CSE 360 Project Report Number 3

Team Tu37

Team Member Names:

1. Alma Babbitt
2. Ishan Yelnoorkar
3. Karryl Dumalag
4. Trevor Huss
5. Zachary Litwin

Table of Contents

1. Executive Summary

A data-driven company is a successful company. Proper business software is paramount to the unification of its developers and supervisors. The current system used to monitor development progress and defect detection is no longer suited for the ever-growing size of the company. A new system of systems is proposed to improve the quality and functionality of the agile development process as a whole. This new application will spearhead the issue of increasing team size while also holding information security and confidentiality of the utmost importance. The new system will lead to an increase in developer productivity and significantly fewer errors. Migrating from the traditional Scrum methodology to an enterprise-scaled agile development process will take some time and resources but will result in a greater outcome in performance. Our team understands the factors of risk at hand and will take that into account when completing this task. This new system will not only present a large change in fluidity but will also provide the flexibility to seamlessly implement new updates in the future.

Some of the new features of this improvement will include the following:

- A new, aesthetically pleasing look and feel.
- Efficient new ways to set plans and log time spent on projects.
- Flexibility to hire a larger number of developers and keep concurrent members updated.
- The ability to utilize EffortLogger information to provide insight into planning poker sessions
- Greater security of user data.
- Unification of cross-functional teams
- Better understanding of defects and why they occur.
- Adaptability and concurrency of software
- Compatibility with older versions of EffortLogger

Future problems and prospects will need to be addressed using this new system if a proper and quality result is to be achieved. The increasing complexity of problems in modern times calls for a streamlined and well-oiled team to accomplish them. Data drives the users, workers, and managers to push each other to succeed at the current task. The potential for improvement is limitless.

2. Customer Problem

The customer has a system called EffortLogger. It is a set of tools that allow the customer to track necessary project progress and do accounting effectively. However, since the creation of the system 20 years ago, the firm has been smaller, and the business requirements have grown significantly. The previous solution was an Excel-based solution created in Visual Basic that provided a set of Excel tabs that gave the appearance of an application interface. While the customer's leadership team is content with the current solution, a newer system is imminent to accommodate the changes in the business.

During the interview, we identified three stakeholders that contribute to the success of this project and the product. The three primary stakeholders are the customer, the user, and the supervisor. The customer dictates the needs of the firm and the vision for the newer system, while simultaneously the users dictate the efficiency of the product. Since the users are primarily dealing with the system to report performance data, they dictate what is required from the system. Finally, the supervisors are responsible for the firm's success, for which they dictate the tools required for them to supervise performance data efficiently and effectively.

Customer Problem

1. Employee privacy: The customer holds employee privacy as the highest priority. However, added pressures have expected the customer to ask employees to provide details about productivity rates, defect rates, and other relevant data to resize the firm and reduce contingencies. Employees are concerned about improper use of personal performance data and hence prefer the Excel-based Effort Logger because it allows them better control over their data. However, the older system prevents the customer from getting automated performance data and synchronous cross-functional team projects and prevents the customer from maintaining the anonymity of their employees.
2. Confidential information security: The leadership team is concerned about hackers accessing confidential information, including plans, schedules, budgets, and specifications. Therefore, security is of utmost importance. While planning and formulating project plans, timetables, and details, those who are involved in the management and formulation need to have a deep understanding of security risks and how to avoid them. Moreover, they also need to be adept at responding to security threats and the corresponding risk avoidance while retaining maintainability and integrity across all products.
3. Enterprise-scale support for agile and quality: The customer has increased its solution output by a factor of 20 since the Excel-based EffortLogger was used. The upkeep has become larger, causing the customer to create a QA (quality assurance) engineering organization. Moreover, the previous system lacks the architectural and design support to enable future efforts to improve planning and managing tasks. The customer wants to move from the traditional scrum to a customized process based on enterprise-scale agile. Traditional scrum prevents the customer from dealing with size, complexity, and quality assurance.

User problems

The users, in this case, are the customer's employees. Users are responsible for logging their efforts. After an interview, here is a summary of a user’s problem while using EffortLogger.

1. Planning poker sessions tend to be long because team members will often spend time accessing personal repositories for data related to user stories. The previous system did not allow for criteria-specific planning of poker sessions, or it became tedious to put up historical data relevant to specific criteria.
2. The older system has a slower way of assigning weight to passing items during the initial screening. This reduces the ability to detect the significance of story points during the planning process.
3. The Excel-based effort logger does not support planning poker; therefore, some estimates significantly deviate from the team’s consensus. It lacks the ability for a "quick look," which would allow the employees to look at the contribution of each item.
4. The user also expressed that the old system is not very aesthetically pleasing and could use the ability to generate stylized poker cards with the ability to share them.

Supervisor problems

1. The firm works towards effort predictability and defect avoidance, which require anonymity; however, the previous system could not anonymize employee-identifying information.

3. Concept of Operations

3.1 Planning Poker Operational Details

3.1.1 Setup

- Initiate the application and import historical data relevant to the current project
- Provide a distinct project name
- Input the user story and associated descriptive keywords

3.1.2 First Planning Poker Round

- The user listens to the user story description and reviews the list of historical projects produced
- Historical items that do not apply to the current estimation are removed
- The application is requested to generate an estimated number of story points based on the remaining historical data
- The user selects the appropriate physical planning poker card and once ready, the user shares the estimate with all the participants

3.1.3 Subsequent planning poker rounds

- Similar to the first round, include or exclude historical data
- Weights for each historical project's actual story points are adjusted, and other modifications are made
- Participants engage in discussion regarding the results of the previous round, emphasizing the reasons behind extremely high or low estimates
- Final adjustments are made, including weight modification
- A new estimate is requested, and a user selects the appropriate planning poker card

3.1.4 Finalization

- After reaching a consensus, the user makes final adjustments to the user story, keyword, and other additional notes. The results are saved for future use.

3.2 Effort Logger System Description

This section defines the current capabilities and features of the EffortLogger software and proposes additional features that fit the needs and requirements of the above section.

3.2.1 System Overview

EffortLogger was designed as a way to capture effort and defect information for up to ten people over a week. The projects, life cycles, deliverable types, and defect types are user-defined and can be configured anytime. Then they are used to organize what the programmers are doing with their time.

3.2.2 Functional Elements

- Screens
    - Acts as a canvas to place other elements.
    - Created by the EffortLogger developer and not configurable by the user.
    - The program shows one screen at a time.
- Labels:
    - Pieces of text were placed on the screen.
    - Gives context to the other elements.
- Buttons:
    - Do some action when clicked on.
- Dropdown menus:
    - Allows the user to choose between some defined options.
    - Used on the effort console to pick a project.
    - Options can be customized on the Definitions screen.
- Textboxes:
    - Creates a place for the user to enter information.
    - Used in the defect screen so the user can give a detailed description of the defect.
- Excel Grid:
    - Used in the Logs and Definitions screens

3.2.3 System Interfaces

The interface of EffortLogger consists of five different screens:

- The Effort console, which has:
    - A clock to track the time spent on activities.
    - A collection of drop-down menus to specify information about the activity currently being worked on
- The Effort Log Editor has
    - A menu to select the project and activity that needs to be edited.
    - Buttons to update, split, and delete activities
    - The split button splits the activity time in half and duplicates all other information.
    - Text boxes to change the date and time an activity has been done
- The defect console consists of:
    - A project and defect selection menu.
    - Textboxes to name and define information about a found defect.
    - A menu to give when the defect was found and when it was fixed.
    - Buttons to update and remove defects.
- The Logs Screen gives a place to view all logs, defects, and activities for all projects.
    - Logs are shown in a large spreadsheet.
    - Each project is allocated a thousand rows.
    - All information about each activity and defect is clearly defined in a series of columns and is the same as what is shown on the console screens.
- The Definitions Screen:

    - Is where the user customizes the projects, life cycles, effort categories, plans, deliverables, and other items to their needs.
    - Definitions are organized in a collection of charts.
    - The charts are placed on a spreadsheet.
    - Customization is done by changing the values in the chart.

3.3 Stakeholders' Needs and Requirements

3.3.1 Customer's needs and requirements

- The customer has a growing firm that requires an enhanced version of an existing solution they use to log employee productivity rates, defect rates, and other relevant information.
- The current solution prevents the customer from collecting employee performance data without collecting identifying information or without tediously anonymizing performance data.

3.3.2 User needs and requirements

- The user requires data-driven effort estimation, which will lead to reduced overtime, stress, and missed personal commitments.
- The user requires personal data privacy through anonymity.
- The Excel-based EffortLogger does not support planning poker; this disturbs workflow and reduces efficiency. The user requires a "quick look" to allow the user to look at the contribution of each item.

3.3.3 Supervisor needs and requirements

- The supervisors need a system that promotes anonymity by removing employee-identifying information while simultaneously being able to look at the contributions of the developer, engineer, and supervisor that have contributed toward a story point.

3.3.4 Mode of Operations

There are three major ways that EffortLogger is used.

- The general user will only need to log their activities and defects. When they start working on an activity, they open the effort console and click the "Start an Activity" button, then use the drop-down menus to pick the project, life cycle step, and effort category. The defect log follows a similar process.
- A manager will have to track a project's progress by going to the Logs screen and looking through the activity and defect logs.
- A project manager will have to use the Definitions Screen to define plans, deliverables, life cycle steps, and interruptions. As well as to create new projects and define their life cycles.

3.3.5 Proposed Capabilities

- Recreate the current program with all of its functions and features in the Java programming language. This will provide a larger and more stable code base to build upon and an easier expansion path to create new features.
- Devise a system that allows creating and managing multiple scrum teams for enterprise agile development support.
- Allow the manager of the teams to view activity and defect information about the team without the individual's information present so that coordinating and tracking the progress of multiple teams is possible while still allowing the individual user privacy.
- Add additional data for each activity and deliverable that assigns it a weight value that corresponds to the various timecards commonly used in planning poker.
- Use JavaFX to create a more navigable and pleasing user interface that will support the additional features.

3.4 Environmental Considerations

3.4.1 Physical Environment

- Ensure that EffortLogger is capable (if needed) of interacting with other existing systems or external hardware components.
- Guarantee transportation capabilities and security if software and its data are ever to be stored on a physical drive, i.e., physical shocks and vibrations during transit.
- The system should be designed to handle off-nominal conditions as much as possible.
    - Ensure that there are existing mechanisms and functions within the system to help it recover from external failures and errors such as power outages, hardware failures, network failures, security breaches, data corruption, and human error.

3.4.2 Support Environment

- Determine how maintenance is going to be carried out, i.e., scheduling, which sections to monitor, etc.
- Define what the future procedures will be regarding changing the system configuration, system updates, and data uploads to maintain security and minimize system operation disruptions.
- Consider and plan to address security vulnerabilities from system updates and how to apply patches promptly to maintain confidentiality and data integrity.
- Ensure that users of the system are trained and make ongoing user support available via user manuals, helpdesk, etc.
- Allow proactive maintenance by implementing both manual and automated performance management to continuously assess system performance and health.
- Plan for backup and recovery procedures to safeguard data and systems in case of off-nominal conditions.

3.5 Risks and Potential Issues

3.5.1 Project risks

Project schedule

- Essentially, that team understands the available resources and the allocation of resources before the start of the project.
- Make sure to allocate enough time to completely understand the customer's vision and requirements. Set aside 1-2 days to clear up confusion and bring up needed clarifications.
- Allocate time for risk assessment, security audits, and implementing security practices. Robust security measures are essential given the nature of the project requirements and design.
- Include time for training the development team on the transitioned custom agile process.
- Plan the balance between implementing essential features and crucial privacy concerns in the product.
- It is essential to provide sufficient time to allow the quality assurance team to apply rigorous testing and quality assurance activities.

Staffing support

- Ensure the project team has members with adequate security expertise to address confidential information and security concerns effectively.
- The privacy compliance team needed to emphasize and keep up with data privacy regulations and compliance during and after production.
- The quality assurance team needed to professionally test the quality of the system.
- Members with architectural and design expertise to address the lack of accessibility and design in the previous system.
- Members with design expertise need to create a scalable and maintainable solution for the long term.
- Agile coaches to aid in the transition to a customized agile process

Implementation approach

- Transition to a customized agile process:
  - We need to train employees in the new agile process. Provide available coach support, user guides, and workshop training.
  - The transition must be gradual for smoother operations.
  - All changes are documented and customized for the team’s needs.
- Security concerns:
  - Secure the development team’s safe coding practices, data encryption, access controls, and continuous monitoring.
  - Carefully plan to allow customer control over data while still enabling automated, secure data collection.
- Scalability concerns:
  - Ensure ease of maintenance for future updates. Provide a clear upgrade path after production.
  - Plan to use flexible and versatile technologies and modular design.
- Data migration between the old Excel-based model and the new EffortLogger system
- Quality assurance must ensure automated testing, continuous integration, and regular code reviews throughout the development process.

4. Received Requirements

4.1 EffortLogger Customer Need V2-0 Document V1-1

The clients who hired our team to develop EffortLogger V2.0 have expressed some concerns and requirements for the EffortLogger update. We have received input from the customer, supervisors/management, and employees. Below are the user stories depicting the stakeholders' requirements for EffortLogger V2.0

4.1.1. Title: Ensuring Employee Privacy and Anonymity

Story: As a member of the development team, I want to ensure that the EffortLogger V2.0 solution provides robust mechanisms for maintaining employee privacy and anonymity so that we can collect essential performance data without compromising personal information.

Acceptance Criteria:

1. The system must allow employees to choose what performance data is shared with their supervisors or managers.
2. Employees should have the option to remain anonymous while providing performance data.
3. The system should support professional role levels (e.g., programmer, software engineer) for categorizing and reporting performance data.
4. Data collection methods should be automated and secure to prevent unauthorized access to personal information.
5. The system should have a clear audit trail to track who accesses performance data and for what purposes.
6. Compliance with data privacy regulations (e.g., GDPR, CCPA) must be ensured in the design and implementation.
7. Employees should have full control over their personal information and the ability to revoke access at any time.

4.1.2. Title: Strengthening Data Security and Protection

Story: As a member of the leadership team, I want to ensure that EffortLogger V2.0 provides robust security measures to protect our confidential information from potential data breaches and unauthorized access to safeguard our competitive advantage.

Acceptance Criteria:

1. The system must implement robust encryption mechanisms for all data related to project plans, schedules, budgets, specifications, and technical details.
2. Access to confidential information must be role-based, with strict access controls and authentication requirements.
3. Real-time threat monitoring and detection mechanisms should be in place to respond to new threats promptly.

4. All critical work products, including Customer Need Assessments, Concepts of Operations, Storyboards, User Stories, Use Cases, and architectural artifacts, should be protected with encryption and access controls.
5. The system should maintain design integrity and traceability of all work products, ensuring they are not tampered with during the development lifecycle.
6. Regular security assessments and penetration testing should be conducted to identify vulnerabilities and address them proactively.
7. Compliance with industry standards for information security (e.g., ISO/IEC 27001) and legal requirements should be maintained.
8. In the event of a security breach, the system should have a response plan in place to minimize damage and ensure rapid recovery.

These user stories address the key requirements outlined in the document, focusing on employee privacy and confidentiality concerns. They provide a clear description of the desired functionality and criteria for success.

4.1 EffortLogger Supervisor Input 2023-08-11

4.2.1. Title: Protecting Employee Privacy while Maintaining Transparency

Story: As a member of the process improvement team, I want to ensure that EffortLogger V2.0 maintains a balance between individual privacy and transparency in data reporting so that we can continue data-driven decision-making and improvement efforts effectively.

Acceptance Criteria:

1. The system must anonymize all individual effort and defect reports before they are accessed by any team, project, program, or organizational analysis tool.
2. Identifying information about individuals (e.g., developer 1, engineer 3, supervisor 2) should be included in individual effort and defect reports for targeted improvement efforts and knowledge sharing.
3. When there are insufficient reports to ensure anonymity, access to the source data must not be provided, and summary data should not reveal individual identities.
4. The process flow of data from individuals to analysis tools must be transparent, and a clear explanation of how privacy is maintained should be provided in user-friendly terms.
5. An audit trail should be maintained to track access to individual-identifying information, ensuring accountability and compliance with privacy standards.

4.2.2. Title: Facilitating Targeted Improvement Efforts

Story: As a first-level supervisor, I want to ensure that EffortLogger V2.0 provides the capability to direct improvement efforts effectively by including identifying information in individual effort and defect reports so that we can enhance predictability and gather best-practice insights from high-performing groups.

Acceptance Criteria:

1. The system should allow the inclusion of identifying information about individuals (e.g., developer 1, engineer 3, supervisor 2) in individual effort and defect reports.
2. Reports should be accessible to authorized users and improvement teams to analyze performance data.
3. Effective improvement efforts should be directed toward groups where predictability improvement is needed based on the analysis of individual-identifying data.
4. High-performing groups should be recognized, and their best practices should be documented for knowledge sharing.
5. The system should support data-driven decision-making by providing tools for analyzing and visualizing performance data.

These user stories address the requirements outlined in the document related to maintaining individual privacy while supporting effective improvement initiatives and data-driven decision-making. They provide clear descriptions of the desired functionality and criteria for success.

4.3 EffortLogger User Input 2023-08-11

4.3.1 Title: Improving Planning Poker Sessions

Story: As a member of our scrum team, I want to ensure that EffortLogger V2.0 improves the poker planning sessions so that our scrum team can effectively and quickly specify the criteria of the project and the weights for each item.

Acceptance Criteria:

1. The system should allow users to input project criteria such as programming language and application domain before a Planning Poker session.
2. The tool should filter historical data based on the specified project criteria, presenting only relevant items for estimation.
3. During the Planning Poker session, the system should enable users to quickly assign weights (e.g., 0 through 4) to each item, indicating its relevance to the current user story.
4. The application should have a stylized look and should share the team's poker card weight once it is time to share.
5. A weighted average calculation should be performed automatically, reflecting the contribution of each item to the story points.
6. Users should be able to adjust weights during the session as discussions progress.
7. The two-step process should significantly reduce the time spent on accessing personal repositories and manual calculations, making Planning Poker more efficient.

These user stories address Kim's suggestions for functional improvements to EffortLogger V2.0, focusing on making Planning Poker sessions more efficient, transparent, and collaborative. They provide clear descriptions of the desired functionality and criteria for success.

5. Derived Requirements

5.1. Ensuring Employee Privacy and Anonymity

A comprehensive approach to safeguarding employees' data. This includes providing user-friendly profile management features, strict access controls, and secure data protocols.

5.1.1. User Profile Management

1. When user data sharing is updated for supervisor view, delete all stored data and history of data the employee no longer wishes to share
2. Options for employees to revoke access to their data at any time and implement a confirmation process to ensure employees understand the implications of revoking access

5.1.2. Secure Data Collection

1. Access controls and authentication methods are in place to restrict authorized access to personal information
2. Use secure protocols for data transfer

5.1.3. Audit Trail

1. Include details such as who accessed the data, when it was accessed, and for what purpose, and provide the authorized admin with the ability to review the audit trail
2. Implement alerts for suspicious or unauthorized access attempts

5.1.4. Compliance with Data Privacy Regulation

1. Conduct a thorough assessment of data privacy regulation (e.g., GDPR, CCPA) to identify specific compliance requirements for company security
2. Integrate compliance checks into the system’s data handling processes and document compliance measures for auditing purposes
3. Regularly update the system to stay in compliance with evolving regulations

5.2. Strengthening Data Security and Protection

A proactive approach describing comprehensive measures to safeguarding data, preventing security breaches, and breach response plan procedures.

5.2.1. Data Encryption and Protection of Products

1. Classify data based on sensitivity (e.g., confidential, sensitive, public)
2. Strong encryption algorithms for different types of data for process optimization
3. Implement encryption for both data at rest (e.g., database encryption), and data in transmission using protocols like HTTPS

4. Robust key management system to securely generate, store, and rotate encryption keys
5. Ensure only authorized personnel have access to decryption keys

5.2.2. Threat Monitoring and Security Assessments

1. Conduct regular security assessments and penetration testing to identify vulnerabilities and address them proactively
2. Configure alerts for suspicious activities and potential security threats
3. Employ intrusion detection systems (IDS) and intrusion prevention systems (IPS) for real-time monitoring of network traffic and system activities

5.2.3. Compliance with Industry Standards

1. Ensure compliance with industry standards for information security, such as ISO/IEC 27001, and legal requirements.
2. Demonstrates the thoroughness and commitment to maintaining a high level of security

5.2.4. Security Breach Response Plan

1. Develop a plan for responding to security breaches both internally through the software and externally within the company practices.
2. Minimize damage and ensure a swift recovery in case of a breach
3. Create an incident response team with clearly defined roles and responsibilities and establish a communication plan for notifying affected parties
4. Define procedures for containing security breaches and rapidly recovering affected systems and conduct post-incident analysis to identify root causes and improve security measures

5.3. Protecting Employee Privacy while Maintaining Transparency

This requirement aims to safeguard employee privacy within the system while ensuring transparency in data handling and access.

5.3.1. Data Anonymization

1. Implement automatic data masking algorithms when users enter their credentials and encrypt logs for secure data storage
2. Require restricted access controls to limit who can access original individual effort and defect reports
3. Ensures transparency and allows users to make informed privacy choices for both present and future best practices.

5.3.2. Transparent Data Flow

1. Provide a user-friendly interface for clear understanding and privacy explanations such as tooltips and notification pop-ups that explain privacy mechanisms

2. Allow users to provide explicit consent during various data-sharing points to track their work throughout the process.
3. Ensures transparency and allows users to make informed privacy choices.

5.3.3. Audit Trail for Access

1. Within restricted pages, provide access controls for viewing the audit trail for the users with timestamped system alerts and notifications every time a new log is published
2. Create automized compliance reports that summarize audit trail data to facilitate compliance audits and accountability reviews
3. Audit trail data is stored securely for an extended period to meet legal and regulatory requirements
4. Enhances accountability, and transparency, and supports compliance audits.

5.4. Facilitating Targeted Improvement Efforts

Identifying individual information involves enabling authorized users and improvement teams to access and analyze performance data through password-protected data pages categorized into their respective teams. This requirement aims to facilitate a data-driven approach to improve initiatives and empower supervisors with the necessary tools for informed decision-making.

5.4.1. Access to Reports

1. Ensure that authorized users and improvement teams can access and analyze performance data given a password-required data page for each individual, categorized into their working teams
2. Supports data-driven improvement efforts and decision-making.

5.4.2. Data-Driven Decision-Making Tools

1. Provide a page that displays user-accessible tools for analyzing and visualizing performance data to support the supervisor's data-driven decision-making.
2. Enables supervisors to make informed decisions based on data.

5.5. Improving Planning Poker Sessions

This updated feature enhances the process by allowing users to input project criteria, assign weights and adjust them if needed mid-process, customize team representation, and generate comprehensive session documentation.

5.5.1. Project Criteria Input

1. Before allowing access to begin the session, users would be required to input project criteria (programming language, application domain, etc.) before a Planning Poker session.

5.5.2. Efficient Weight Assignment

1. Users will be able to quickly assign weights to each item during the Planning Poker session by assigning the weight attribute to the program class through text input with automatic weighted average calculations later on.

5.5.3. Stylized Look and Team Card Sharing

1. Allow teams to choose or create unique tags or avatars to represent a sense of collaboration
2. Prevent two users from selecting the same display card to ensure clear communications

5.5.4. Adjustable Weights

1. Adjusting weights will be through simple mechanisms and changes will be shared in real-time with other users for consistent understanding
2. Adjustments are reflected in the automatic weighted average calculations

5.5.5. Final documentation

1. Generate a summary report that has all items discussed during the session, but also includes any adjustments made during the session and comments or notes provided by participants
2. Allow export for record-keeping and sharing purposes
3. Provide an option to send the summary report to session participants automatically

6. High-Level System Architecture

6.1. Architectural Overview

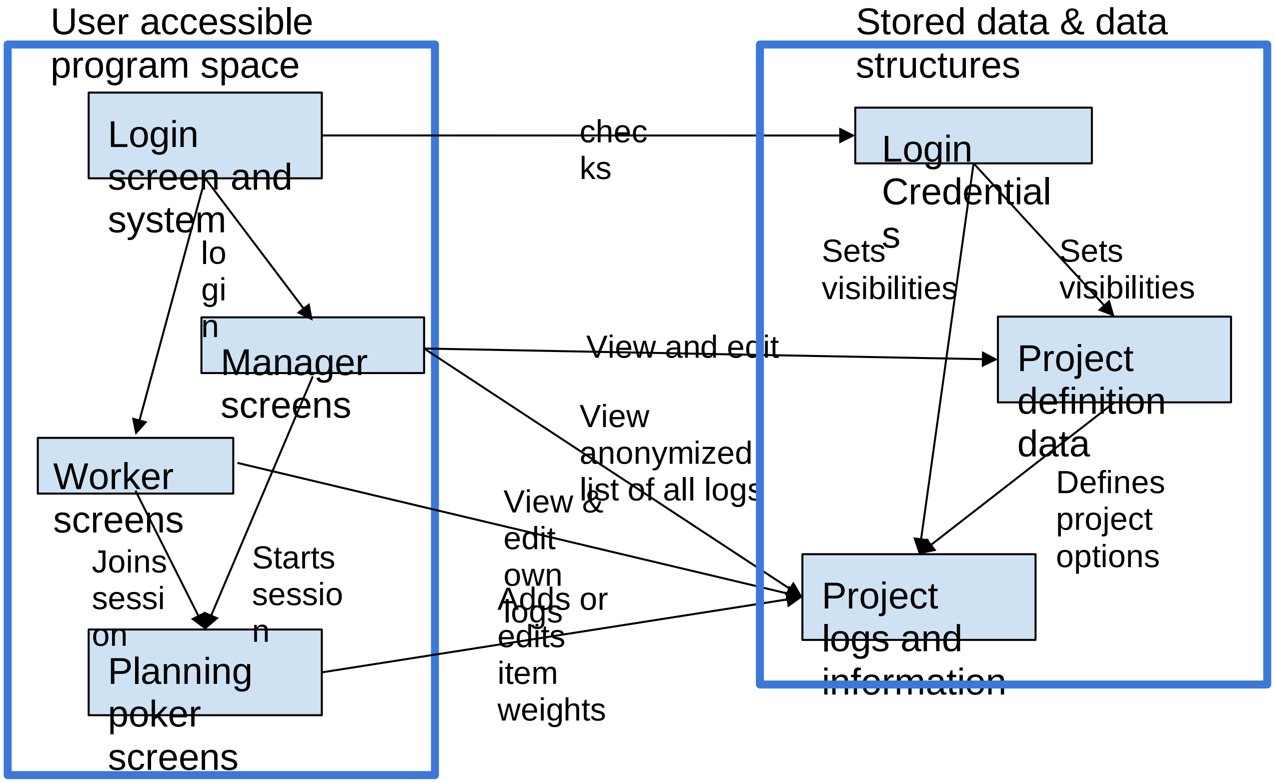
6.1.1. User-accessible program space

- This is the portion of the program that will be interacted with by the end users. Consists of the GUI made in JavaFX and controlled pathways to access stored information.
- The different screens will offer different options and will change based on who is logged in.

6.1.2. Stored data and data structures

- Handles storing login information, project definitions, and project logs. All data will be encrypted, and access permissions will be changed based on the current logged-in user.
- Everything in this section of the program will not be directly accessed by the user and instead will be viewed and edited through the different screens of the user interface.

6.2. Draft Architectural Diagram



6.3. Login Screen and system

6.3.1 Purpose: The major purpose of the login screen is to control who has access to the program as well as to identify the various users. The screen will have two text boxes, the first for the user's name and the second for their password. This will be checked against the stored login credentials to confirm what data the user can access and modify.

6.3.2 Rationale: To allow for data protection there needs to be identity checking which acts as a gateway between the user and the database. This login screen will be that gateway.

6.4. Worker screens

6.4.1 Purpose: The user screens will consist of three screens. They are the effort console, effort log editor, and defect console whose purpose is largely unchanged from EffortLogger v1. The three screens will allow the user to make, check, and edit their logs but will not have any other worker's logs

6.4.2. Rationale: To allow for a smooth transition for workers accustomed to EffortLogger v1. Their main interactions with the system are to be changed only as much as needed. The process of starting a stopped activity as well as creating a defect will be mostly unchanged.

6.5. Manager screens

6.5.1 Purpose: The manager screens will reuse all of the worker screens with two additional ones. The first will be the anonymized list of all logs for a project and the second will be one to access and modify the project definitions.

6.5.2 Rationale: These screens will allow EffortLogger v2 to include a virtual representation of the planning poker sessions.

6.6. Planning poker screens

6.6.1 Purpose: These screens will be how the scrum team runs their planning poker sessions. They will allow a manager to start a session and the workers to join that session in which the weights of the various deliverables defined in the project definition will be given.

6.6.2 Rationale: The manager will be the only one who can start a session and will be in control of it. The workers and manager will be able to vote on the weight of the deliverable. These screens are made to mimic the normal face-to-face planning poker sessions with card weights being the same and a similar system to handle disagreements.

6.7. Login credentials

6.7.1 Purpose: The login credentials will not have a viewable representation and instead are just how the program will store people and their passwords. It will be given a name and a password and will check if that combination of name and password exists in the stored credentials, returning true or false depending.

6.7.2 Rationale: The login credentials will be used to authenticate the user before showing that user, data that should be private.

6.8. Project definition data

6.8.1 Purpose: The project definition data is what was defined in the definitions page of EffortLogger V1. The main difference is now only a project manager will be able to edit that data.

6.8.2 Rationale: A worker working on a project shouldn't need to change the deliverables, life cycle steps, or other options defining the project, so they don't need the ability to change them. If any issue with them did arise they should notify the project manager.

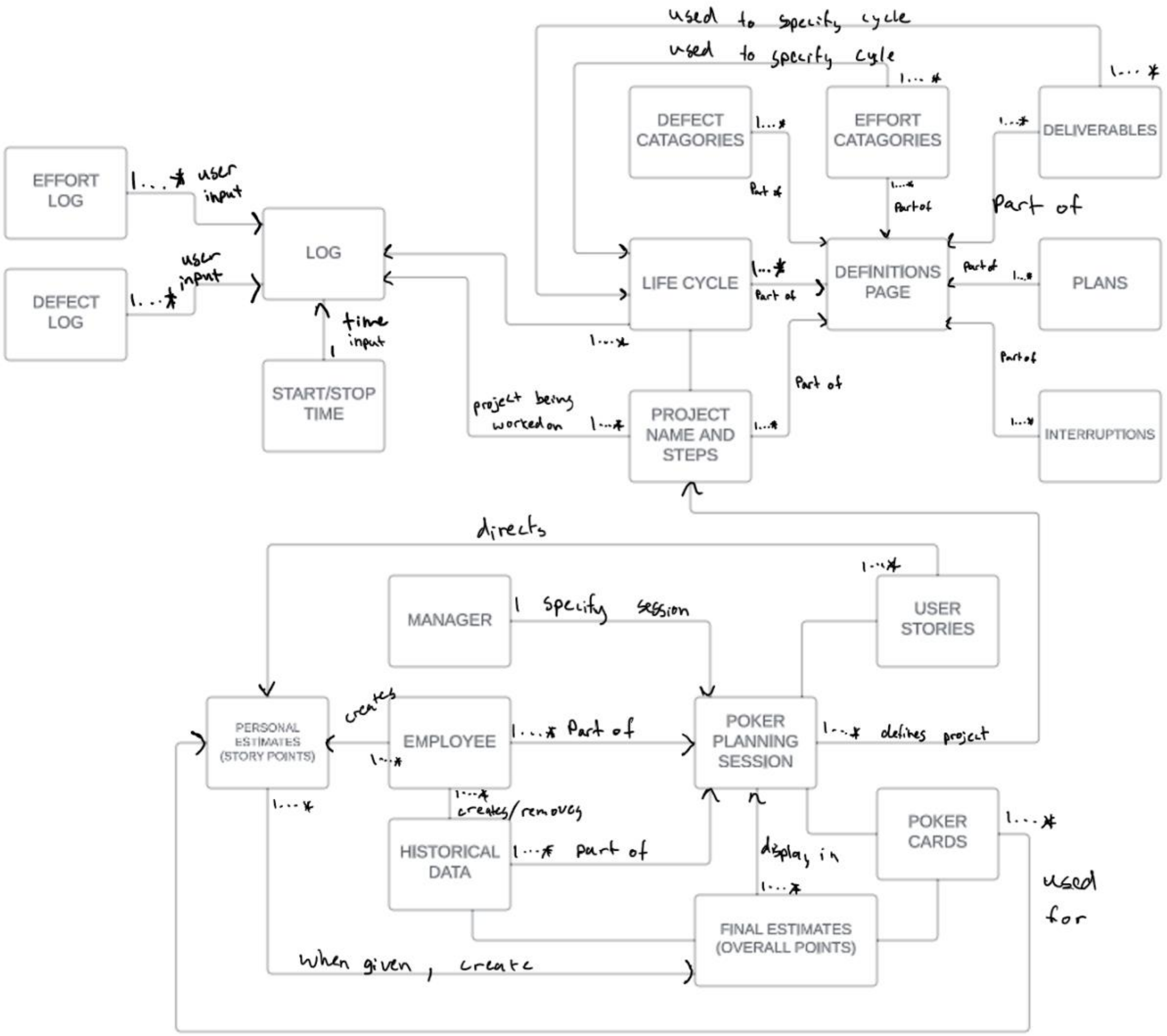
6.9. Project logs and information

6.9.1 Purpose: This is the bulk of the database with each project log having many parameters including one main new one. A log will have all the same info as an EffortLogger v1 log but will also include a viewability ID. This ID will be checked against the current logged-in user and if it matches the log can be viewed and edited. If a manager is then logged in they can view all logs but won't be able to see the ID.

6.9.2 Rationale: The inclusion of the ID will allow the accessibility of the data to be checked before giving the user any data.

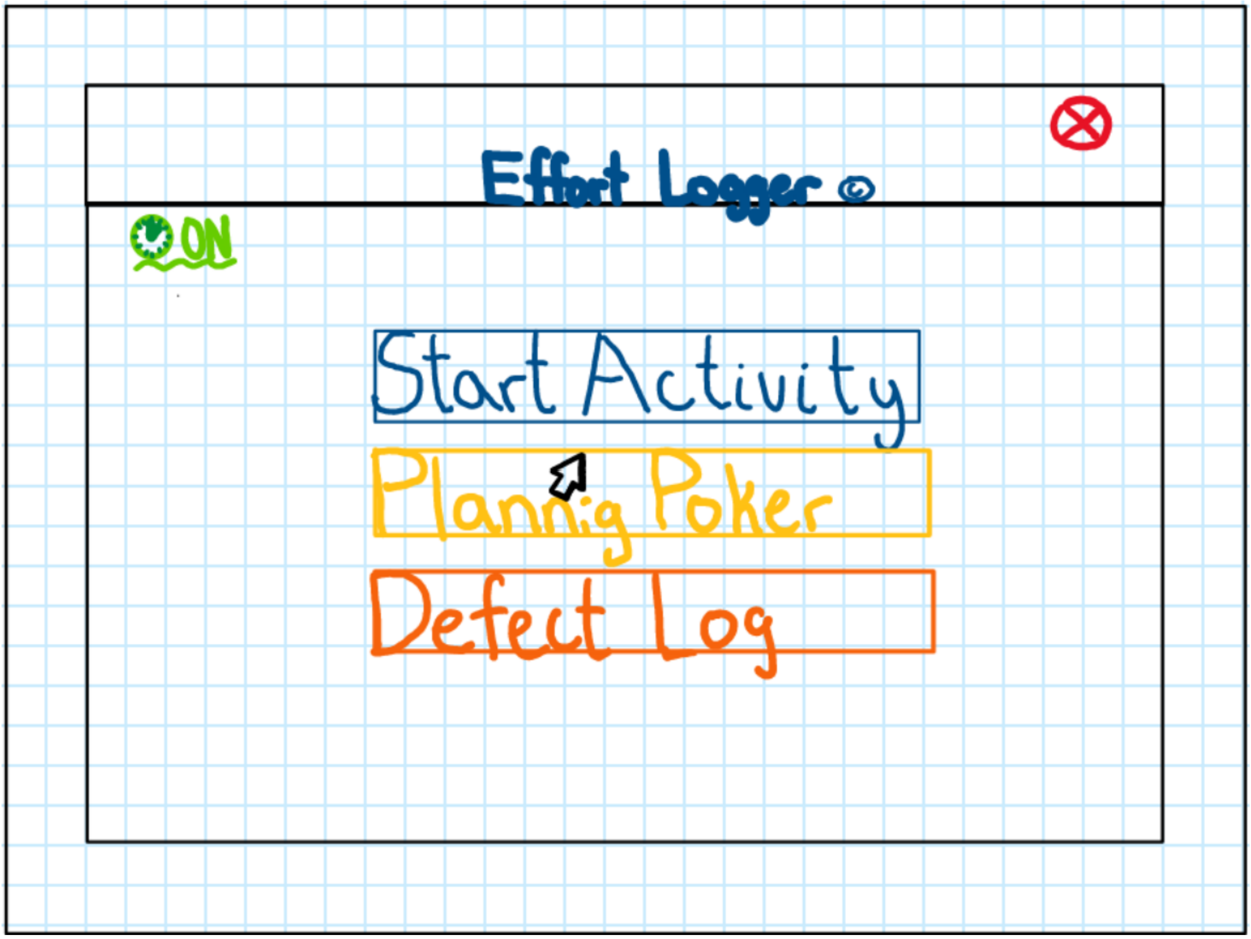
7. Requirements Analysis

7.1 Data Semantic Map

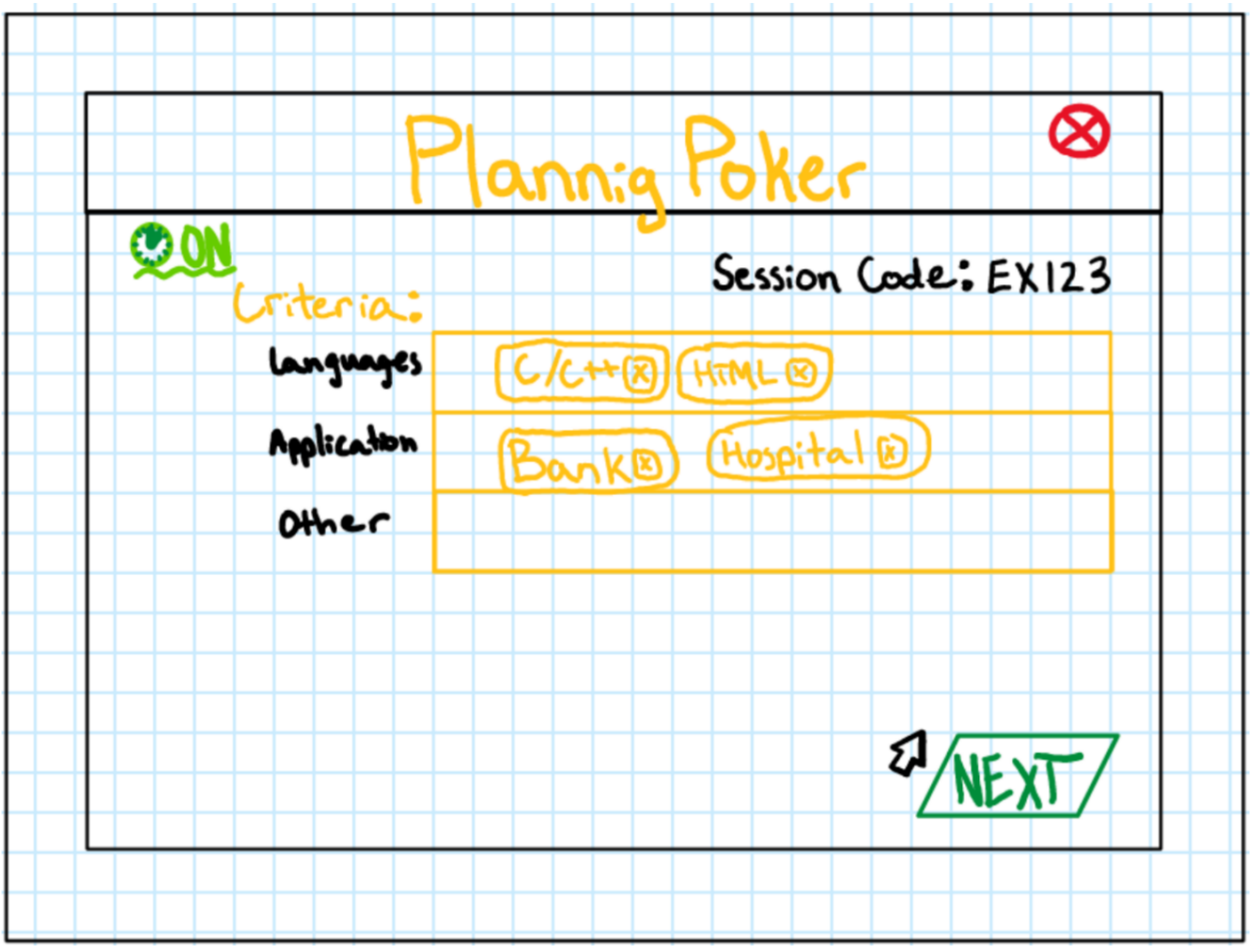


- **Description:** The top section describes the original EffortLogger data flow. The lower sections describe how poker planning session data will flow into the EffortLogger. Once the manager creates a session and gives the session code to employees, they will be able to access the poker planning session. Each employee will provide certain criteria for the project that will be stored in the historical data. The user story data will help the employee determine the story points by selecting a poker card for that user story. The employee estimates will be displayed, and the employees will be able to change their estimates until a consensus is reached. This process iterates in one poker session multiple times to flush out all user stories.

7.2.1 Planning Poker

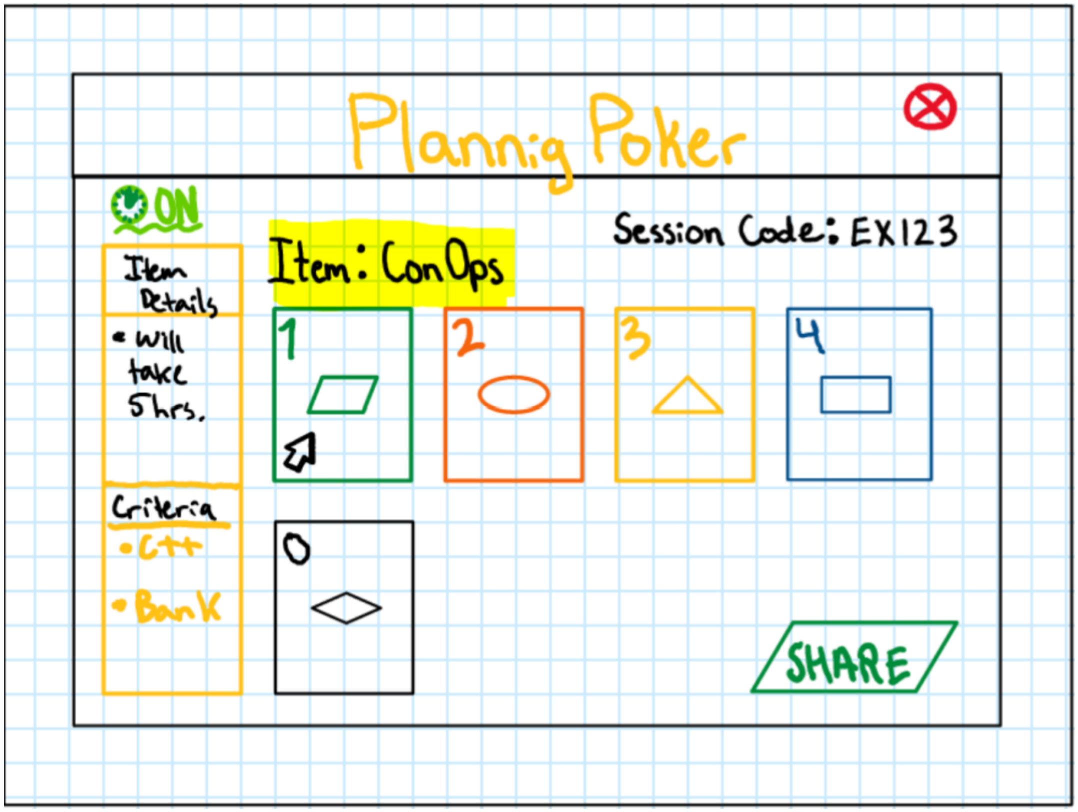


- **Description**: The starting Screen from which each employee lands at the opening of the application
    - This information on the starting screen gives the user multiple options to continue to different functionalities of the application.
        - The clock at the top left can be triggered when any of the buttons, start an activity, planning poker, or defect log
        - Planning Poker Button redirects to planning poker with a team
        - Start Activity redirects to an individual log of activity for the project
        - The defect Log redirects to an individual log of defects for the project

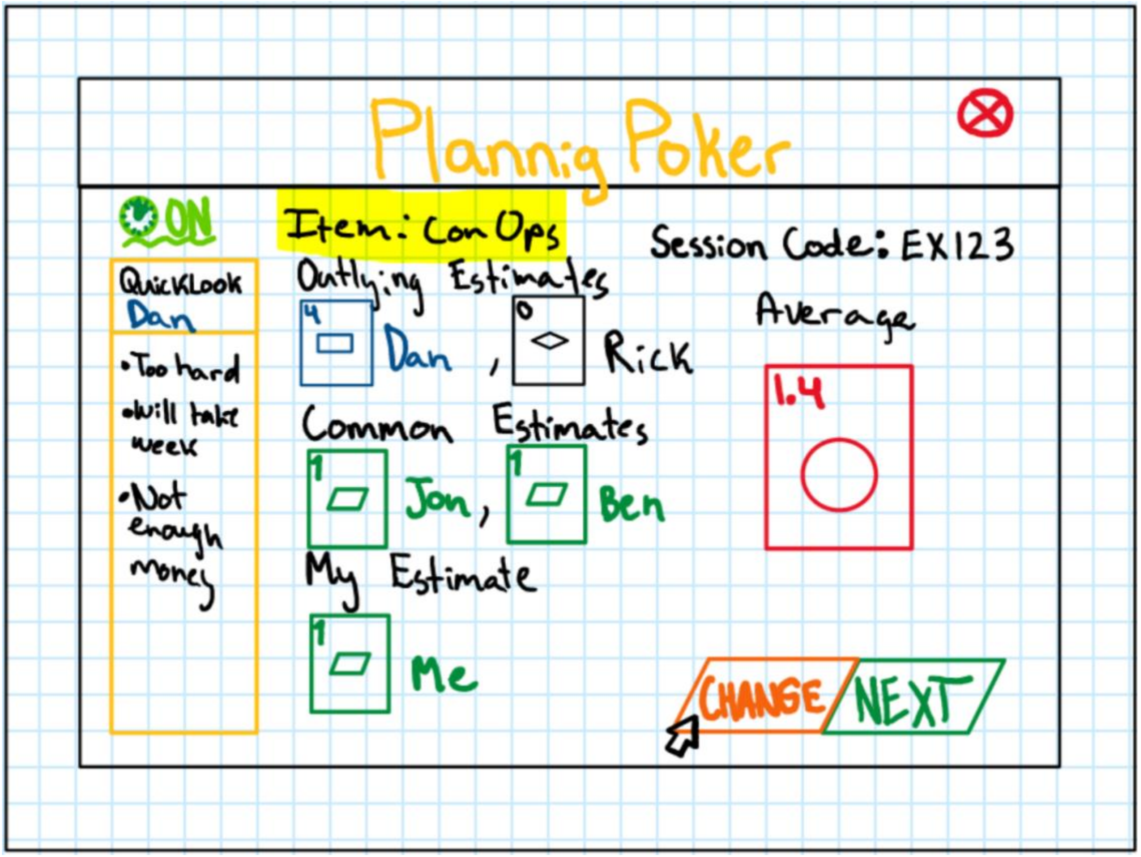


Session Code: EX123

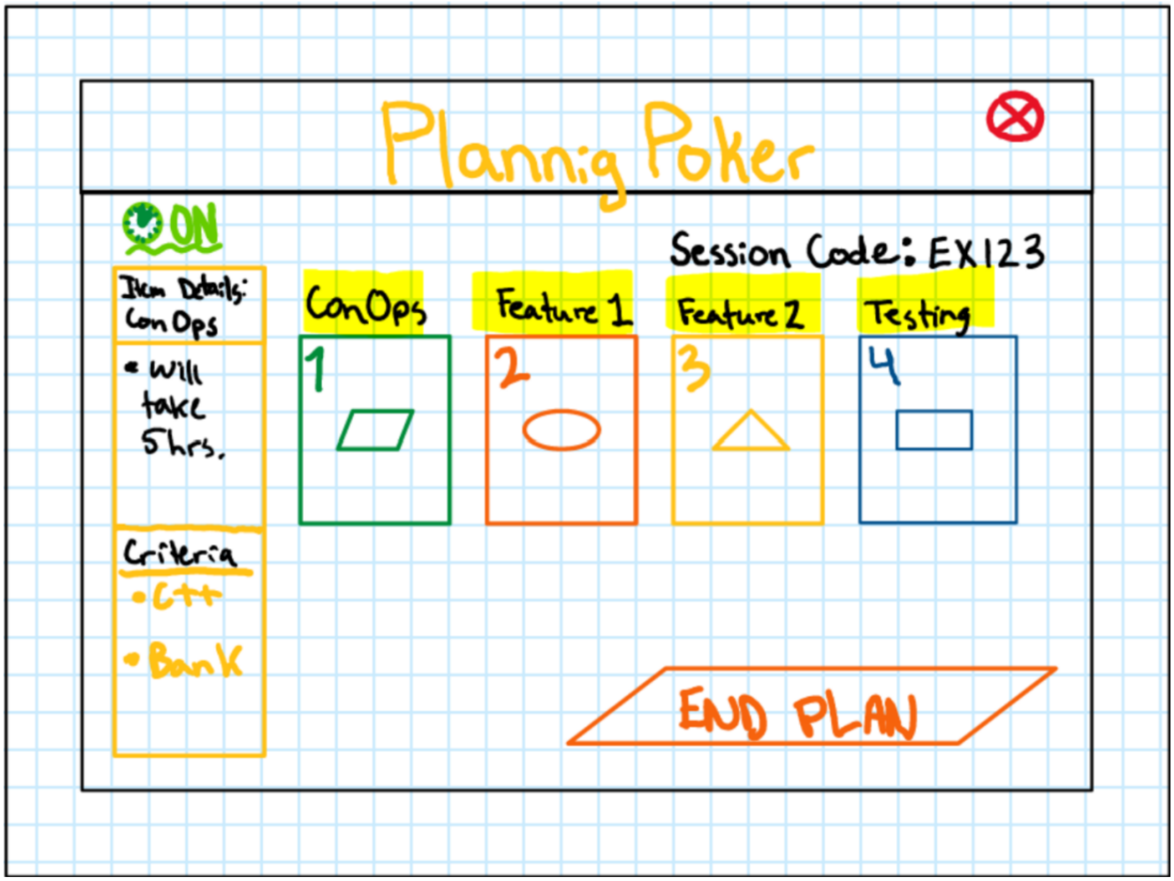
- **Description:** The opening page of Planning Poker first displays multiple options for criteria in a project. A team can in real-time connect with the planning session
  - The session code at the top right
    - This code is how teams connect to discuss items on the same project
  - Criteria section
    - This is a menu of criteria the user can select to determine things already known about the project. This information is winnowed to other sections of the planning poker
  - Next Button
    - Goes onto the next section of the planning



- **Description:** The user plays planning poker by selecting different cards with different weights for an item or feature in a project. Users must write down reasoning and details about items. Know criteria will be displayed for all users in the team group
  - Item Details/Criteria section
    - Item Details is a textbox the user writes in. This information is saved for the user
    - Criterion is a list of known things about the project
  - Poker Cards
    - Clickable cards with weights for the item are displayed
    - The weights of the item are saved for that specific user.
  - Share Button
    - One card is selected and item details filled in, the user can share their estimate

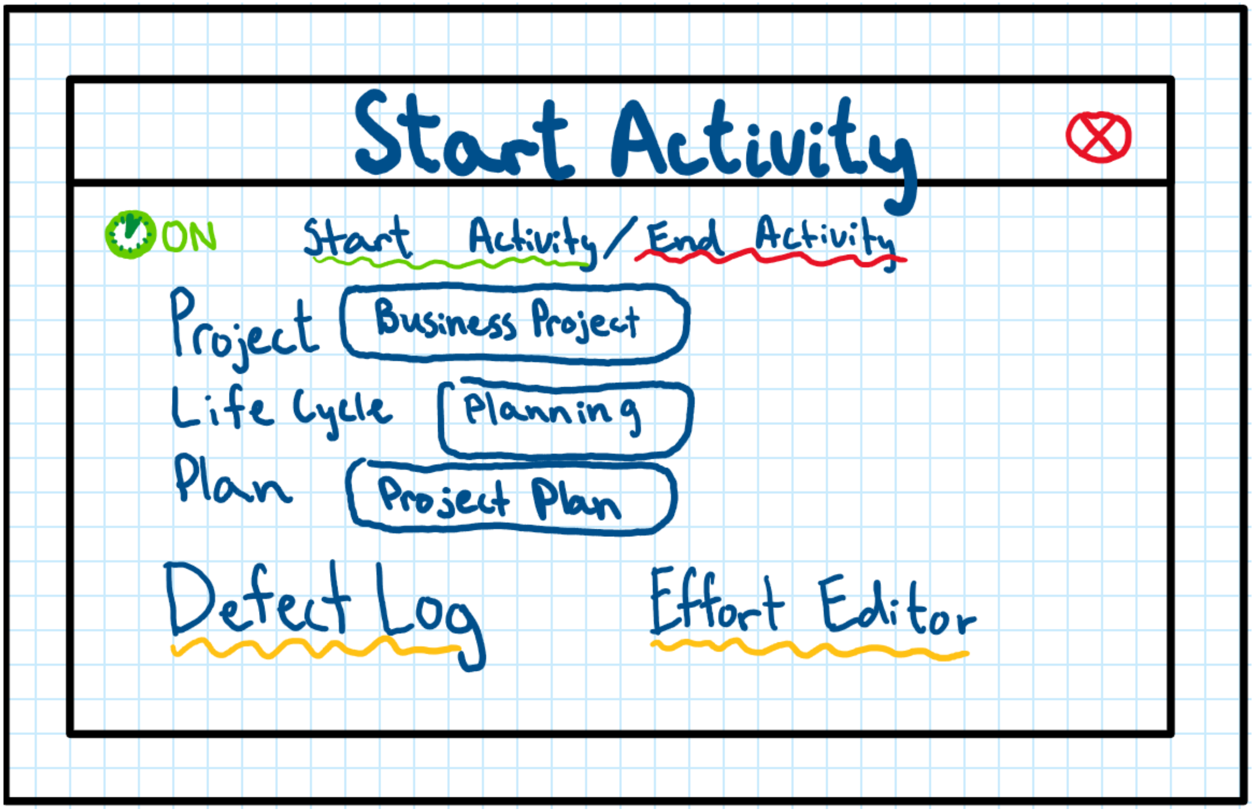


- **Description:** Only when every user in a team group is done with their answers, all team members' cards are displayed under Outlying Estimates, Common Estimates, and My Estimate.
    - QuickLook
        - This text box displays team members’ reasoning for their weighting once a team member card is selected
    - Team Cards
        - Team members’ weighting is displayed under the appropriate section and once selected display information on the QuickLook text box.
    - Average Card
        - Displays unique card of team members’ average weighting
    - Change Button
        - Once members discuss reasoning over a third-party app like Slack or in person, members can change weight to reach unanimity by going back to the previous page
    - Next Button
        - Once unanimity has been reached, all team members can select to go to the next item and iterate the previous two steps for each item in the project.

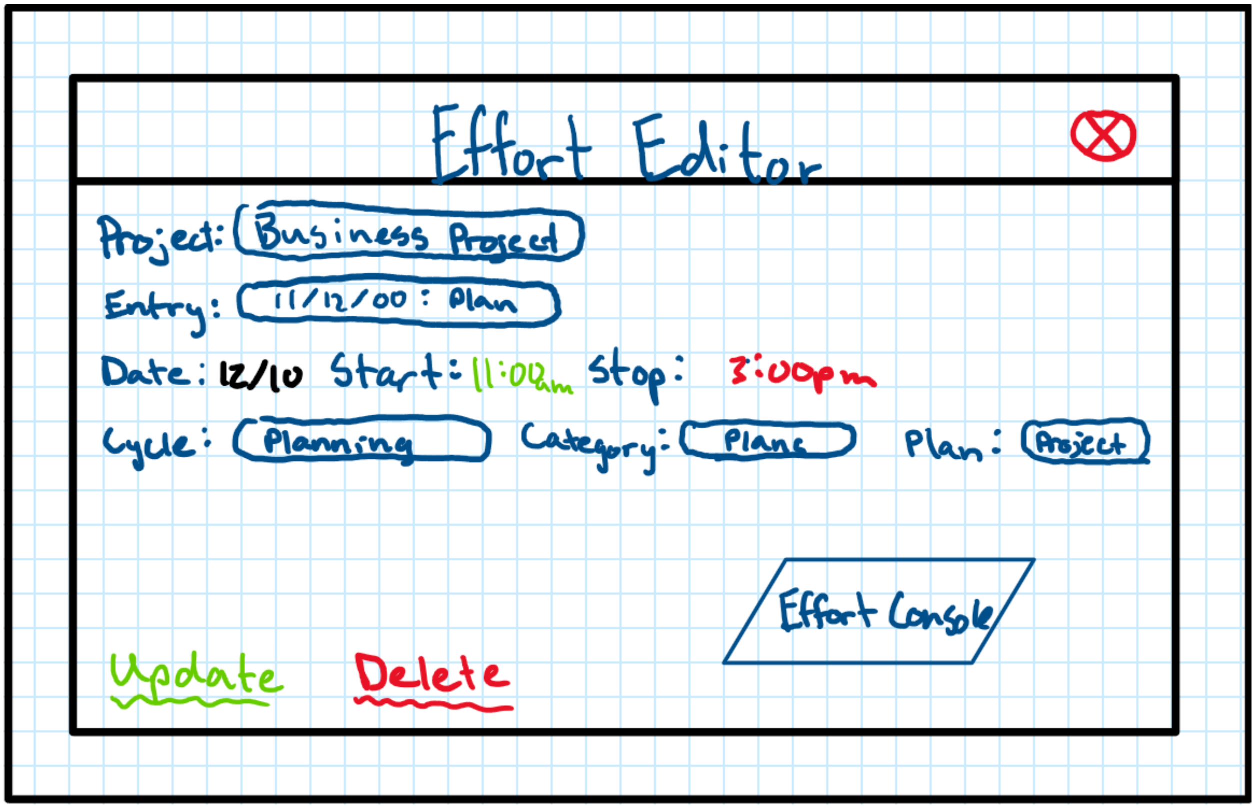


- **Description: The final** page for planning poker displays the final information about all items in the project
    - Item Details
        - This box displays item descriptions specified by all team members for each item once the item card is selected
        - This information is saved for each feature but is not team member-specific
    - Criteria
        - Displays known things about the project
    - End Plan Button
        - Exits the planning session and goes back to the main menu of the effort logger.

7.2.2 Activity Logger



- **Description:** Start Activity is where users can work on specified items of a project.
    - Start/End activity
        - This activates the clock for the specific item
    - Project/Life Cycle/ Plan
        - Specify the project, life cycle of the item, and item that the user will be working on
    - Defect Log/ Effort Editor
        - Redirect user to different parts of the application



Project: Business Project

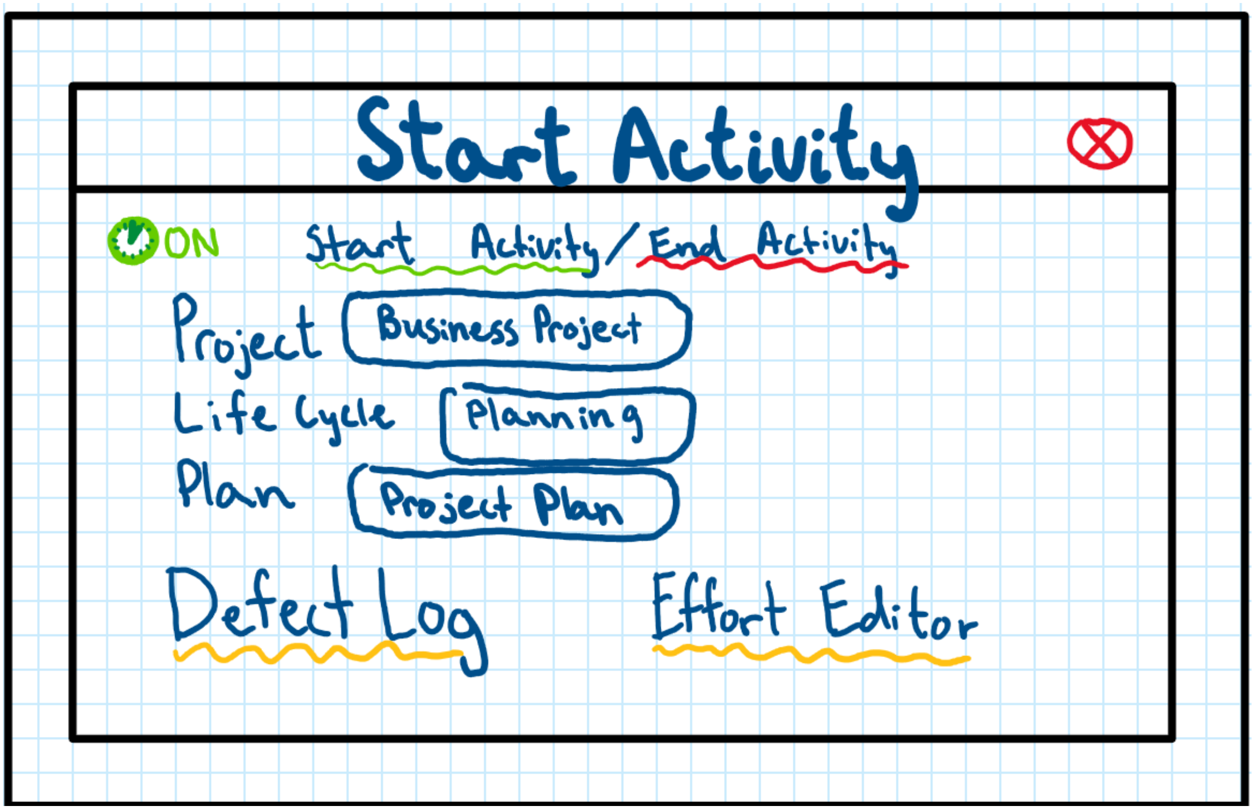
Entry: 11/12/00 : Plan

Date: 12/10 Start: 11:00am Stop: 3:00pm

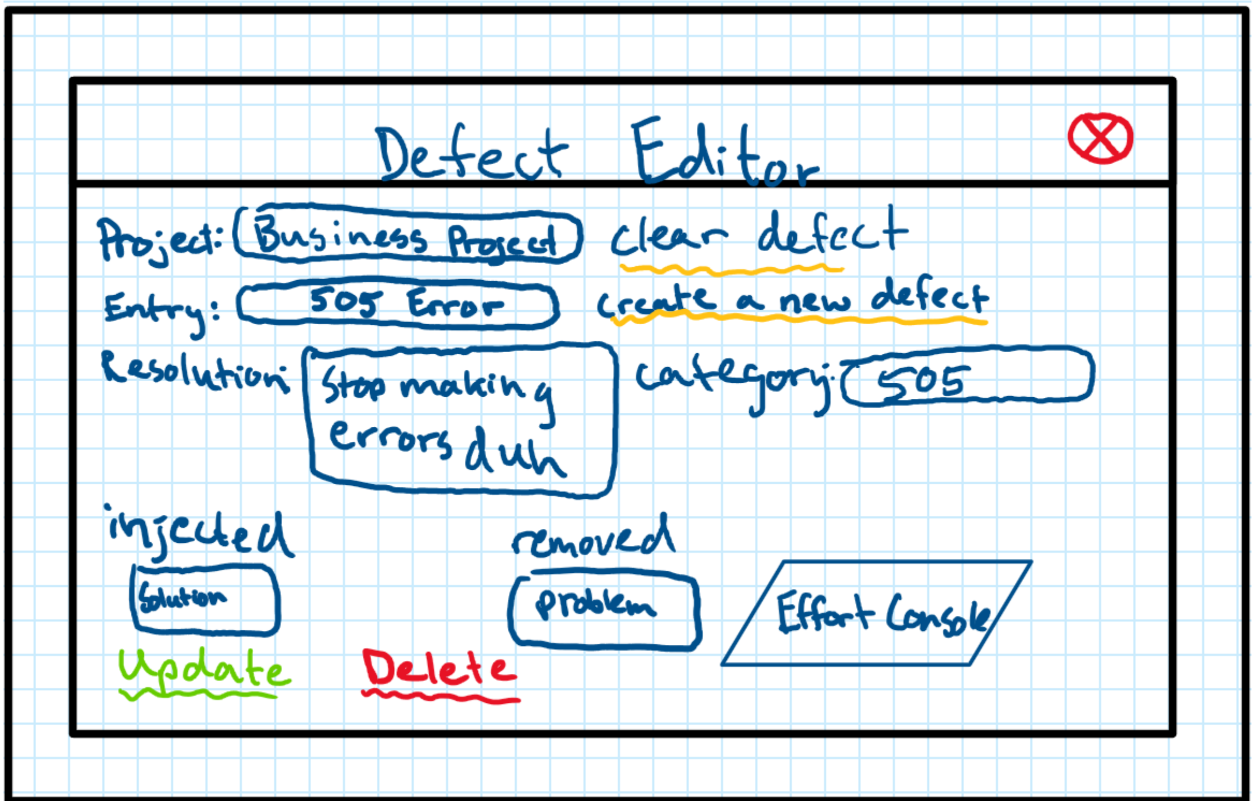
Cycle: Planning Category: Plans Plan: Project

- **Description:** Effort Editor allows users to edit different aspects of the activity they are working on. Different aspects may be added depending on system requirements.
    - Project/Entry/Date/Start/Stop/Cycle/Category/Plan
        - Edit specific attributes of the activity entry the user wants to edit
    - Update/Delete
        - Updates the activity selected or deletes the activity specified
    - Effort Console
        - Go back to the Effort Logger menu

7.3. Defect Editor



- **Description:** Start Activity is where users can work on specified items of a project.
  - Start/End activity
    - This activates the clock for the specific item
  - Project/Life Cycle/ Plan
    - Specify the project, life cycle of the item, and item that the user will be working on
  - Defect Log/ Effort Editor
    - Redirect user to different parts of the application



Project: Business Project

Entry: 505 Error

Resolution: Stop making errors duh

category: 505

injected: Solution

removed: problem

- **Description:** Defect Editor edits and adds defects that come across during the user's activity logs
    - Project/Entry
        - Specify specific entries to be edited or created in a project
    - Resolution Text Box
        - Write possible solutions to defect
    - Category
        - The type of error is selected or edited
    - Inject/Removed
        - Shows where the error was encountered and where the solution will be encountered
    - Update/Delete
        - Update error log
    - Effort Console
        - Redirect to the effort logger menu

7.3 Risk Reduction Throwaway Prototype

Implementing the EffortLogger V2.0 involves several potential risks that may impact the success of the project. We need to identify and prioritize these risks to address them. Here is an outline of the potential risks.

7.3.1 Security Enhancement Prototype

7.3.1.1 Risk: Security Vulnerabilities

Description: Inadequate encryption and access control. Inadequate encryption mechanisms or weak access control may lead to unauthorized access to sensitive project data. Encryption keys need to be managed securely hashing techniques to encode data and give the keys to access the data as private for only the users to use. Any other user trying to access the desired data, even managers, will not be able to access the data without the correct hash key.

7.3.1.2 Risk-Reduction Prototype

- Implement robust encryption mechanisms and secure key management.
- Perform regular security assessments.
- Use intrusion detection mechanisms for prompt security breach notifications.

7.3.2 Privacy Compliance Prototype

7.3.2.1 Risk: Data and User Privacy Compliance

Description: Regular security assessments may point to vulnerabilities to be addressed. The system requires intrusion detection to prompt security breaches. The privacy compliance team needed to emphasize and keep up with data privacy regulations and compliance during and after production. The system needs to integrate compliance checks in the data handling process to prevent legal consequences.

7.3.2.2 Risk-Reduction Prototype

- Emphasize data privacy regulations and compliance.
- Integrate compliance checks into data handling processes.
- Conduct regular audits to ensure ongoing compliance.

7.3.3 Scalability Prototype

7.3.3.1 Risk: Scalability challenges

Description: Poorly optimized database queries may lead to performance bottlenecks as the user base grows. The system needs to be able to handle an increasing user base and resource base. By allocating adequate server space, and having an appropriate plan for scaling, EffortLogger should be able to handle corporate-level use. Lack of redundancy and failover mechanisms may result in system downtime. If a query for data fails, we can have a flag that detects if the data was accessed, if not accessed, we can send another query to the database. This is especially useful if there is high traffic to the web applications

7.3.3.2 Risk Reduction Prototype

- Optimize database queries for improved performance.
- Ensure redundancy and failover mechanisms to prevent system downtime.

- Allocate resources for system scaling.

7.3.4 Planning Poker Prototype

7.3.4.1 Risk: Incomplete Planning Poker Features

Description: If the planning poker is not implemented effectively, it may not streamline the estimation process**.** Inaccurate data or disagreements during sessions may lead to incorrect estimations**.** If the Planning Poker session is not appropriately set up, a back error to reach previous parts of the session should be implemented to allow users to fulfill all parts of the sessions if some parts were skipped or missed.

7.3.4.2 Risk-Reduction Prototype

- Ensure effective implementation of Planning Poker.
- Provide real-time collaboration, discussion tracking, and automated data validation to reduce inaccuracies in estimations
- Implement error handling to address incomplete sessions and disagreements.
- By improving collaboration and data integrity, this prototype will mitigate the risk of Planning Poker issues, such as inaccuracies in estimations due to incomplete collaboration and disagreements. It ensures a smoother and more accurate estimation process.

7.3.5 Project Management Prototype

7.3.5.1 Risk: Poor Project Management

Description: Insufficient resources, which include time, and skills may hinder progress. Resource constraints may result in delays and compromise quality.

7.3.5.2 Risk Reduction Prototype

- This prototype focuses on improving project management within EffortLogger V2.0. It provides features for resource allocation, progress tracking, and risk assessment, enabling effective project management.
- By enhancing project management capabilities, this prototype mitigates the risk of inadequate resource allocation, delays, and compromised quality. It ensures efficient resource utilization and timely project delivery and reduces project management challenges.

8. Conclusion

The new system of systems will implement the enterprise-scale version of agile team development. With added flexibility for expansion and ease of access to critical information will skyrocket team fluidity and morale. The developers will be able to easily maintain a steady workflow and will be able to detect and resolve defects much simpler.

This comprehensive report has outlined the key aspects of the project, including requirements, derived requirements, system architecture, and user stories. In this conclusion, we summarize the critical points to remember moving forward and identify items that require further attention.

9.1. User Privacy and Data Security

9.1.1. User Profile Management

- Users can control what performance data is shared with supervisors.
- Anonymity options are provided, allowing employees to provide performance data without revealing their identity.
- Robust access controls are in place, ensuring data privacy and compliance with data privacy regulations.
- An audit trail tracks data access for accountability and transparency.

9.1.2. Data Handling

- Data is encrypted and protected both at rest and during transmission.
- Regular security assessments and intrusion detection systems are implemented to identify and address vulnerabilities proactively.
- Compliance with industry standards and legal requirements is maintained.
- A security breach response plan is in place to minimize damage and ensure rapid recovery in the event of a breach.

9.2. User-Centric Functionality and Planning Poker Enhancement

9.2.1. User-Friendly Features

- Users can input project criteria before Planning Poker sessions, improving efficiency and relevance.
- Quick assignment of weights during Planning Poker sessions streamlines the estimation process.
- The system provides a stylized look and supports customizable team representation.
- Weights can be adjusted in real time, promoting collaboration and flexibility.
- Detailed session documentation is generated, allowing for record-keeping, and sharing.

9.3. Identifying risks

9.3.1. Implementation and Testing

- Thorough implementation and testing of security measures, ensuring they are effective and robust.
- Continuous monitoring for compliance with evolving data privacy regulations and industry standards.
- Ongoing improvement and refinement of the Planning Poker feature based on user feedback and evolving best practices.

9.3.2. User Training and Adoption

- Comprehensive user training to ensure employees and managers can utilize the system effectively while understanding privacy implications.
- Monitoring user adoption and gathering feedback for iterative improvements.
- Maintaining a proactive stance regarding potential data privacy concerns and emerging security threats.

In conclusion, EffortLogger V2.0 is designed to provide a secure, user-friendly, and feature-rich platform for managing work efforts, enhancing privacy, and supporting data-driven decision-making. By keeping the key points outlined in this report in mind and addressing the items in the "Parking Lot" for further attention, EffortLogger V2.0 can continue to evolve and meet the needs of its stakeholders effectively.

9. Appendix A: Credit Sheet

Team Member Name	Contributions
Team Member 1 Alma Babbitt	Organized group meetups. Parsed each member's work to do. Requirement Analysis: Storyboard, Semantic Map User Requirements Risk Reduction Prototype Revised final document and submitted it
Team Member 2 Trevor Huss	Executive Summary, Conclusion Risk Reduction Prototype
Team Member 3 Karryl Dumalag	Derived Requirements Risk Reduction Prototype
Team Member 4 Zachary Litwin	High Level System Architecture Risk Reduction Prototype
Team Member 5 Ishan Yelnoorkar	Team Norms, Risk Reduction Prototype Revised conclusion Ensure submission standards Update Concept of Operations

10. Appendix B: Current Team Norms

Unsigned Norm Agreement:

https://drive.google.com/file/d/1u-mFyAV-puvFaXHqHHKCrbE5Cu1RqBAJ/view?usp=sharing

Goals

- The team will try to abide by the client's requirements and put forward its best efforts to make that a reality

Meeting and communication norms

- Class time will be utilized to gather notes and information regarding software project management processes.
- Class time will be utilized to brainstorm how these ideas could be used in the project
- The team will meet every Friday to perform a scrum, discuss, manage backlog, and plan future action
  - The team will meet at Noble Library, per convenience and the meeting will last 2 hours
- Apart from weekly meetings, the team will communicate via discord/text chain, giving minor updates every time, a task is complete
- The team will communicate effectively and swiftly to avoid delays in work
- During holidays and long weekends, work will be allocated per the team member's unavailability for the holiday. Work will be redistributed and overworked individuals will be compensated with fewer workloads for the following week after the holiday

Work norms

- The team will work 5 hours every week to ensure timely delivery of deliverables
- The team will split work according to the necessary skills required for the task to be completed
- In case a member of the team is not getting work done, they will be given a warning first, and then a mail to the TAs will be sent for a repeat offense
- The team will set deadlines based on the urgency and time requirements of a task
- Every week a different member of the team will be allocated to proofread work that has been done so far
- Everyone is allowed to work in their manner as long as progress is made, and it does not impede the group's ability to make progress
- All team members are expected to adhere to team norms and meet its expectations

Decision Making

- For a decision to be made, the majority of team members should agree with the agenda put forward, and simultaneously try to understand and help understand why the other's point of view may or may not work with the task at hand

- Team members will listen actively and take into consideration everyone's point of view, as well as try to resolve disagreements

We, group members of Tu37 have agreed to follow the terms listed above and plan to adhere to them until the culmination of this project. Our listed names below indicate our acceptance of the norms and are used as our digital signature.

Member 1: Alma Babbitt

Member 2: Zachary Litwin

Member 3: Trevor Huss

Member 4: Karryl Dumalag

Member 5: Ishan Yelnoorkar