CSE 360 Project Report Number 4
Team Tu37

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

Table of Contents:

1. EffortLogger Risks

1.1 Risk 1: Log Information for Project Management

1.1.1 Description: Insufficient resources, which include time, and skills may hinder progress. Resource constraints may result in delays and compromise quality.
1.1.2 Importance: Improves project management within EffortLogger V2.0. It provides features for resource allocation, progress tracking, and risk assessment.
1.1.3 Mitigation: Provide a page that displays user-accessible tools for analyzing and visualizing performance data to support the supervisor's data-driven decision-making.
1.1.4 Assigned to: Alma Babbitt

1.2 Risk 2: Risk Assessment and Security Measures for Login Screen

1.2.1 Description: The major risk related to the login screen is that, if not implemented securely, it could expose the program to unauthorized access or data breaches. On the upside, a well-implemented login screen enhances data protection and ensures that only authorized users can access and modify data.
1.2.2 Importance: The successful implementation of a secure login screen is paramount to the program's functionality and reputation, making it one of the program's highest priorities. The importance of a secure login screen serves as the primary line of defense against unauthorized access, data breaches, and potential misuse of the program. It is a critical component for ensuring data protection, user privacy, and the overall integrity of the system.
1.2.3 Mitigation: Appropriate error handling and logging to identify and respond to potential threats, regular audit checks of both successful and unsuccessful login attempts, enforcing password complexity rules, and account lockout policies.
1.2.4 Assigned to: Karryl Dumalag

1.3 Risk 3: Project and Login Information Storage

1.3.1 Description: File Storage will allow the number and size of projects to increase. However, insecure file storage is a vulnerability that puts employee information at risk. Encrypting the files mitigates that risk by preventing the data from being read by unwanted programs.
1.3.2 Importance: Gives Effortlogger the ability to store and load projects from encrypted files. Improves the security of employee information and provides a mechanism for stopping unwanted access to stored information.
1.3.3 Mitigation: Convert the project and login information to an encrypted byte stream then store that byte stream to a file. Then when the file is requested by the program it is able to decrypt the byte stream and convert it back to usable information.
1.3.4 Assigned to: Zachary Litwin

1.4 Risk 4: Inadequate roles-based access control

1.4.1 Description: Unauthorized access to sensitive project data due to weak role-based access control increases the risks of uninterested users gaining access to confidential project information, potentially leading to data breaches.
1.4.2 Importance: Improves security by maintaining strict access control over project data, effort entries, and defect entries. Prevents unauthorized users from making changes to project data.
1.4.3 Mitigation: Clearly define specific user roles and permissions that align with the user's responsibility. The permissions include rights to view, edit, create, or delete data.
1.4.4 Assigned to: Ishan Yelnoorkar

1.5 Risk 5: Excel Sheet Use Reduction and UI Improvements

1.5.1 Description: The usage of Excel in EffortLogger V1.0 is not regulated enough and leads to less productivity and security. The improvements on the user interface will give EffortLoggerV2.0 a new look and will make using it much more seamless.

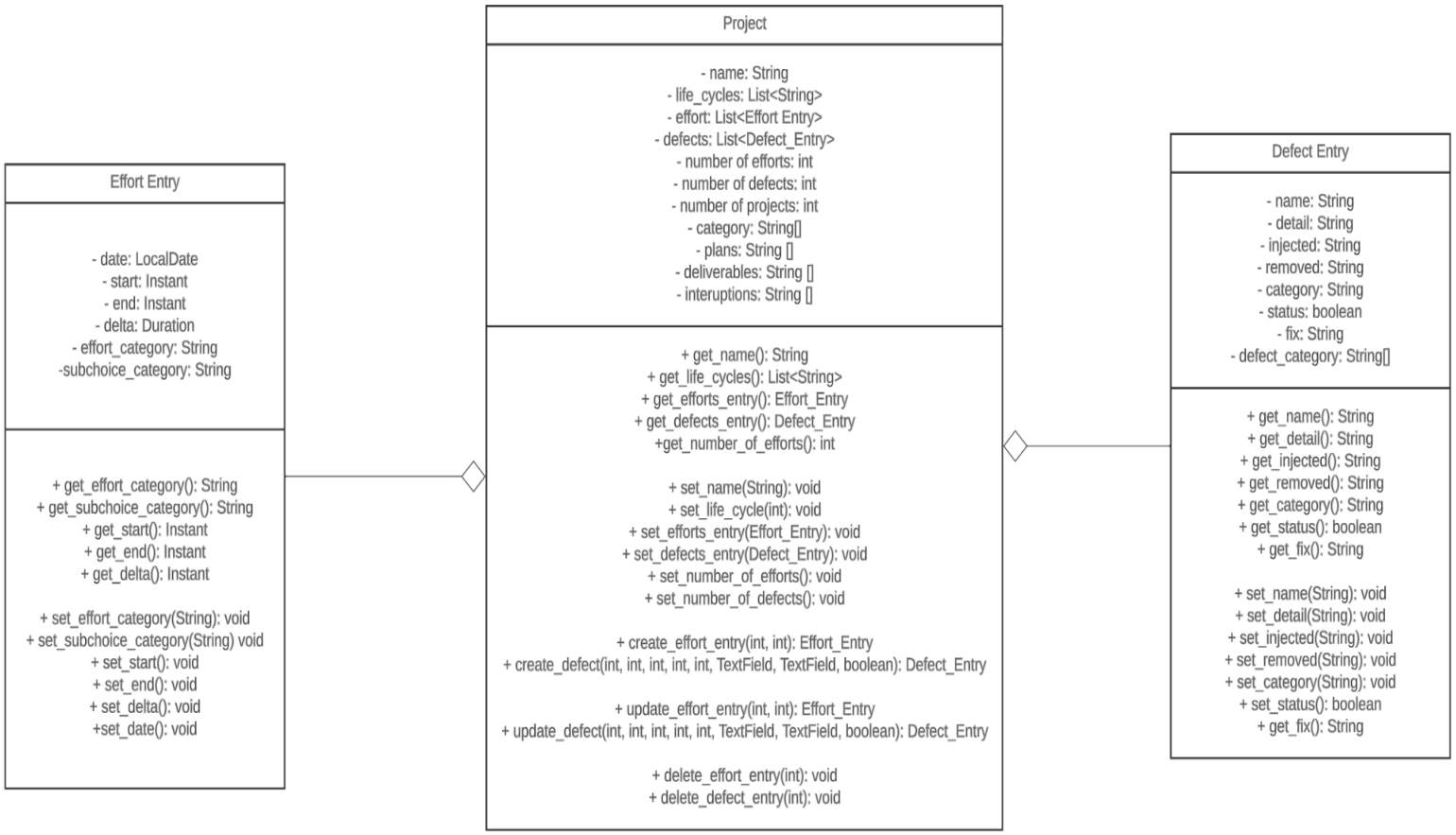
1.5.2 Importance: This will give EffortLoggerV2.0's users a much smoother experience as well as mitigate the risk of data leakage.
1.5.3 Mitigation: Create a user interface that will make using the program easy for the employees and will not grant too much access to important information.
1.5.4 Assigned to: Trevor Huss

2. EffortLogger Risk Reduction Prototypes

2.1 Prototype: Log Information and Sensitivity for Project Management

2.1.1 Risk Reduction Prototype Description: In my prototype, I will clearly define project relationships with effort entries and defect entries to create a log that managers and employees can see. Their information will be stored on a text file that will be created in a separate prototype. Each class will have methods that will produce the functionality of buttons and textfields.

2.1.2 Risk Reduction Prototype UML Design Diagrams



2.1.3 Risk Reduction Prototype Testing: My test will be to see if objects are made, methods work properly, and the data shared among the classes works. I will use a main program to test this functionality.

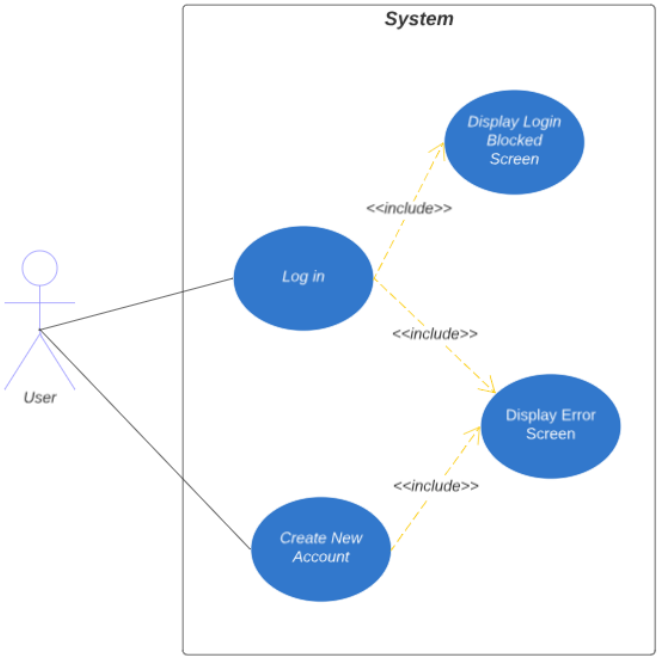
2.1.4 URL to a Screencast for Prototype 1

https://drive.google.com/file/d/1apGbg6KXUl2FY-R_VkCo0mjxFN4PQYRx/view?usp=drive_link

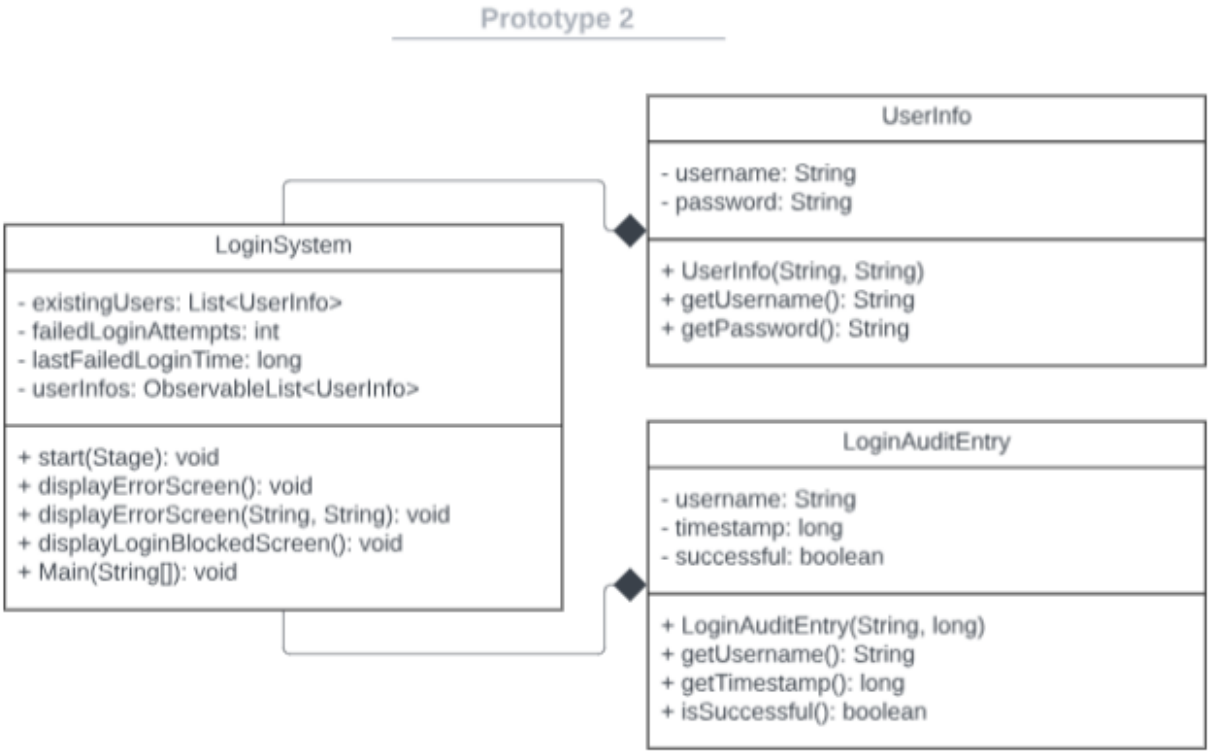
2.2 EffortLogger Risk Reduction Prototype 2

2.2.1 Risk Reduction Prototype Description: This prototype simulates a basic login page, featuring a GUI for users to log in and create new accounts, and also maintains a list of existing users for authentication, with failed login attempts tracked and blocked when necessary. The login screen features text fields for entering a username and password, which then stores that input into a list 'userInfo.' The login process is initiated by clicking the 'Login' button, which is then followed by either an alert screen if the login fails (or a trigger for a locking mechanism if too many unsuccessful login attempts are observed), or an open path if credentials entered match the 'existingUsers' stored data. A 'New Account' page opens after the associated button is triggered, which then provides fields for entering a new username and password, with validation to ensure both fields are filled correctly.

2.2.2 Risk Reduction Prototype Use Case(s)



2.2.3 Risk Reduction Prototype UML Design Diagrams



2.3.4 Risk Reduction Prototype Testing: This prototype has undergone testing of usability, user authentication, error handling, and new account creation process in regard to acceptance criteria. The following scenarios were performed and passed:

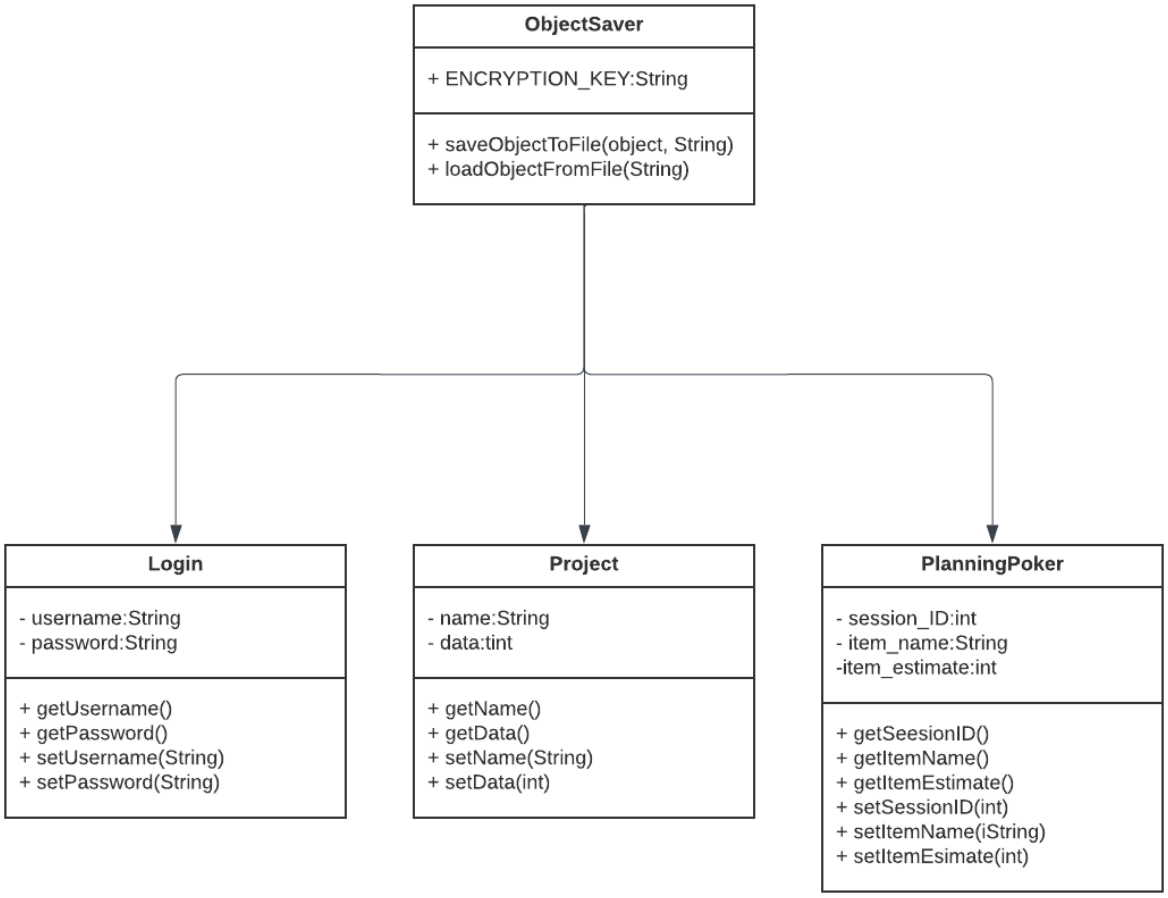
1. Logging in with a valid username and password
2. Logging in with an invalid username and password
3. Testing the "Login Blocked feature" that prevents any new login attempts for 2 minutes after more than 5 unsuccessful tries
4. Creating a new account with a valid password and logging in with the account
5. Testing account creation with the creation of an invalid password and making sure that the 'newUser' was not entered into the 'existingUsers' list

2.3.5 URL to a Screencast:
https://drive.google.com/file/d/1fj5WMYy8q3_bzX_3VSBFDs6K5-x9jPu3/view?usp=sharing

2.3 Prototype: Project and Login Information Storage

2.3.1 Risk Reduction Prototype Description: In this prototype, I will make a class called ObjectSaver and its job will be taking in an object, of any kind, and converting it into a byte stream. Then that byte stream will be encrypted using a cipher and stored in a file of a specified name. The class' other job is to read a file reverse the previous operation and return the object.

2.3.2 Risk Reduction Prototype UML Design Diagrams



2.3.3 Risk Reduction Prototype Testing: I will test that objects of the various template classes will properly write to and read from a file. I will also test for when the program tries to read from a file that doesn't exist. This prototype also uses three "dummy" classes, Login, Project, and PlanningPoker, these classes were just for testing ObjectSaver and will not be used in EffortLogger

2.3.4 URL to a Screencast for Prototype 3:

https://drive.google.com/file/d/14v0I8_yc8uEXSD_1OiRqpODgk94HQMiL/view?usp=sharing

2.4 Risk Reduction Prototype 4: Role-based access control

2.4.1. Description: My prototype aims to introduce role-based access control in EffortLogger V2 to mitigate the risk of unauthorized access to sensitive project data. It ensures that users are assigned permissions that align with their roles and job functions. This helps improve data security, and unnecessary interference from third parties.

2.4.2 Use cases

Use case: Employee Accesses Effort Logger

Actors: Employee

Data: User credentials

Stimulus: Employees log into the EffortLogger tools. The system authenticates employees with login credentials

Response: The system gives access to the employee for the EffortLogger features

Use case: The manager views all effort entries and defect entries

Actors: Manager

Data: Effort and Defect Entries

Stimulus: The manager accesses the effort and defect entries

Response: The manager is permitted to view all effort and defect entries for a project, enabling them to oversee and manage progress

Use case: Access control

Actors: Employee

Data: User roles and associated permissions and the application features

Stimulus: The employee accesses the creation of a project, and the system checks the user's associated permissions

Response: Employee is denied permission to create a project

Use case: Create new users

Actors: Manager:

Data: Login credentials

Stimulus: The manager puts in login credentials, and the system authenticates the credentials

Response: The manager is allowed to create a new user

Use case: Access effort entries

Actors: Employee

Data: Effort Log entries

Stimulus: The employees stop an activity to view effort entries, and the system checks if an employee has permissions

Response: Employee is allowed to access and edit their effort entry

Use case: Oversee project progress

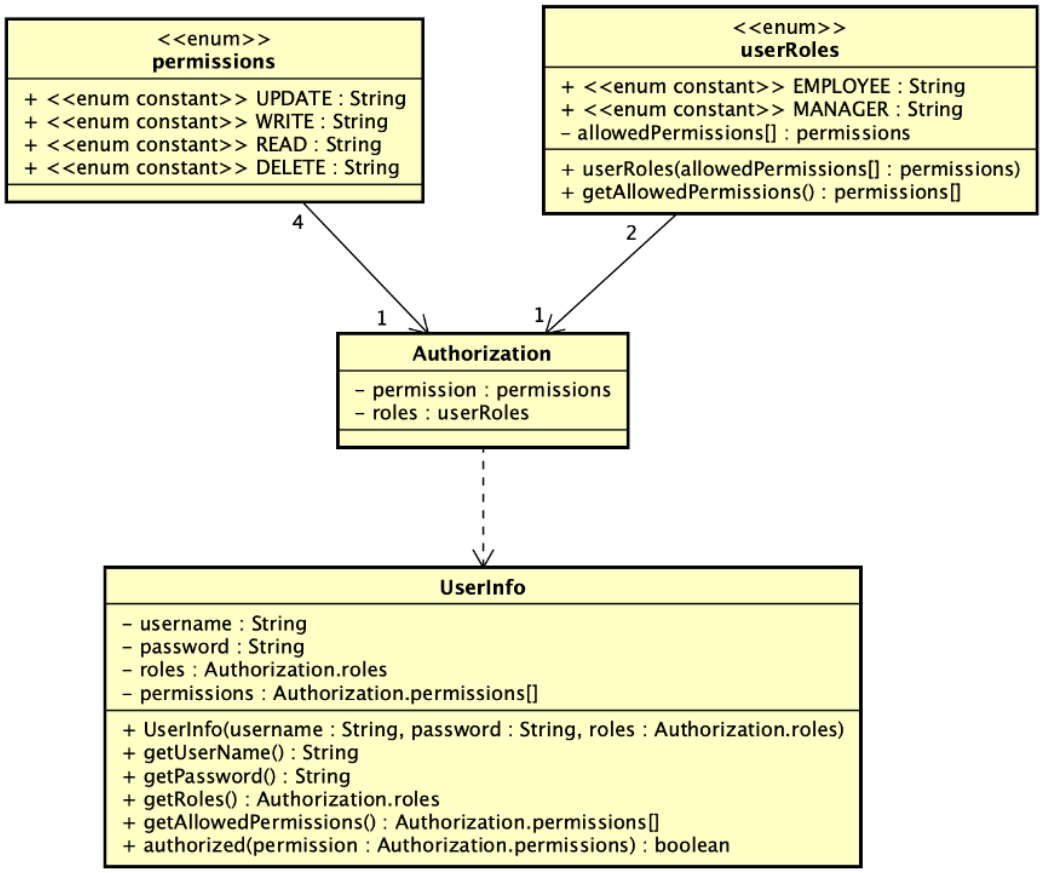
Actors: Third party

Data: Effort Logs

Stimulus: A third-party user tries to update effort logs, and the system checks user permissions

Response: The user is denied permission due to inadequate authorization

2.4.3. UML Design



2.4.4 Risk Reduction Prototype Testing: To validate that the role-based authorization prototype is working correctly, here is how the prototype will be tested:

Role validation: I will ensure that users are assigned the correct roles and that roles determine their permissions accurately

Permission testing: I will test whether each permission works correctly, that is, ensure a user can read, edit, update, or delete

Security assessment: I will check if the system blocks authorized access attempts.

Use-case testing: Test key use cases, including the creation of a project, accessing effort logger, etc.

2.4.5 Screencast Link:
https://drive.google.com/file/d/1PoB7BgxhjVdPD_9cpuiuM1qPKZEVr27M/view?usp=sharing

2.5 Prototype: Excel Sheet Use Reduction and UI Improvements

2.5.1 Risk Reduction Prototype Description: In my prototype, I will create a user interface that will reduce the amount of interaction with Excel. This will increase productivity and security so that users cannot gain access to large amounts of data in one instance. I will create well-placed buttons, choice boxes, and text fields to seamlessly guide users to what they need to do without giving access to too much information.

2.5.2 Risk Reduction Prototype Use Case(s) and Diagram:

Use Case 1 Access the Effort Log console and perform respective actions

Actors: Employees

Data: Start and Stop times and effort categories

Use Case 2 Access the Effort Log Editor and perform respective actions

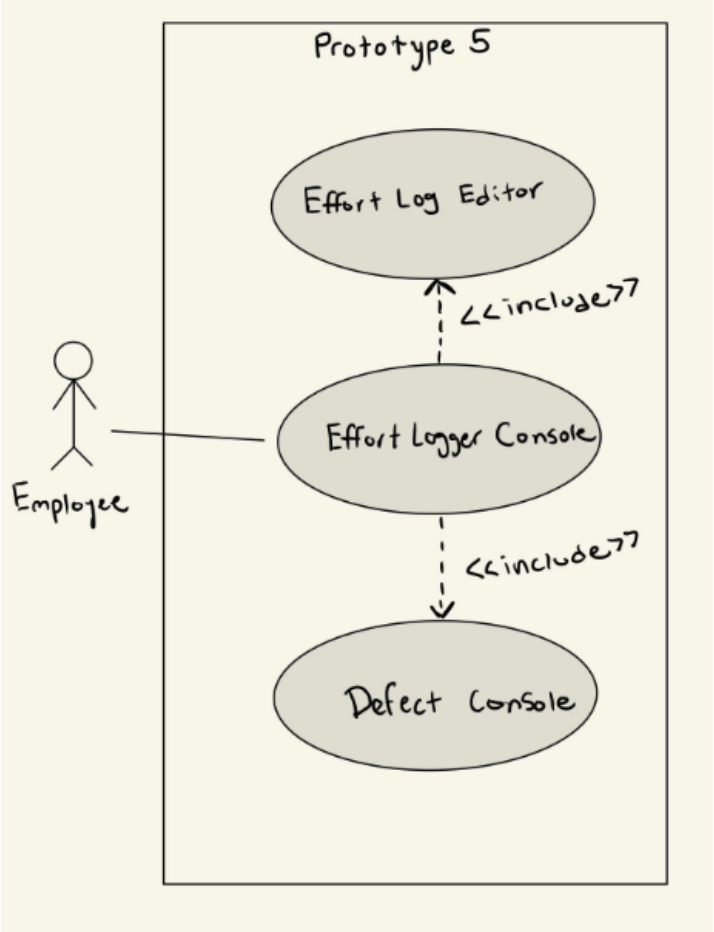
Actors: Employees

Data: Effort Data

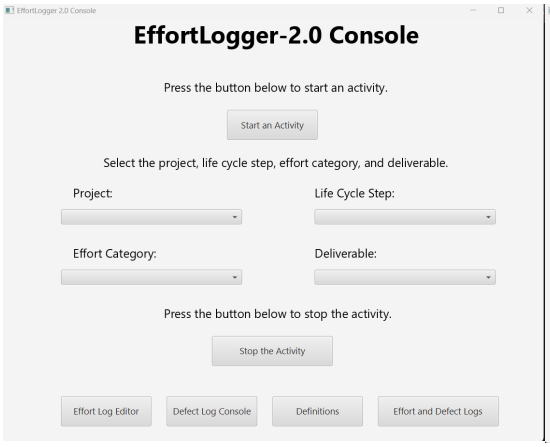
Use Case 3 Access the Defect Log and perform respective actions

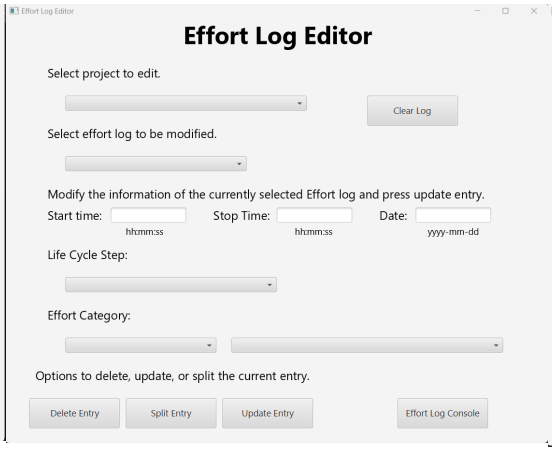
Actors: Employees

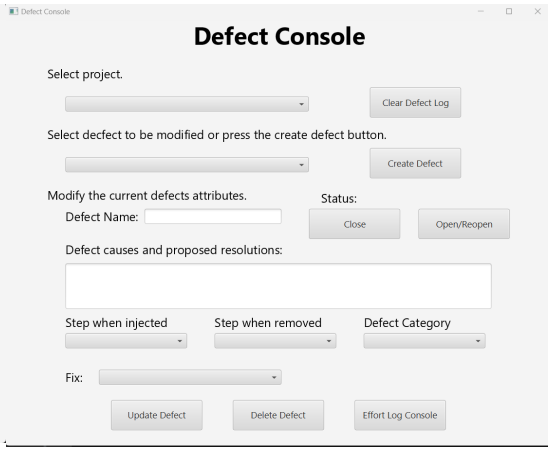
Data: Defect data



2.5.3. Risk Reduction Prototype Testing: Each Section of the UI compiles and works perfectly. The buttons that transfer between the different sections work as intended. The screens look like the pictures below.





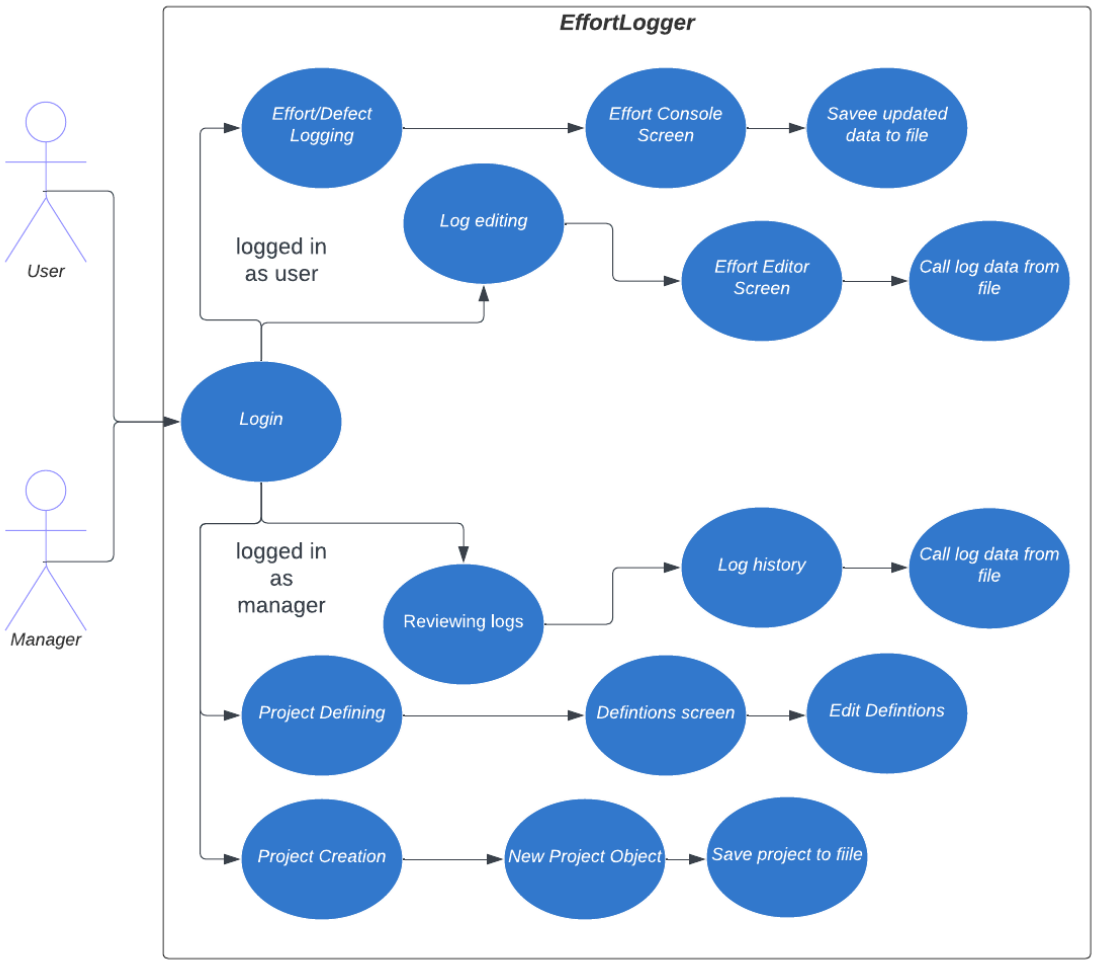


2.5.4. URL to a Screencast

https://drive.google.com/file/d/1AKSMau5IIXd-6uQe3IsFIJFUPnkdIgKL/view?usp=sharing

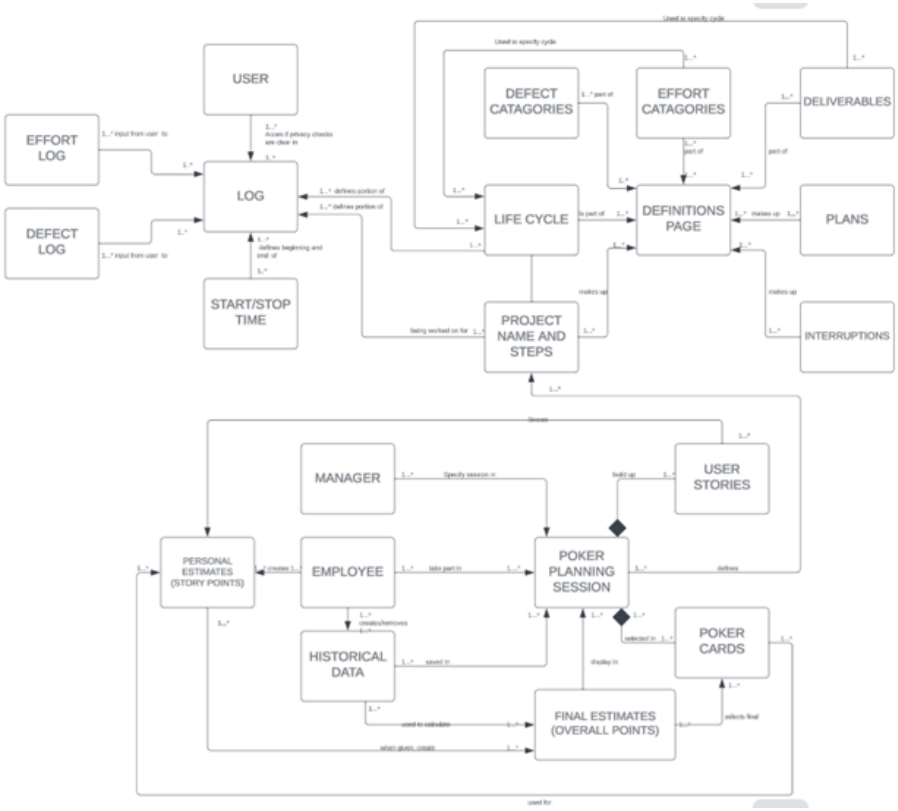
3. EffortLogger V2 Integrated Team Prototype

- **Description:** EffortLogger V2 is the enhanced project management and performance tracking tool that prioritizes employee privacy, data security, and agile development. It offers user-controlled data sharing, employee anonymization, robust data protection along tools for data-driven decision making.
- **EffortLogger V2 Prototype Use Case(s):**



- Depending on if a user or a manager is logged in potential actions in the program change. Managers will review the past project logs, create and define projects, and users will create and edit effort and defect logs.
- Project creation will initialize an instance of the Project class and then save that to a file.
- Reviewing past logs will load the project from a file and then display it on the logs screen.
- A project's details can be defined in the definitions screen and that will both load and then save the project to and from a file.
- Creating effort and defect logs will use their respective screens and they will be saved to their respective project.
- Editing effort and defect logs involve loading the log information into the editor screens and then saving the updated logs back to the project after changes have been made

- **EffortLogger V2 Prototype UML Design Diagrams**



- **URL to a Screencast**

https://drive.google.com/file/d/1MWpBUIYrLEVnh6N7uWYqI29j5E0IADqS/view?usp=sharing

4. Conclusion

The initial development of the EffortLogger V2.0 has provided valuable insights into potential risks and their mitigation strategies. Below is the summary of how risks were addressed and what future actions look like.

4.1. The most important conclusions

4.1.1. Efficient Project Management

- There are risks of insufficient resources being allocated to participating, potentially causing delays, and compromising the quality of the project.
- To mitigate the risk, we have developed a prototype that defines project relationships with effort and defect entries, creating a log for managers and employees.

4.1.2. Secure Login Screen

- Implementing a secure login screen is crucial for the program's functionality and reputation. It acts as the primary defense against unauthorized access and data breaches.
- The risk of insecure login screens has been mitigated through the development of a prototype with features such as error handling, audit checks, password complexity rules, and account lockout policies.

4.1.3. Data Storage and Security

- The EffortLogger handles vital project information and sensitive user details, therefore it is essential to ensure the security of this data.
- This prototype focuses on converting project and login information into encrypted byte streams, thus safeguarding employee information from unwanted access.

4.1.4. Role-Based Access Control

- Unauthorized access to sensitive project data due to weak role-based access control poses a significant risk.
- To address this, we have designed a prototype that enforces strict access control by defining specific user roles and permissions. This ensures that users can only view, edit, create, or delete data according to their responsibilities.

4.1.5. Excel Sheet Use Reduction and UI Improvements

- The excessive use of Excel in EffortLogger V1.0 can lead to productivity and security issues.
- To address this, our prototype focuses on improving the user interface by reducing reliance on Excel and providing users a more seamless and secure experience.

4.2. Upcoming Important Activities

4.2.1. Prototype Testing and Integration

- The next crucial step is to rigorously test the EffortLogger against known risks and errors.
- This will involve comprehensive testing to ensure that the risk mitigation measures are effective and do not introduce new issues.

4.2.2. Merging the planning poker tool

- The next step is to integrate the planning poker tool to allow users to plan projects their project better

4.3. Parking Lot for Items we need to address moving forward

4.3.1. Security Audits

- Regular security audits and testing will be utilized to ensure data breaches are hard to come by, moreover, make sure every feature has a robust role access behavior to prevent unauthorized behavior.

4.3.3. Cross-Functional Team Collaboration

- EffortLogger V2.0 development requires collaboration between cross-functional teams, including developers, security experts, and end-users. Ensuring effective communication and cooperation is crucial in addressing risks and achieving project success.

5. Appendix A: Credit Sheet

Team Member Name	Contributions
Team Member 1 Alma Babbitt	Prototype 1 Risk Reduction 1 Conclusion Activity BenchMark
Team Member 2 Trevor Huss	Prototype 5 Combined Project Video Risk 5 Risk Reduction 5
Team Member 3 Karryl Dumalag	Prototype 2 Risk 2 Risk Reduction 2
Team Member 4 Zachary Litwin	Prototype 3 Use cases
Team Member 5 Ishan Yelnoorkar	Risk 4 Prototype 4 Conclusion

6. 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