being logged?	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	We are manually logging any errors we find
1. Is logging being done on each server tier?	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
1. Are the logs kept in a single location / directory / database?	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	We use an internet application where we report our bugs/errors
1. Are the logged errors specific enough to assist support personnel in troubleshooting production problems?	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
1. Is the code that logs the error messages written in a modular way?	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
1. Are the log files free of personally sensitive or identifiable information?	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

1. Field Validations				
Validation Questions	Yes	No	NA	Comments
1. Are fields being checked for the correct type (e.g. date, integer, etc.) and the correct range of values (e.g. 1 – 12 for month)?	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
1. Are field values being validated with regular expressions where possible (e.g. validating email addresses and dates for valid formats)?	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Date of birth is not currently being validated
1. Do the validations resulting in error messages prevent data from being written to persistent storage (databases, files, etc.)?	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
1. Are the validations being performed within the business logic, as well as on the presentation layer?	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
1. Have the validations been written so that users cannot bypass them?	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
1. Are all of the field lengths and types within the application consistent with the column lengths and types declared within the underlying database tables?	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

1. Field Validations				
Validation Questions	Yes	No	NA	Comments
1. Are user inputs being sanitized (without exceptions) according to OWASP recommendations?	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

1. Dates				
Validation Questions	Yes	No	NA	Comments
1. Does the application validate dates in a way that is consistent with the system design specifications and business rules?	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Dates are automatically inputted by the code, not by the user, so there is no date validation. DOB is not yet validated and mandated to be in the month/day/year format but will be added in a future version
1. Do all relevant dates include a timestamp?	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

1. Hard-Coded Values				
Validation Questions	Yes	No	NA	Comments
1. Does the application code avoid use of hard-coded values?	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
1. Do all hard-coded values reside exclusively within configuration and constant, centralized locations? (Central Locations that enable changes without recompiling source code)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

1. System Testing			
Validation Questions	Yes	No	Comments
1. Did the application pass all positive tests?	<input checked="" type="checkbox"/>	<input type="checkbox"/>	There was one positive test that failed, but the rest of them passed. The test that failed was editing a user's ID, and then searching for that user afterwards
1. Did the application pass all negative tests?	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
1. Have client testers completed the formal test plan in its entirety?	<input type="checkbox"/>	<input checked="" type="checkbox"/>	We have not made a formal test plan yet
1. Did the application pass all tests included in the formal test plan?	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
1. Have all positive / negative test cases and test case results been documented?	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

1. Regression Testing				
Validation Questions	Yes	No	NA	Comments
1. As new capability is introduced, is the new capability tested?	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
1. Have all previous tests been reconducted with the results compared against expected results?	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
1. Is every capability of the software supported with a test case and is the test case added to the test case library to support final and future system testing?	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
1. As bugs are detected and fixed, is the test that exposed the bug recorded and regularly re-tested after subsequent changes are applied to the application?	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

1. Load Testing/Volume Testing				
Validation Questions	Yes	No	NA	Comments
1. Has the application been tested with a large number of concurrent users (i.e. a number of users that is representative of peak system usage)?	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
1. Has the application been tested with large numbers of concurrent transactions (i.e. a number of transactions that is representative of peak system usage)?	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
1. Did the system perform well with a large number of concurrent users?	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
1. Did the system perform well with a large number of concurrent transactions?	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
1. Are end-users satisfied with the application's performance and responsiveness during everyday use?	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	

1. Certificates / Environment Software				
Validation Questions	Yes	No	NA	Comments
1. Has all proprietary and copyrighted software been properly licensed for government use?	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
1. Have special software/certificate requirements been documented?	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
1. Does the documentation provide expiration dates and instructions for renewal?	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
1. Is the system / application free from trial versions of software?	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	

1. Business Requirements - Traceability				
Validation Questions	Yes	No	NA	Comments
1. Have all of the business requirements been met by the finished application?	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	The application is not yet finished, but we have met the requirements given for this current phase
1. Has all of the required functionality been met by the finished application?	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	The current application's requirements are met by what we currently have for our application

1. Source Code				
Validation Questions	Yes	No	NA	Comments
1. Has the final approved version of the Application Code been provided to Application Services for use and maintenance during the Transition Period?	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	We do not have the final version of our code
1. Has a test build been completed by Application Services using the code that has been handed over?	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
1. Has a copy of the version of Open Source Code used by the application been provided to Application Services for retention? (Links are not recommended)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
1. Have the 3 rd party developer code / plug-ins (e.g. Axis2, Eclipse) been identified and provided to Application Services for the continued maintenance of the application? (Links to the utility not satisfactory, 3 rd party products need to be provided)	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	

1. Database Design				
Validation Questions	Yes	No	NA	Comments
1. Have the database tables been normalized?	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
1. Keys based on sequence numbers have unique sequences.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

1. Database Design				
<i>Validation Questions</i>	Yes	No	NA	Comments
1. Are all keys and required fields set to 'not null' in all tables of the database?	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
1. Have triggers, stored procedures, sequences, and constraints been properly utilized?	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	

1. Transition To Support Personnel				
<i>Validation Questions</i>	Yes	No	NA	Comments
1. Have accounts been created on all servers for the appropriate support personnel?	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
1. Have the necessary firewall rules been added to allow Application Services support personnel to access the relevant servers (i.e. via the Jump Box)?	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
1. Have all server environments (development, test/staging, and production) been fully created?	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
1. Are all of the server environments entirely consistent with each other?	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

Backlog

Summary: The sprint five backlog covers the tasks in the sprint and the team member that is assigned to the task, along with a due date attached to that task. It also has the tasks from the previous sprint that were not completed in that sprint.

~ SPR5 Task List		
Task	Assigned To	Due Date
Create Backlogs with Timeboxes - Includes Task Assignment	Tyler	04/13/2022
Collect Indicators	Cody	04/06/2022
Create Metric Report(s)	Cody	04/15/2022
Current Sprint Meeting Logs	Jaden	04/18/2022
Change Order Development	Jaden	04/15/2022
Gather Defects Data	Tyler / Ethan	03/30/2022
Defects Report	Kole	04/08/2022
Last touch-ups on program	Jaden	04/15/2022
Current Sprint Work Breakdown Structure	Cody	03/30/2022
Current Sprint Pert, Gantt, and Burndown charts	Tyler	04/18/2022
Gather and Format decomposed Charts	Tyler	04/15/2022
Performance Plan	Cody	04/18/2022
Function Points Chart	Cody	04/15/2022
Ishikawa diagram(s) of a defect (not the easiest nor the hardest to address)	Kole	04/15/2022
User Acceptance	All Team Members	04/06/2022
Testing of Implemented Change Orders	Jordan	04/06/2022
Help sections	Ethan	04/13/2022
Sprint PDF Document	Jaden	04/18/2022

Team Members
Jaden Albrecht
Cody Strange
Ethan Taylor
Jordan Van Patten
Kole Davis
Tyler Deschamps

~ Previous Sprint Backlog		
Task	Assigned To	Due Date
Finish Test Cases	Jordan	04/01/2022
Updated Class Diagram	Ethan	04/06/2022
Correct CSV Database to Text Database	Ethan	04/15/2022

Team Performance

Summary: This review tracks each team member and the total amount of hours they logged for each sprint. It also covers the amount of meetings they attended per sprint and the tasks that they completed during the sprint.

T2-002 PERFORMANCE REVIEW

SPR-1

Ethan Taylor:

- Total Hours: 9 hours
- Total Meetings Attended: (3/3)
- Tasks Completed:
 - Personal MoSCoW chart for Delphi method (100%)
 - Thoroughly read and reviewed all SPR-1 document requirements (100%)

Jaden Albrecht:

- Total Hours: 9 hours
- Total Meetings Attended: (3/3)
- Tasks Completed:
 - Personal MoSCoW chart for Delphi method (100%)
 - Thoroughly read and reviewed SPR-1 document requirements (100%)

Cody Strange:

- Total Hours: 9 hours
- Total Meetings Attended: (3/3)
- Tasks Completed:
 - Personal MoSCoW chart for Delphi method (100%)
 - Thoroughly read and reviewed SPR-1 document requirements (100%)
 - SPR-1 document (100%)

Tyler Deschamps:

- Total Hours: 9 hours
- Total Meetings Attended: (3/3)
- Tasks Completed:
 - Personal MoSCoW chart for Delphi method (100%)
 - Thoroughly read and reviewed SPR-1 document requirements (100%)

SPR-2

Ethan Taylor:

- Total Hours: 6 hours 45 minutes
- Total Meetings Attended: (4/4)
- Tasks Completed:
 - GUI template (100%)
 - Rough draft for HLD (100%)
 - Review SPR-2 document requirements (100%)
 - Develop GUI classes (100%)

- Plain-English pseudocode (100%)
- Develop use-case scenarios based on requirements (100%)
- Docstring/comment outlining (100%)

Jaden Albrecht:

- Total Hours: 35 minutes
- Total Meetings Attended: (4/4)
- Tasks Completed:
 - Distribution of MoSCoW charts for Delphi method (100%)
 - Review SPR-2 document requirements (100%)
 - Review GUI template and use-cases for HLD (100%)
 - Review plain-English description of HLD (100%)
 - Establish connections between project requirements and tasks (100%)
 - Develop minute-tracking method (100%)
 - Keep meeting logs up to date (100%)

Cody Strange:

- Total Hours: 9 hours 20 minutes
- Total Meetings Attended: (4/4)
- Tasks Completed:
 - Fix SPR-1 document (100%)
 - Work breakdown structure for SPR-2 (100%)
 - Construct requirements spec (100%)
 - Synchronize requirements spec with final MoSCoW chart (100%)
 - Choose SPR-1 prototype candidate (100%)
 - SPR-2 document (100%)

Tyler Deschamps:

- Total Hours: 45 minutes
- Total Meetings Attended: (4/4)
- Tasks Completed:
 - Develop Pert/Gantt/burn-down charts for SPR-2 (100%)
 - Review SPR-2 document requirements (100%)
 - Logging for Pert/Gantt charts throughout SPR-2 (100%)

Jordan Van Patten*:

- Total Hours: 1 hour, 30 minutes
- Total Meetings Attended: (3/4)
- Tasks Completed:
 - Help with GUI class development (100%)
 - Review SPR-2 document requirements (100%)
 - V&V covering requirements and documentation (100%)
 - Review use-cases for HLD (100%)

SPR-3

Ethan Taylor:

- Total Hours: 20 hours, 45 minutes
- Total Meetings Attended: (6/6)
- Tasks Completed:
 - Review all SPR-3 documents (100%)
 - Review prototype code (100%)
 - Pseudocode for Login GUI page (100%)
 - Pseudocode for search page (100%)
 - Pseudocode for view/add/edit page (100%)
 - Login GUI (100%)
 - Search GUI (100%)
 - View/add/edit GUI (100%)
 - Deactivate employee GUI (100%)
 - Payroll GUI (100%)
 - Tkinter research

Jaden Albrecht:

- Total Hours: 19 hours, 40 minutes
- Total Meetings Attended: (6/6)
- Tasks Completed:
 - Review all SPR-3 document requirements (100%)
 - Keep meeting logs up to date throughout SPR-3 (100%)
 - SPR-3 change order request form (100%)
 - Develop version history folders in google drive (100%)
 - Build UML class diagrams (100%)
 - Review prototype code (100%)
 - UML research
 - Tkinter research
 - Pytest research

Cody Strange:

- Total Hours: 17 hours, 40 minutes
- Total Meetings Attended: (6/6)
- Tasks Completed:
 - Review all SPR-3 document requirements (100%)
 - Work breakdown structure for SPR-3 (100%)
 - Risk management plan (100%)
 - Testing documents (100%)
 - Field validation code for login page (100%)
 - Review Prototype code (100%)
 - Fix SPR-2 document (100%)
 - Reorganize google drive folders (100%)

- Change request board (100%)
- SPR-3 document (100%)

Tyler Deschamps:

- Total Hours: 15 hours, 30 minutes
- Total Meetings Attended: (6/6)
- Tasks Completed:
 - Review all SPR-3 document requirements (100%)
 - Update Pert/Gantt/burn-down charts throughout SPR-3 (100%)
 - Review prototype code
 - Search employee database pseudocode (100%)
 - Database pseudocode (100%)
 - CSV pseudocode (100%)
 - Database tables to CSV (100%)
 - JSON research

Jordan Van Patten:

- Total Hours: 6 hours, 50 minutes
- Total Meetings Attended: (3/6)
- Tasks Completed:
 - Review all SPR-3 document requirements (100%)
 - Review prototype code (100%)
 - Team file/folder naming conventions document (100%)
 - Quality assurance plan document (100%)
 - V&V documents for SPR-3 (100%)
 - Pytest research
 - Black research
 - Debugging software research

SPR-4

Ethan Taylor:

- Total hours: 9 hours, 40 minutes
- Total Meetings Attended: (3/3)
- Tasks Completed:
 - Review all SPR-4 document requirements (100%)
 - Implement add/edit validation code into GUI (100%)
 - Integrate database with payroll page updates (100%)
 - Update payroll page (100%)
 - Tkinter research

Jaden Albrecht:

- Total hours: 5 hours
- Total Meetings Attended: (3/3)
- Tasks Completed:

- Review all SPR-4 document requirements (100%)
- Keep meeting logs up to date throughout SPR-4 (100%)
- User manual (100%)

Cody Strange:

- Total hours: 14 hours
- Total Meetings Attended: (3/3)
- Tasks Completed:
 - Review all SPR-4 document requirements (100%)
 - Work break-down structure for SPR-4 (100%)
 - Usability tests (100%)
 - Risk management plan for SPR-4 (100%)
 - Maintenance plans (100%)
 - Deployment plan (100%)
 - SPR-4 document (100%)

Tyler Deschamps:

- Total hours: 15 hours, 30 minutes
- Total Meetings Attended: (2/3)
- Tasks Completed:
 - Review all SPR-4 document requirements (100%)
 - Update Pert/Gantt/burn-down charts throughout SPR-4 (100%)
 - Report receipts (100%)
 - Implement database logic into GUI (100%)
 - Merge database with Emp_Dat_V3 (100%)
 - Payroll and CSV integration (100%)

Jordan Van Patten:

- Total hours: 6 hours, 30 minutes
- Total Meetings Attended: (3/3)
- Tasks Completed:
 - Review all SPR-4 document requirements (100%)
 - V&V documents for SPR-4 (100%)
 - Test cases for add/edit/view and search GUI pages (100%)
 - Update quality assurance plan (100%)
 - Pytest documents for payroll page (100%)
 - Implement validation code for add/edit/view GUIs (100%)

SPR-5

Ethan Taylor:

- Total Hours: 25 hours 30min
- Total Meetings Attended: (9/11)
- Tasks Completed:
 - Review All SPR-5 document requirements (100%)

- Defects analysis (100%)
- Update UML diagrams (100%)
- Help button/user manual integration (100%)
- Integrate old database (100%)
- Integrate CSV database (100%)

Jaden Albrecht:

- Total Hours: 30 hours, 15 minutes
- Total Meetings Attended: (11/11)
- Tasks Completed:
 - Review all SPR-5 document requirements (100%)
 - Functional/non-functional requirements testing (100%)
 - Personal user acceptance test (100%)
 - Add search parameters and improvements to search page (100%)
 - Help with function-point analysis (100%)
 - SPR-5 meeting logs (100%)

Cody Strange:

- Total Hours: 19 hours, 40 minutes
- Total Meetings Attended: (11/11)
- Tasks Completed:
 - Review all SPR-5 document requirements (100%)
 - Help with function-point analysis (100%)
 - SPR-5 document (100%)
 - Metrics reports (100%)
 - Personal user acceptance test (100%)
 - Defects list (100%)
 - Risk management plan for SPR-5 (100%)

Tyler Deschamps:

- Total Hours: 12 hours
- Total Meetings Attended: (11/11)
- Tasks Completed:
 - Review all SPR-5 document requirements (100%)
 - Defects analysis (100%)
 - Backlogs for SPR-5 (100%)
 - Personal user acceptance test (100%)
 - SPR-5 charts development (100%)

Jordan Van Patten:

- Total Hours: 3 hour, 10 minutes
- Total Meetings Attended: (9/11)
- Tasks Completed:
 - Review all SPR-5 document requirements (100%)
 - V&V documents for SPR-5 (100%)

Kole Davis*:

- Total Hours: 2hrs 30 minutes
- Total Meetings Attended: (9/11)
- Tasks Completed:
 - Review all SPR-5 document requirements (100%)
 - Ishikawa diagrams (100%)
 - User acceptance test report (100%)

*Jordan and Kole joined the team in sprint 2 and 5 respectively

Plans

SPR-5 Risk Management Plan

Summary: This is the sprint five risk management plan.

ID	Category	Description	Consequence	Probability	Impact	Risk Minimalization plan	Contingency plan
1	Metrics	missing certain information for metric reports	Inaccuracy in metric reports	High	Medium	Not deleting any information, making sure to keep records of all documents	Get the most out of the information that has been gathered
2	Program	Final touchups on the project cause it to stop working	Program couldn't be delivered to customer	Low	High	Keeping records of previous versions of the code so we can always get back in case something does happen	Use the newest version that still works
3	Shareholder	The shareholder requesting changes at the last minute	May not have enough time to implement the changes that the shareholder requests	Medium	High	Constant communication with the shareholder, understanding what they want as soon as possible	depending on what the shareholder requests, may have to save it for a later release
4	Change orders	Unable to complete the change orders that the shareholder has approved	Missing agreed upon functionality	Low	High	Proper cost estimation for the change orders, and time management to make sure it gets done	Let the shareholder know that we will have to save the change order for the next release
5	Submission	One or multiple team members either forgets to submit the final document or turns in the incorrect link	Penalties for incorrect submission	Low	High	Everyone meets a final time before submission and everyone confirms that they have the proper link	If no link was sent, send one and have a meeting to see what went wrong

Updated Class UML

Downloaded from <http://ajph.org/> on November 10, 2014

Functional/Non-functional testing

Summary: For this test we went over each of the function and non-functional requirements and made sure that the program met each of them.

We have verified the contents of this document using our requirements specification documentation

Functional Requirements:

- Database Merging: Recognizes empty data fields based on both JSON and CSV databases and notifies the user of the missing data.
- Add employee (Admin access): inputs all required fields and constructs a new employee object to be added to an internal database.
- Edit employee (General & admin access): Lists selected employee's editable information fields (First name, Last name, Address, Office phone, Personal phone, Bank info, Office email, Personal email, etc.)
- Search employee by last name, ID, Title, or Classification (General & admin access): When user selects search parameter from search combo box, they are then able to start searching for employees. If a user enters numbers, the program will either display employees based on classification or ID number. Otherwise, the program will display employees based on job title or last name.
- View employee (General access): Displays selected employee's information in read-only mode and allows user to search for other employees' read-only information
- View employee (Admin access): Behaves similar to general access with the addition to deactivate an employee and run payroll.
- Deactivate employee (Admin access): Allows admin to input ID of employee they wish to deactivate. Once this occurs, a confirmation message appears, along with the selected employee's information to confirm the admin wishes to deactivate the selected employee
- Pay report (Admin access): Program Generates pay report including delivery information for each active employee.
- Garbage proof entries (General & admin access): Program ensures each data field contains valid information.
- Warn user of empty data fields (Admin access): When admin adds/edits an employee, issue a warning if any data fields are left empty or incomplete.

Non-Functional Requirements:

- The Program runs on Windows 10.
- Intuitive GUI: Mouse over input boxes to see what information is required, help buttons on each page.
- User Manual: Provides information on how to use and navigate the current page in the program (can be accessed by clicking the "Help" button).

- Readme.txt: A short description of what data the code contains.
- Simple “just download” installation that runs: Download software and then the user is good to go
- Employee can view all personal fields: When employee logs in with his/her ID, that ID’s corresponding information is pulled up
- Bug free: Software will go through extensive testing to minimize/eliminate as many bugs as possible using Pytest and or Black, or any undiscovered software
- Requirements for employee information: First name, Last name, Address(use separated fields), Office phone, Personal phone, Emp ID(Specific length, only numbers), Pay type(commission, hourly, salary), D.O.B, SS#(Specific length, only numbers), Start date, End date, Bank Info(if Direct Deposit), Permission level, Title Dept., Office email, Personal email

Metrics

Defect Analysis

Document Defects

Summary: This analysis goes over each of the defects that have appeared in our sprint documents and explain how it could have been avoided, how we could have caught it earlier, and how we could have caught it before the shareholder.

SPR-1

- Incorrect link location
 - Could have been avoided by directly checking with the shareholder prior to submission.
 - We could have caught this earlier and beat the shareholder to the discovery by finishing the document quicker so we could double-check the link.
- Unspecified prototype candidate
 - Could have been avoided by being more specific about which prototype we were leaning toward.
 - We could have caught this earlier and beat the shareholder to the discovery by more thoroughly reviewing the requirements and rubric with him.
- Document Versioning missing
 - Could have been avoided by more thoroughly reviewing the sprint requirements.

- Could have been caught earlier and beat the shareholder to the discovery by finishing the document early enough to review it with the shareholder.
- Coding standards missing
 - Could have been avoided by more thoroughly reviewing the sprint requirements.
 - Could have been caught earlier and beat the shareholder to the discovery by finishing the document early enough to review it with the shareholder.
- Meeting schedule missing
 - Could have been avoided by more thoroughly reviewing the sprint requirements.
 - Could have been caught earlier and beat the shareholder to the discovery by finishing the document early enough to review it with the shareholder.
- V&V document missing
 - Could have been avoided by more thoroughly reviewing the sprint requirements.
 - Could have been caught earlier and beat the shareholder to the discovery by finishing the document early enough to review it with the shareholder.
- Requirements specifications missing
 - Could have been avoided by more thoroughly reviewing the sprint requirements.
 - Could have been caught earlier and beat the shareholder to the discovery by finishing the document early enough to review it with the shareholder.

SPR-2

V&V document missing

- We could have avoided this by checking with our V&V teammate and ensuring they had completed all their assigned tasks.
- We could have caught this earlier by checking in on our V&V teammate more frequently.
- Submitted incorrectly
- Could have been avoided by directly checking with the shareholder prior to submission.
- We could have caught this earlier and beat the shareholder to the discovery by finishing the document quicker so we could double-check the link.

SPR-3

- GUI images missing
 - We could have avoided this by remembering to include the images we had in our user manual in the sprint document itself.
 - We could have caught this earlier by checking our sprint document more frequently for missing required information.
- Backlogs missing
 - Could have been avoided by keeping better track of the tasks we did and did not complete in previous sprints.
 - We could have caught this defect earlier and beat the shareholder to the discovery by more thoroughly reviewing the sprint requirements to know we needed to be keeping track of backlogs.
- Code doesn't run

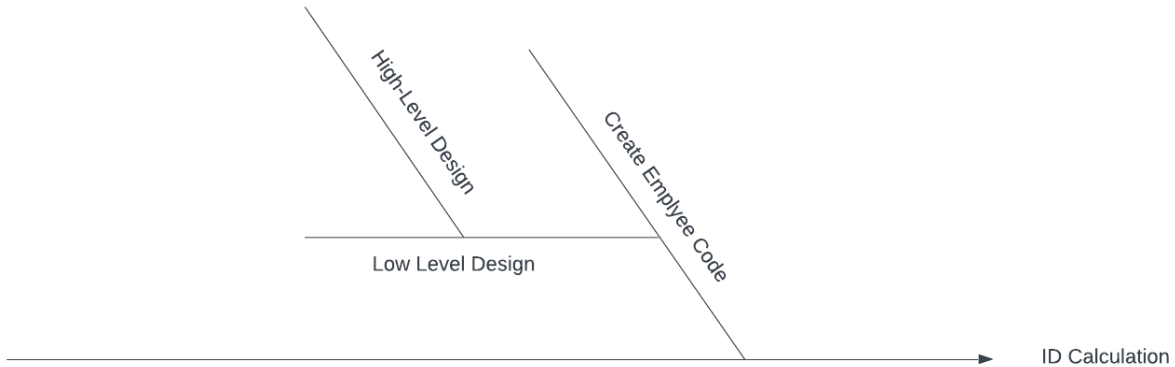
- This defect could have been avoided by more thoroughly testing the code on a virgin computer.
- We could have caught this defect earlier and beat the shareholder to the discovery by making sure to test the code on the platform designated in the requirements specification.
- Insufficient info in readme.txt
 - If we would have assigned the task at the beginning of the sprint, this file could have been produced incrementally through the sprint.
 - We included the info in the readme.txt very late. Finishing the development portion of this sprint earlier would have allowed us to add more to the file.
 - To summarize, early assignment and earlier completion of the development stage would have allowed more info to be in the readme.txt file to satisfy the shareholder.

SPR-4

- Missing backlogs
 - This defect could have been avoided if we had kept track of early project tasks as we progressed through sprints.
 - We hadn't left behind too many undone tasks so a backlog still could have existed, it just would have been empty. Perhaps having more team members would have allowed us to catch this defect.
 - Again, more people on the project would have allowed us to have more attention on this area of the project.
- Missing updated class diagram
 - This could have been avoided from V&V of the requirements and current contents of our program and sprint documents.
 - It could have been found earlier if our V&V had taken place earlier in the sprint.
 - Progressive V&V of progress and requirement comparison would have brought this to our attention before the shareholder discovered it.
- Missing User Acceptance Testing
 - This could have been avoided from V&V of the requirements and current contents of our program and sprint documents.
 - It could have been found earlier if our V&V had taken place earlier in the sprint.
 - Progressive V&V of progress and requirement comparison would have brought this to our attention before the shareholder discovered it.
- Missing Functional/Non-functional Testing
 - Our understanding of the requirements of this sprint and of the Functional/Non-functional Testing was not the same as the expectation from the shareholder. Further review with the Shareholder would have brought this to our attention so it could have been included.
 - Meeting early with the shareholder could have increased our likelihood of understanding our shareholders expectations.

- Early detection would have kept the shareholder from seeing that this report was missing.

Ishikawa Diagram



Software Metrics

Team Meetings

Summary: This table is designed to gather all of the information that has been gathered from the Team meetings, such as how many meetings we had, how often each member attended said meetings, and how many tasks were completed.

Meeting #	Sprint #	Date	# of attendees	Length of meeting	tasks completed	Tasks remaining	New tasks added
1	SPR-1	January 20, 2022	4/4	1hr 12min	n/a	n/a	n/a
2	SPR-1	January 21, 2022	4/4	41min	n/a	n/a	n/a
3	SPR-2	January 24, 2022	4/4	55min	4/15	11	15
4	SPR-2	January 28, 2022	5/5	1hr 5min	5/14	9	3
5	SPR-2	January 31, 2022	5/5	1hr 15min	8/16	8	7
6	SPR-2	February 4, 2022	4/5	55min	4/9	5	1
7	SPR-2	February 7, 2022	5/5	40min	5/8	3	3
8	SPR-3	February 11, 2022	4/5	1hr 4min	3/9	5	6
9	SPR-3	February 14, 2022	5/5	1hr 3min	5/14	9	9
10	SPR-3	February 18, 2022	4/5	1hr 8min	7/19	12	10

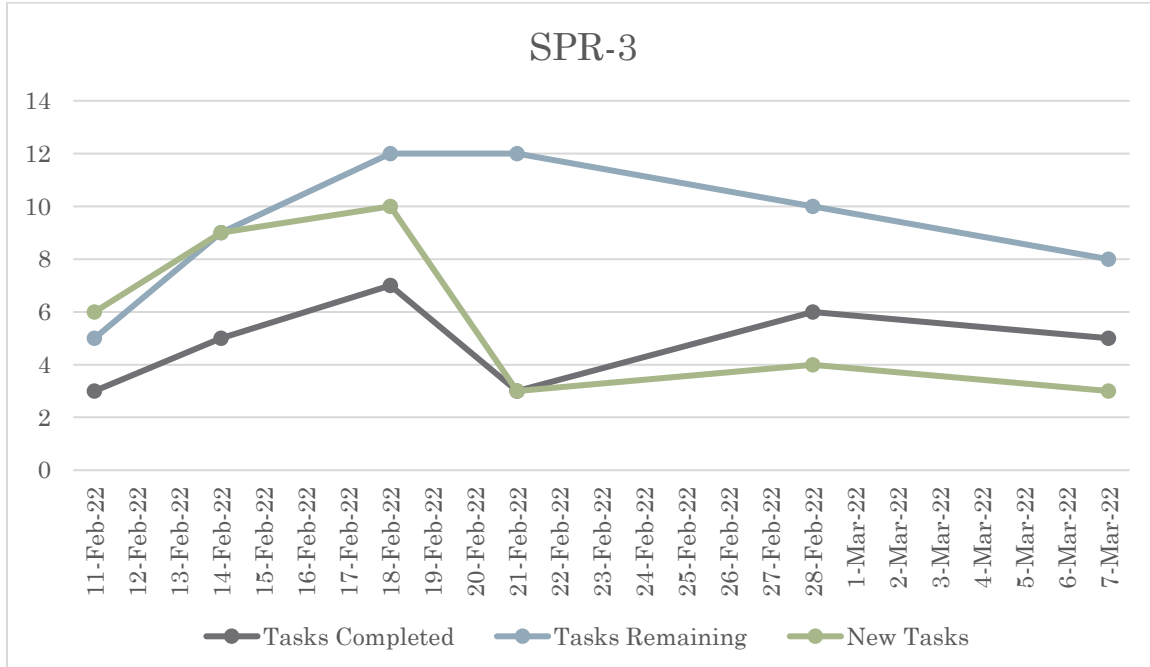
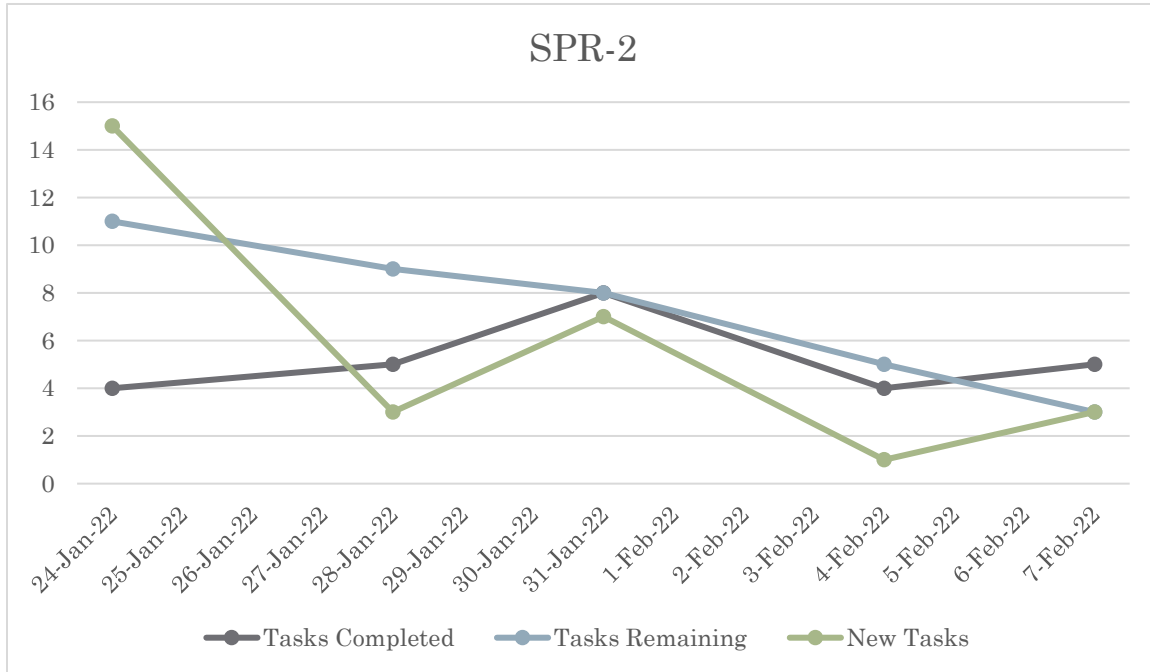
11	SPR-3	February 21, 2022	4/5	1hr 6min	3/15	12	3
12	SPR-3	February 28, 2022	5/5	1hr 4min	6/16	10	4
13	SPR-3	March 7, 2022	6/6	20min	5/13	8	3
14	SPR-4	March 14, 2022	5/6	1hr 4min	5/19	14	11
15	SPR-4	March 16, 2022	5/6	44min	10/21	11	7
16	SPR-4	March 18, 2022	6/6	45min	14/18	4	7

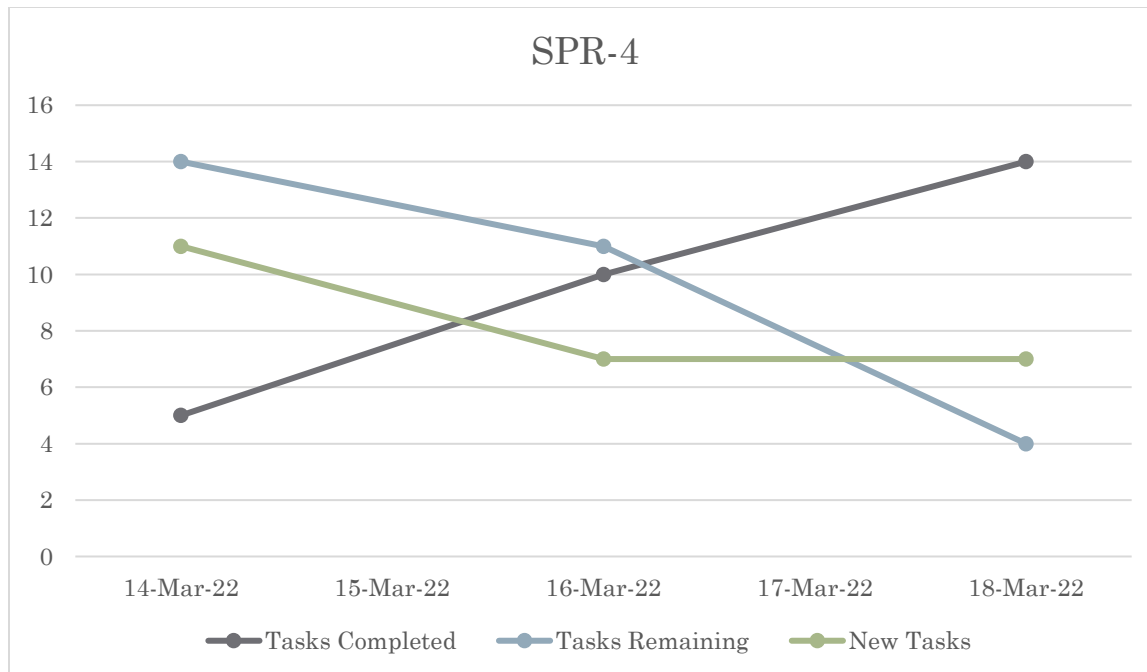
Report Analysis:

- What we did
 - We had two meetings a week on Monday and Friday for roughly one hour per meeting. We eventually added in a short meeting on most Wednesdays, we also didn't have many meetings/brief meetings over holidays and spring break.
 - Meetings tasks were not being tracked for the first sprint
- What we keep
 - Three meetings a week seems to have worked pretty well, and for projects of similar size we would keep the schedule we currently have. An hour length has been enough time for everyone to say everything that they need to, so we would keep that as well.
- What we would do better
 - Meeting more often over long holidays
 - Keep track of the short meetings as well as the normal ones
 - Keep track of tasks starting with the first sprint

Tasks Completed to Tasks Remaining by Sprint

Summary: These charts are designed to show how many tasks were completed by each meeting where they could be reported to the overall tasks remaining. It also shows the amount of new tasks that were being added which would affect the overall tasks remaining.





Report Analysis

- What we did
 - Each sprint we would figure out each of us would need to do and then we would set out to do it
 - As new things were discovered that needed to be done, we would bring it up in a one of our meetings and decide who got to take care of it
 - We would report when we finished it through the group chat and how long it took us to do it
- What we would keep
 - The group chat worked out really well and we would likely keep it
 - Everyone figuring out what they needed to get done based on their role
- What we would do better
 - We have already changed this, keeping better tabs on what everyone is doing and how far along they are on their tasks
 - Try to figure out all of the tasks for the sprint sooner

Assert Testing

Summary: This table is designed to look at groups of assert tests and states what they test for, how many tests they have total, what different inputs they are testing for, who ran the tests, and whether or not they all passed.

Activity	Metric Measure	number of tests	Test Inputs	Source	Current State
Tests that employee has the correct ID, classification and pay	Assert	3	ID numbers: 2, 3, 7	Jordan	Passed
searches for a user that should be in the database	Assert	5	ID numbers: 5, 1, 100 Last name: johnasdk, LastNameChanged	Jordan	Passed
Tests that fields are registered as valid when valid results are input	Assert	1	Employee object: With all valid field	Jordan	Passed
Tests that fields are registered as invalid when invalid results are input	Assert	1	Employee object: Missing one field	Jordan	Passed
Tests that the information shown on the general users viewpage matches their informatin in the database	Assert	4	ID numbers: 8, 2, 3, 4	Jordan	Passed
Tests that the information shown on the admin users viewpage matches their information in the database	Assert	2	ID numbers: 8, 6	Jordan	Passed
Tests that using the correct username and ID does NOT trigger the "Incorrect password or ID" message	Assert	7	ID numbers: 8, 2, 3, 4, 5, 6, 7	Jordan	Passed
Tests that using the incorrect username but correct password does trigger the "Incorrect password or ID" message	Assert	8	ID numbers: Jim, tony, 123, 57, gfad, wrong, almost, so close	Jordan	Passed
tests that using the incorrect password but correct username does	Assert	11	Passwords: tests, test!, tesT, Test, nope, tESt, password, letMeIn,	Jordan	Passed

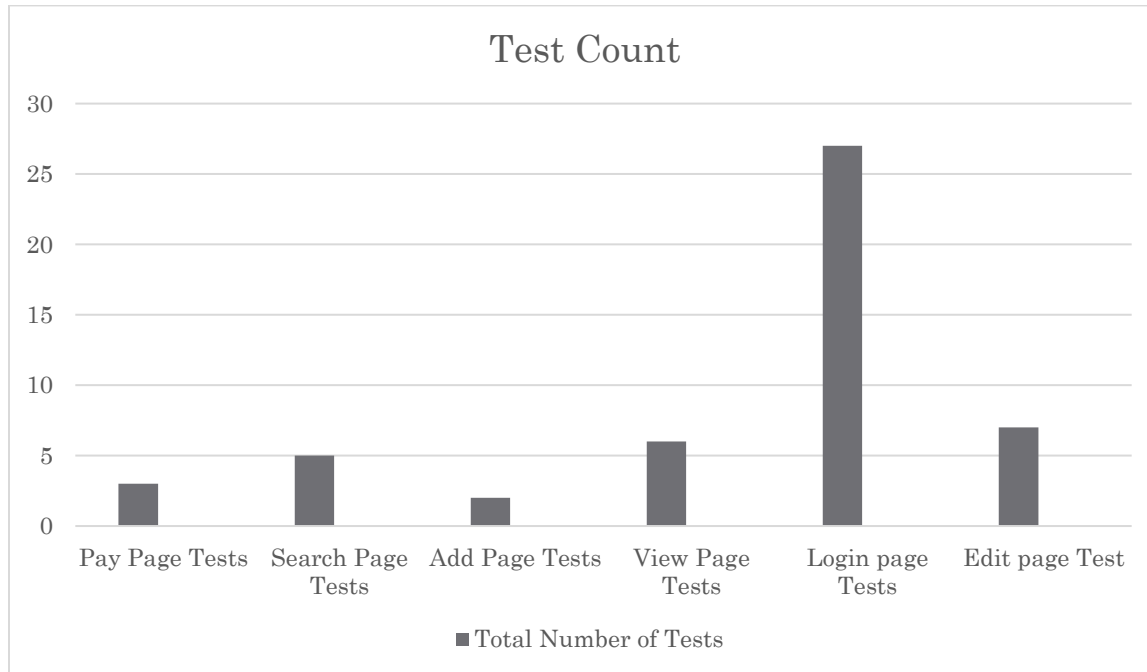
trigger the "Incorrect password or ID" message			please, prettyplease, 123456		
Tests that all fields are valid when valid inputs are given	Assert	6	ID numbers: 8, 2, 3, 4, 6	Jordan	Passed
test_edit_invalid_variable	Assert	1	Employee object: Invalid fields	Jordan	Passed

Report Analysis

- What we did
 - Extensive testing was we programmed so that our code was more likely to pass all of the assertion tests
 - Write assertion tests for each page
 - Test for a variety of inputs
- What we would keep
 - Testing the code as you write it, that way you can catch and fix a bug as soon as possible
 - Using pytest to do assertion testing
- What we would do better
 - Write tests for more unique scenarios, such as negative numbers, special characters like @, and decimal numbers

Test Count by GUI Page

Summary: This chart shows the number of tests that were written for each page. We increased login tests to make sure that the information would be correct. For the add page each individual test involved a lot of fields being tested at once. The other pages didn't need as many fields to be checked and had less that could be incorrect for each test, so not as many tests were required.



Report Analysis

- What we did
 - Wrote tests to check for what a user might be inputting and make sure that it would run as we the developers would expect it to run.
- What we would keep
 - We would keep all of the tests that we have
- What we would do better
 - Add boundary tests

Document Defects

Summary: This table is designed to show the document defects that arose throughout the sprints in the project and the current status of said defects. Document defects are items that are missing from the sprint documents that are required for the project to match the requirement specifications.

SPR#	Document Defect Description	Status
SPR-1	Incorrect link location	
SPR-1	Unspecified prototype candidate	
SPR-1	Document Versioning missing	
SPR-1	Coding standards missing	
SPR-1	Meeting schedule missing	
SPR-1	V&V document missing	
SPR-1	Requirements specifications missing	
SPR-2	V&V document missing	
SPR-2	Submitted incorrectly	
SPR-3	GUI images missing	
SPR-3	Backlogs missing	
SPR-3	Code doesn't run	
SPR-3	No readme.txt	
SPR-4	Missing backlogs	
SPR-4	Missing updated class diagram	
SPR-4	Missing User Acceptance Testing	
SPR-4	Missing Functional/Non-functional Testing	

Report Analysis

- What we did
 - We each researched on our own what was required for each of our roles for the sprint
 - We came up with what we believed were the requirements for the sprint and tasks that need to be completed to meet those requirements in team meetings
- What we would keep
 - Individual research
 - Team discussions
- What would we do better
 - Contact the shareholder more often to help understand everything that is required to avoid creating defects

Program Bugs

Summary: This table is designed to show the bugs that have appeared in the code, what the bug caused, what was changed to fix the bug, when the bug was found and fixed, and whether the bug is currently fixed or not.

ID#	Bug Description	Change Description	Date Found	Date Fixed	Status
1	-The GUI currently functions, but the data being operated on (employee objects) is currently hard coded into the program. All new data, changes, etc. is lost when program closes.	-Add JSON text files as databases so we can add, edit, and retrieve records for employees, timecards, and receipts. -Add modules for the employee class that will add, edit, and retrieve records from the employees JSON file.	02/16/2022	03/02/2022	
2	-Tkinter message boxes aren't being imported when the program loads. (Only for development in windows)	-Include the specific library that applies to Tkinter libraries.	02/23/2022	02/23/2022	
3	-The GUI page for payroll did not have the run payroll or run csv file buttons functional.	-Add a CSV and handling module for the imported csv's that need to calculate the final pay for hourly, and compensated employees. -Add JSON text file for a database as a backup and add accessibility functions for this database in a module.	03/15/2022	03/18/2022	
4	-The ID was being keyed off of the number of employees in the database. With testing, and perhaps future functionality, deleting employees would cause conflicts with newly created employee ID's.	-Whenever a new employee is added, their Id is one plus the last employee in the database ID.	03/18/2022	03/19/2022	
5	-The user would have been unable to access the program as there was no user information provided when the program was loaded onto a new computer.	-Included a readme.txt file that included information regarding basic login for admin. From there, the user would be able to look up other employee information and create / edit employees as necessary.	03/20/22	03/20/2022	
6	-There was trouble for the shareholder originally with starting up the GUI. They were trying to launch the program from an	-There were some irrelevant files that existed in the next version of the program that could have caused more confusion. These were removed, and the file that the program needs to be launched from	03/12/2022	03/20/2022	

	incorrect file. File naming isn't explanatory.	was renamed to main.py for clarity. This was then mentioned in the readme.txt file.			
7	-There were issues caused by the validation function when it tried to update the field-specific error messages.	-The issues were caused by a logic error in the validation function attempting to alter label widgets that had not been created. The issue was solved by including a try/except block that checks for widget presence before alteration.	03/18/2022	03/18/2022	
8	-The error messages on all GUI pages were hardly noticeable.	-Solved by implementing a specific style with larger, more colorful text for error messages.	03/27/2022	03/27/2022	
9	-All fields were editable in the edit page, including things that realistically should not be changed, even by an admin (e.g. employee id, start date, etc)	-Solved by enabling read only mode on select entry boxes. -Admins can change everything except employee id, start date, end date, and employee status, while general permission employees editing themselves can change only personal information and not anything related to the company like their pay and job title.	03/03/2022	03/03/2022	
10	-There were issues attempting to set the entry boxes of the view page to "readonly" when in general view	-Issue was caused by a logic error in the method meant to set all entry boxes to "readonly". It would change the widgets created by the admin view, which included many not used for the general view. Issue was solved by creating a second list to be passed to the readonly method when not in admin view.	03/01/2022	03/01/2022	
11	-The shareholder did not approve of the ability to add timecards and receipts individually	-Solved by reverting the pay page of the GUI back to its initial state with only the ability to import the timecard and receipt csv files.	03/25/2022	04/10/2022	
12	-The search page on the GUI was not very user-friendly, requiring prior knowledge of the database to operate easily.	-Implemented a listbox widget below the search bar that shows all the employees in the database that match the current contents of the search bar.	03/27/2022	03/28/2022	
13	-There were issues moving from the search page to the view page of the searched employee when	-Issues were caused by the program attempting to alter widgets that had not been created (they are created only in admin view or when viewing oneself).	03/18/2022	03/18/2022	

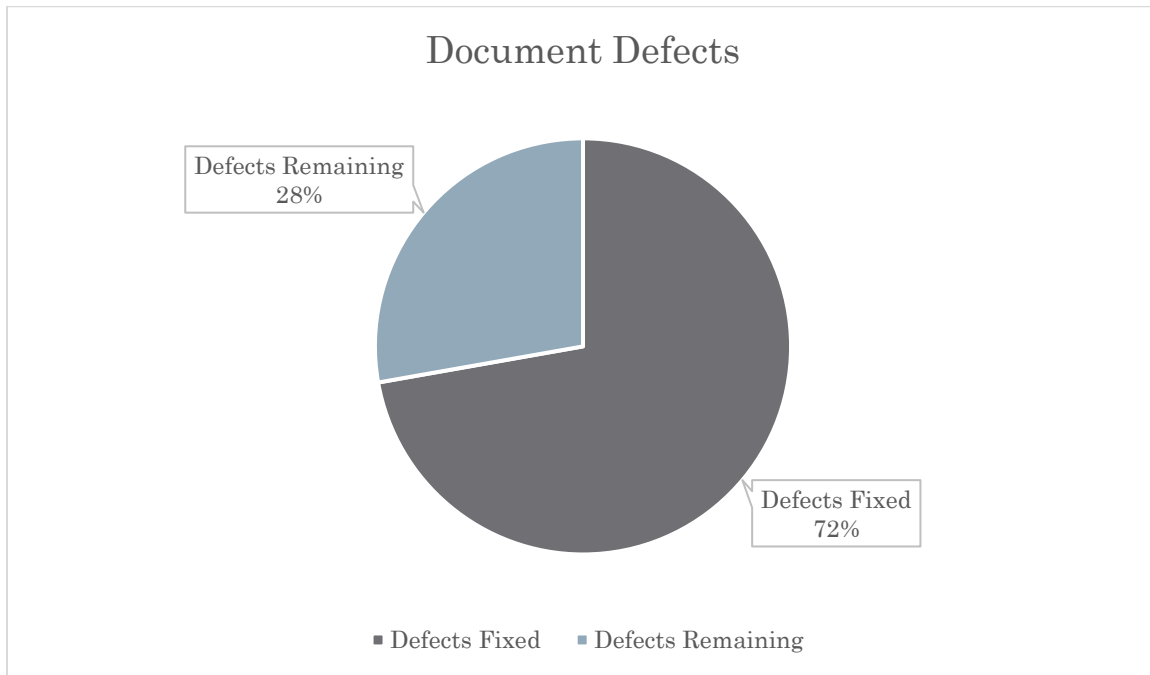
	logged in under general permission.	Solved the issues by placing the code that altered those widgets within an if statement that checks whether they exist or not first.			
14	-The pages of the GUI would not correctly refresh/update when switching between pages (e.g. old data on the add page would remain instead of being cleared).	-Created a new method for the Application class to be called with every page change that destroys and then recreates the pages to allow them to update properly.	02/13/2022	02/13/2022	
15	-There was trouble getting the entry boxes of the view page to correctly update their contents/configurations for inheritance with the add and edit pages.	Moved the code for creating the entry boxes out of the constructor for the View_Page class and into a method that can then be called by that class or any of its children as necessary.	02/13/2022	02/13/2022	

Report Analysis

- What we did
 - We fixed the bugs as we wrote the code, so the dates are not guaranteed to be 100% accurate but all of the bugs have been fixed because they were dealt with as they were found
- What we would keep
 - Keep fixing bugs as we code
- What we would do better
 - Document bugs as soon as they are found

Document Defects Remaining vs Defects Fixed

Summary: This chart shows the percent of defects that have been found in the document and have yet to be fixed.

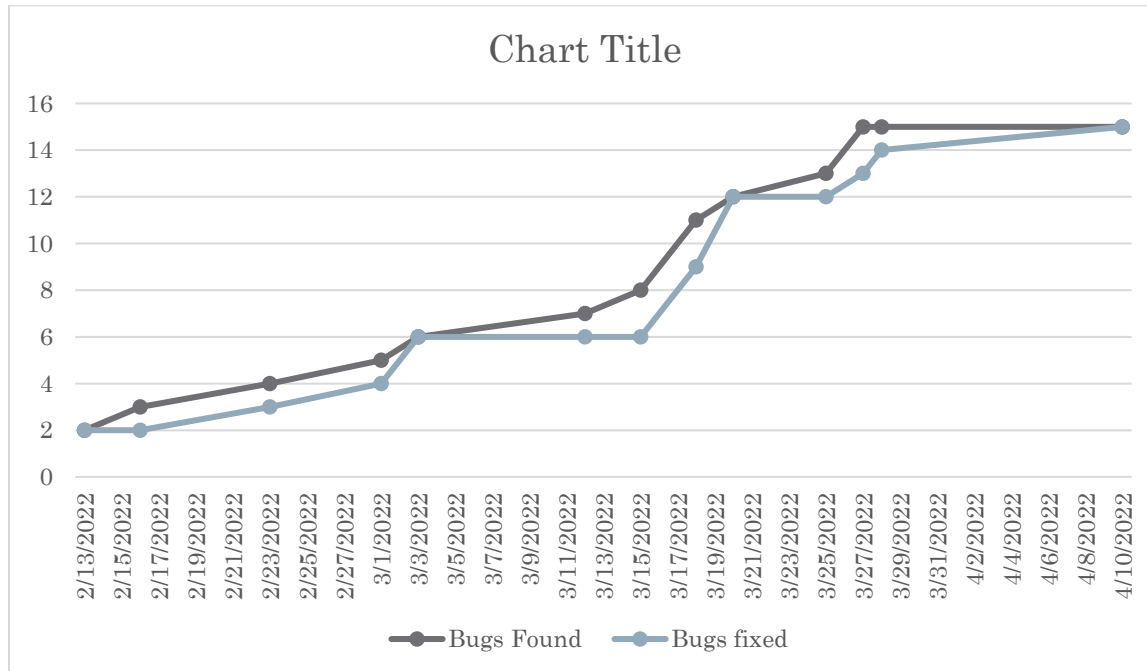


Report Analysis

- What we did
 - Fix defects from the previous sprint in the current sprint
 - Go through the comments left on sprint submission to see what defects are in the document
- What we would keep
 - Keep fixing defects in the sprint when they are found
 - Document defects found
- What we would do better
 - document dates of when defects are found and fixed

Bugs Found to Bugs Fixed

Summary: This chart shows the total number of bugs that were found in the program and when they were found, along with the total number of bugs fixed in the program and when they were fixed.



Report Analysis

- What we did
 - We fixed bugs as they were found which meant we had to go back and figure out when they were found and when they were fixed.
- What we would keep
 - Fixing bugs as we found them
- What we would change
 - Document bugs as they are found

Function Point Analysis

Summary: The function point analysis calculates a complexity value for each of the items that are being counted and uses them to get a sense of the overall complexity.

Category	Number		Low	Medium	High		Result
Inputs	50	x	3	4	6	=	150
Outputs	26	x	4	5	7	=	104
Inquires	9	x	3	4	6	=	27
Internal Files	8	x	7	10	15	=	56
External Files	0	x	5	7	10	=	0

Total = 337

Importance	0
Irrelevant	1
Minor	2
Moderate	3
Average	4
Significant	5
Essential	6

Factor	Rating
Data communication	2
Distributed data processing	4
Heavily used configuration	1
Transaction rate	1
Online data entry	0
End user efficiency	0
Online update	0
Complex processing	0
Reusability	5
Installation ease	5
Operational ease	5
Multiple sites	0
Facilitate change	3
Total (CAV)	26

Calculate Adjusted FP

$$FP = (337) * (0.65 + 0.01 * 26) = 307$$

Alpha/User Acceptance Test

Bugs

Summary: We combined all of our individual bug tests into one larger report. This is the Alpha portion.

- When we try to search for an employee, the page header shifts to the left once the search box gains keyboard focus and then shifts back to its original location once it is no longer in focus.
- When we hover our mouse over the search parameter combo box, the edit cursor appears, which makes no sense if I'm selecting an item from the list rather than typing something into the entry (My suggestion would be to make it so the mouse doesn't change when hovering over the search parameter combo box in order for users to not be caught off guard by the edit cursor when they're supposed to be selecting a list item).
- When we begin searching for an employee and wish to go back to change my search parameter, both the search list box and the search parameter combo box are stacked on top of one another, and this doesn't look very clean.
- Deactivated employees are still included in the payroll, and get paid out. A deactivated employee shouldn't be paid out at all.
- Can pay someone negative amount. This should be an easy fix with a simple entry validation.
- Employee ID 7 isn't on the payroll
- After trying to login with an incorrect ID and then immediately logging in with a proper ID, the "Incorrect ID" flashes whenever I change tabs. Same if I type in the incorrect password.
- Many fields can be left blank. From my testing, most of the fields that are still required on the add new employee don't necessarily need to be filled out just yet. An employee may not have an office phone set up yet, not have bank info yet, not have been given a title, etc.
- If we deactivate an employee, there should be a way to reactivate them as well.
- We need to develop a read me txt file for the program.

Functional or Design Improvements

Summary: This is the user acceptance portion. Here we focused on more functional and design improvements.

- The employee classification field label would look a lot cleaner without the numerical classification options in the parentheses (My suggestion would be to make this entry field a combo box, which would prevent users from entering wrong information into this field).
- When we select an employee from the search drop-down list, the search box is filled with the employee's information, but the box itself is too small to fit all the information.

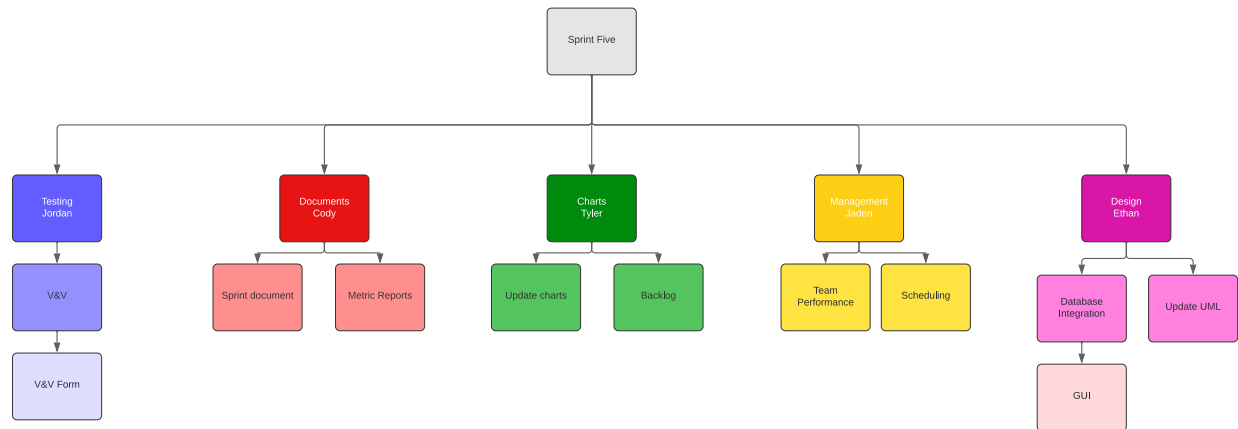
- Sometimes it can be difficult knowing if we have typed in all the correct employee information into each data field before attempting to change the employee's information and getting a message saying the information is incorrect (My suggestion is instant validation).
- The program needs a theme.
- Certain fields like pay method, classification, and permission level would be more user friendly and easily validated with dropdowns.
- The alerts under each field that is required or needs attention after validation should be styled red.
- Pop-up at bottom of screen that says not all fields are valid probably would be more visible at the top of the screen.
- Have a filter for the pay type classification so that you cannot enter the other pay type fields to avoid messing up payroll.
- Create notifications of some sort to let the user know that they had successfully processes timecards, receipts, and printed out payroll.
- Add a search list that can be dynamically filtered by the search keys entered into the search bar.
- We may want to section off our view window from our nav bar on the left. Help to be more user friendly and guide the eyes of the user better. Ex.

Nav	View

Charts/Forms

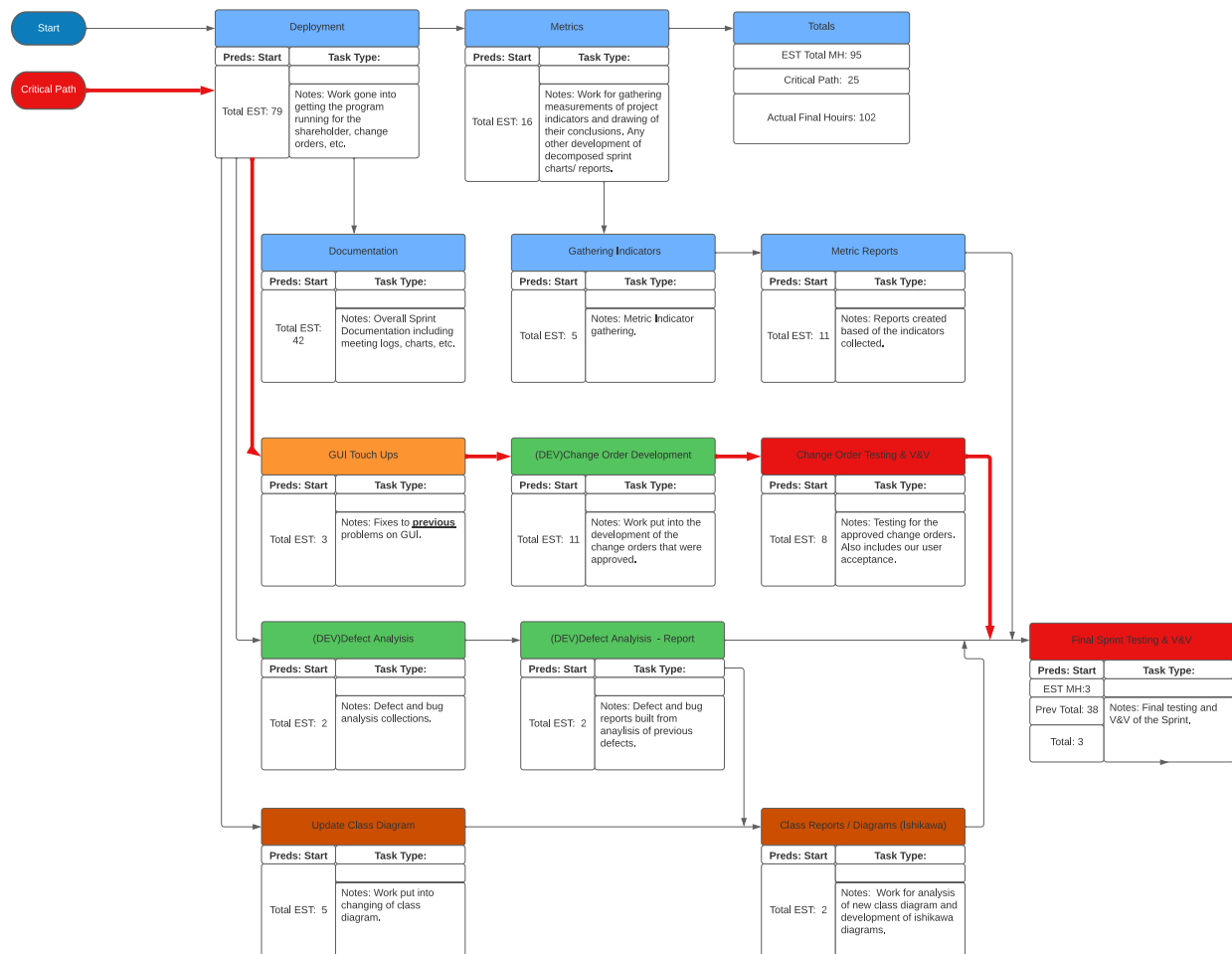
SPR-5 Work Breakdown Structure

Chart



SPR-5 Pert Chart

Chart



Meeting Logs

Meeting Log#17

Meeting Information

- Team #: T2-002
- Meeting log #: 17
- Current Sprint: SPR5
- Date: March 28, 2022
- Time: 7:00pm – 8:00pm (MT)
- Location: MS Teams ([Watch video here](#))
- Attendees: Jaden Albrecht, Tyler Deschamps, Jordan Van Patten, Cody Strange, Kole Davis
- Next team meeting scheduled for: March 30, 2022, 3:45pm (MT)

Progress From Previous Meeting

- Ethan Taylor:
 - Defects analysis (100%)
- Jaden Albrecht:
 - Personal user acceptance test (100%)
- Tyler Deschamps:
 - Defects analysis (100%)

Topics Discussed

- Ishikawa diagrams
- User acceptance testing
- Functional/non-functional requirements testing
- Function-point analysis
- Backlogging for SPR5
- Team performance review

Obstacles Encountered

- No obstacles

Finished Items

- Ethan and Tyler have gone through and put together a list of defects found during the implementation of our final project

Unfinished Items

- SPR-5 document
- Metric reports
- Team performance review

- V&V documents for SPR-5
- Defect analysis/report
- Functional/nonfunctional requirements test
- Updated UML diagram
- WBS for SPR-5
- Backlogs for SPR-5
- Ishikawa diagram
- Pert/Gantt/burn-down charts for SPR-5
- Implementation of approved change order requests from previous sprints
- Usability test
- User acceptance test report

Tasks Until Next Meeting

- User acceptance test report
- Ishikawa diagram(s)
- WBS for SPR5
- Pert/Gantt/burn-down chart development for SPR5
- Functional/non-functional tests

Notes

- Ethan was sick and couldn't attend this meeting

Meeting Log#18

Meeting Information

- Team #: T2-002
- Meeting log #: 18
- Current Sprint: SPR5
- Date: March 30, 2022
- Time: 3:45pm – 4:00pm (MT)
- Location: In-class meeting
- Attendees: Jaden Albrecht, Tyler Deschamps, Jordan Van Patten, Cody Strange, Kole Davis
- Next team meeting scheduled for: April 1, 2022, 7:00pm (MT)

Progress From Previous Meeting

- Jaden Albrecht:
 - Functional/non-functional requirements testing (20%)
- Tyler Deschamps:
 - Defects research

Topics Discussed

- Ishikawa diagrams
- User acceptance testing
- Functional/non-functional requirements testing
- Function-point analysis
- Backlogging for SPR5
- Team performance review

Obstacles Encountered

- No obstacles

Finished Items

- No items finished

Unfinished Items

- SPR-5 document
- Metric reports
- Team performance review
- V&V documents for SPR-5
- Defect analysis/report
- Functional/nonfunctional requirements test
- Updated UML diagram
- WBS for SPR-5
- Backlogs for SPR-5
- Ishikawa diagram
- Pert/Gantt/burn-down charts for SPR-5
- Implementation of approved change order requests from previous sprints
- Usability test
- User acceptance test report

Tasks Until Next Meeting

- User acceptance test report
- Ishikawa diagram(s)
- WBS for SPR5
- Pert/Gantt/burn-down chart development for SPR5
- Functional/non-functional tests

Notes

- No additional notes

Meeting Log#19

Meeting Information

- Team #: T2-002
- Meeting log #: 19
- Current Sprint: SPR5
- Date: April 1, 2022
- Time: 7:00pm – 7:30pm (MT)
- Location: MS Teams ([watch video here](#))
- Attendees: Ethan Taylor, Jaden Albrecht, Tyler Deschamps, Cody Strange, Kole Davis
- Next team meeting scheduled for: April 4, 2022, 7:00pm (MT)

Progress From Previous Meeting

Cody Strange:

- Assert test report (100%)
- Defect list (100%)

Topics Discussed

- Ishikawa diagrams
- User acceptance testing

- Functional/non-functional requirements testing
- Function-point analysis
- Backlogging for SPR5
- Team performance review

Obstacles Encountered

- No obstacles

Finished Items

- Defects report
- Assertion test report

Unfinished Items

- SPR-5 document
- Metric reports
- Team performance review
- V&V documents for SPR-5
- Functional/nonfunctional requirements test
- Updated UML diagram
- WBS for SPR-5
- Backlogs for SPR-5
- Ishikawa diagram
- Pert/Gantt/burn-down charts for SPR-5
- Implementation of approved change order requests from previous sprints
- Usability test
- User acceptance test report

Tasks Until Next Meeting

- User acceptance test report
- Ishikawa diagram(s)
- WBS for SPR5
- Pert/Gantt/burn-down chart development for SPR5
- Functional/non-functional tests
- Update UML class diagram

Notes

- No additional notes

Meeting Log#20

Meeting Information

- Team #: T2-002
- Meeting log #: 20
- Current Sprint: SPR5
- Date: April 4, 2022

- Time: 7:00pm – 7:15pm (MT)
- Location: MS Teams ([Watch video here](#))
- Attendees: Jaden Albrecht, Tyler Deschamps, Jordan Van Patten, Cody Strange, Kole Davis
- Next team meeting scheduled for: April 6, 2022, 3:45pm (MT)

Progress From Previous Meeting

- No tasks completed

Topics Discussed

- Ishikawa diagrams
- User acceptance testing
- Functional/non-functional requirements testing
- Function-point analysis
- Backlogging for SPR5
- Team performance review
- Updating UML diagrams

Obstacles Encountered

- No obstacles

Finished Items

- No new tasks have been completed from previous meeting

Unfinished Items

- SPR-5 document
- Metric reports
- Team performance review
- V&V documents for SPR-5
- Defect analysis/report
- Functional/nonfunctional requirements test
- Updated UML diagram
- WBS for SPR-5
- Backlogs for SPR-5
- Ishikawa diagram
- Pert/Gantt/burn-down charts for SPR-5
- Implementation of approved change order requests from previous sprints
- Usability test
- User acceptance test report

Tasks Until Next Meeting

- User acceptance test report
- Ishikawa diagram(s)
- WBS for SPR5
- Pert/Gantt/burn-down chart development for SPR5

- Functional/non-functional tests

Notes

- Ethan was sick and couldn't attend this meeting

Meeting Log#21

Meeting Information

Team #: T2-002

- Meeting log #: 21
- Current Sprint: SPR5
- Date: April 6, 2022
- Time: 3:45pm – 4:00pm (MT)
- Location: In-class meeting
- Attendees: Jaden Albrecht, Tyler Deschamps, Jordan Van Patten, Cody Strange, Kole Davis
- Next team meeting scheduled for: April 11, 2022, 7:00pm (MT)

Progress From Previous Meeting

- Ethan Taylor:
 - Updated UML diagrams (100%)
 - Integrate old database (100%)
- Jaden Albrecht:
 - Functional/non-functional requirements testing (20%)
- Cody Strange:
 - SPR-5 document (90%)
- Tyler Deschamps:
 - Backlogs for SPR-5 (40%)
- Kole Davis:
 - Ishikawa diagram (50%)

Topics Discussed

- User acceptance testing
- Functional/non-functional requirements testing
- Function-point analysis
- Backlogging for SPR5
- Team performance review

Obstacles Encountered

- No obstacles

Finished Items

- Updated UML diagrams
- Ishikawa diagrams

Unfinished Items

- SPR-5 document
- Metric reports
- Team performance review
- V&V documents for SPR-5
- Functional/nonfunctional requirements test
- WBS for SPR-5
- Backlogs for SPR-5
- Pert/Gantt/burn-down charts for SPR-5
- Implementation of approved change order requests from previous sprints
- Usability test
- User acceptance test report

Tasks Until Next Meeting

- User acceptance test report
- Ishikawa diagram(s)
- WBS for SPR5
- Pert/Gantt/burn-down chart development for SPR5
- Functional/non-functional tests
- Implementation of previous change order requests

Notes

- No additional notes

Meeting Log#22

Meeting Information

- Team #: T2-002
- Meeting log #: 22
- Current Sprint: SPR5
- Date: April 11, 2022
- Time: 7:00pm – 7:20pm (MT)
- Location: MS Teams ([watch video here](#))
- Attendees: Ethan Taylor, Jaden Albrecht, Tyler Deschamps, Jordan Van Patten, Cody Strange
- Next team meeting scheduled for: April13, 2022, 3:45pm (MT)

Progress From Previous Meeting

- Ethan Taylor:
 - Integrate CSV database (100%)
- Jaden Albrecht:
 - Functional/non-functional requirements testing (90%)

- Search parameters for search page (90%)
- Tyler Deschamps:
 - Backlogs for SPR-5 (80%)

Topics Discussed

- User acceptance testing
- Functional/non-functional requirements testing
- Function-point analysis
- Backlogging for SPR5
- Team performance review

Obstacles Encountered

- No obstacles

Finished Items

- CSV database integration

Unfinished Items

- SPR-5 document
- Metric reports
- Team performance review
- V&V documents for SPR-5
- Functional/nonfunctional requirements test
- WBS for SPR-5
- Backlogs for SPR-5
- Pert/Gantt/burn-down charts for SPR-5
- Implementation of approved change order requests from previous sprints
- Usability test
- User acceptance test report

Tasks Until Next Meeting

- User acceptance test report
- WBS for SPR5
- Pert/Gantt/burn-down chart development for SPR5
- Functional/non-functional tests
- Implementation of previous change order requests

Notes

- No additional notes

Meeting Log#23

Meeting Information

- Team #: T2-002
- Meeting log #: 23
- Current Sprint: SPR5

- Date: April 13, 2022
- Time: 3:45pm – 4:00pm (MT)
- Location: In-class meeting
- Attendees: Ethan Taylor, Jaden Albrecht, Tyler Deschamps, Jordan Van Patten, Cody Strange, Kole Davis
- Next team meeting scheduled for: April 15, 2022, 7:00pm (MT)

Progress From Previous Meeting

- Ethan Taylor:
 - Help button/user manual integration (50%)
- Jaden Albrecht:
 - Personal user acceptance (100%)
 - Add search parameters to search page (100%)
- Cody Strange:
 - SPR-5 risk management plan (100%)
- Tyler Deschamps:
 - Backlogs for SPR-5 (100%)
 - SPR-5 chart development (100%)

Topics Discussed

- Functional/non-functional requirements testing
- Function-point analysis
- Team performance review
- User manual integration

Obstacles Encountered

- No obstacles

Finished Items

- Risk management plan for SPR-5
- Pert/Gantt/burn-down charts for SPR-5

Unfinished Items

- SPR-5 document
- Metric reports
- Team performance review
- V&V documents for SPR-5
- Functional/nonfunctional requirements test
- WBS for SPR-5
- Backlogs for SPR-5
- Implementation of approved change order requests from previous sprints
- Usability test
- User acceptance test report

Tasks Until Next Meeting

- User acceptance test report

- WBS for SPR5
- Functional/non-functional tests
- Implementation of previous change order requests

Notes

- No additional notes

Meeting Log#24

Meeting Information

- Team #: T2-002
- Meeting log #: 24
- Current Sprint: SPR5
- Date: April 15, 2022
- Time: 7:00pm – 7:23pm (MT)
- Location: MS Teams ([watch video here](#))
- Attendees: Ethan Taylor, Jaden Albrecht, Tyler Deschamps, Cody Strange, Kole Davis
- Next team meeting scheduled for: April 18, 2022, 7:00pm (MT)

Progress From Previous Meeting

- Ethan Taylor:
 - Help button/user manual integration (100%)
- Cody Strange:
 - All metrics reports (100%)
 - Personal user acceptance test report (100%)

Topics Discussed

- Functional/non-functional requirements testing
- Function-point analysis
- Team performance review
- Metrics reports

Obstacles Encountered

- No obstacles

Finished Items

- Metrics reports
- Help button integration

Unfinished Items

- SPR-5 document
- Team performance review
- V&V documents for SPR-5
- Functional/nonfunctional requirements test
- WBS for SPR-5
- Backlogs for SPR-5

- User acceptance test report

Tasks Until Next Meeting

- User acceptance test report
- WBS for SPR5
- Functional/non-functional tests
- Implementation of previous change order requests
- Usability test
- User acceptance report

Notes

- No additional notes

Meeting Log#25

Meeting Information

- Team #: T2-002
- Meeting log #: 25
- Current Sprint: SPR5
- Date: April 18, 2022
- Time: 7:00pm – 11:30pm (MT)
- Location: MS Teams (watch video)
- Attendees: Ethan Taylor, Jaden Albrecht, Tyler Deschamps, Cody Strange, Kole Davis
- Next team meeting scheduled for: April 20, 2022, 7:00pm (MT)

Progress From Previous Meeting

- Ethan Taylor:
 - Fix search page change order (100%)
- Jaden Albrecht:
 - Team Performance Review (100%)
 - Functional/non-functional requirements document ((100%))
 - SPR-5 meeting logs (100%)
- Cody Strange:
 - All metrics reports (100%)
 - Personal user acceptance test report (100%)
- Tyler Deschamps:
 - SPR-5 chart development (100%)
- Jordan Van Patten:
 - Calculate total hours per sprint for all team members (100%)
 - V&V documents for SPR-5 (100%)
 - Quality assurance checklist (100%)
- Kole Davis:

- User acceptance report (100%)

Topics Discussed

- Functional/non-functional requirements testing
- Team performance review
- Metrics reports

Obstacles Encountered

- No obstacles

Finished Items

- Search page change order has been fully implemented
- SPR-5 document
- Team performance review
- V&V documents for SPR-5
- Functional/nonfunctional requirements test
- WBS for SPR-5
- Backlogs for SPR-5
- User acceptance test report

Unfinished Items

- No unfinished items

Tasks Until Next Meeting

- No tasks until next meeting

Notes

- No additional notes