CSE 360 Project Report Number 6

Team Th64

Team Member Names:

1. William Adams
2. David Williams
3. Quan Nguyen
4. Jaden Thomas
5. Evan Gittens

Table of Contents

Table of Contents

1. The Problem to be Addressed

1.1. Target Organizations

- A firm experiencing exponential growth, requiring a more robust version of the EffortLogger tool

1.2. The Problem

1. **Employee Privacy and Data Security:** The firm is pressured to provide more detailed data on productivity and defect rates, raising concerns about employee privacy. The existing system uses a third party tool for anonymizing user data, but this does not offer sufficient protection against potential security breaches.
2. **Confidential Information Security:** There are concerns about competitors using hacking methods and accessing the firm's confidential information about planning and operations. This could provide an unfair and unethical look into the firm's sensitive information.
3. **Enterprise-Scale Support for Agile and Quality:** The firm’s system solutions have grown significantly, necessitating a more generic system product capable of supporting enterprise-scale agile teams. This growth led to the creation of a Quality Assurance Engineering organization with ties to the firm's human resources and customer services.

2. A Proposed Solution

2.1. Solution Overview

The proposed solution for the EffortLoggerV2 is to maintain the core functionality of the original EffortLogger while incorporating additional features. These enhancements are aimed at addressing the specific needs and problems identified by the customers and stakeholders. This solutions has major focuses on:

1. **Support for Enterprise-Scale Operations:** Upgrading the EffortLogger to handle larger scale operations that the firm is undertaking.
2. **New User Interface Overhaul:** A complete redesign of the user interface to make it more intuitive and efficient for the users.
3. **Improved User Story Based Estimations:** Enhancing the system's ability to provide better estimations based on user stories.
4. **Planning Poker Tool:** Introducing a new planning poker tool to aid in agile project planning and management.

2.2. Major Solution Scenarios

- **User Story Sorting:** The system will shoe relevant user stories from the user's backlog, sorted by their average weight based on prior input. Users can modify these weights on the fly based on the importance of each user story.
- **Planning Poker Card Sharing:** A feature where users, during a planning poker session can generate and share a stylized "card" that will be sent to their team's applications, enhancing collaboration and communication among team members.

2.3. Activity Diagram

3. Requirements

3.1. User Stories

1. **As a project manager:** I want to be able to access detailed productivity and defect data without compromising employee privacy or productivity, so that I can find areas of improvement while maintaining a level of respect with my team members.
2. **As a team member:** I need a secure way to log my efforts and defects, ensuring my personal data is protected, this way I can feel safe sharing my information.
3. **As a quality assurance engineer:** I require a tool that handles enterprise-scale operations, allowing me to efficiently monitor data across a large number of teams and projects.

3.2. Operational Requirements

1. **Security:**
   a. Passwords are required for accessing the EffortLogger application and certain functionalities. This addresses the major security weakness of missing authorization checks.
   b. An encrypted password file should be stored separately from the application further enhancing security measures.
   c. Data should be stored on a database server to allow for remote access cutoff, password changes, or disconnection from the internet in case of a cyber-attack.
2. **Data Collection and Analysis:**
   a. The database should store detailed information, such as user stories, effort reports, defect reports, team, and project information. This encompasses tracking down user stories and managing large data volumes associated with large scale enterprise-level data.
   b. Metrics need to be determined to measure team performance and ensure employee anonymity.
3. **User Interface:**
   a. EffortLogger and Planning Poker applications should be run on a web application server.
   b. The interface should include a “Share” button, stylized after a poker card, with functionality to display the card on every team member’s screen during planning poker sessions.

3.3. Quality Requirements

- The requirements that are crucial in maintaining the quality of the EffortLogger tool include security, efficient data management, and a user-friendly interface. These aspects must remain as the top concern to maintain a high level of quality.

4. Architecture

4.1. Architectural Overview

The architecture of EffortLoggerV2 is meticulously designed to provide a transformative user experience, combining efficacy with robust security protocols. it represents a harmony of components, each contributing to the overall functionality of the tool. Key aspects include integrated project management and agile support and prioritizing a seamless user experience.

4.2. Architectural Elements and Rationale

1. **User Interface:**
    a. Utilizes JavaFX for design, ensuring an intuitive and user-friendly experience.
    b. Facilitates the creation, editing, and viewing or effort logs, defect logs, and planning poker sessions.
    c. A well designed interface is crucial for user satisfaction and system usability, enhancing user productivity by making interaction with core functionalities easy.
2. **Security and Privacy Layer:**
    a. Shrouds data in secrecy during transmission and storage
    b. Employs cutting-edge data storage solutions and retrieval functionality, enabling agile data extraction and secure information storage.
    c. Ensures data privacy and confidentiality through the use of anonymization and access control measures, providing users with settings to manage their data privacy preferences.
    d. Data privacy is a critical concern for users and stakeholders alike, its assurance builds trust and is an imperative part of complying with legal and regulatory requirements.
3. **Data Management:**
    a. The system is designed to enable applications to send and receive data from the database server.

b. Employees can interact with each other’s screens during planning poker sessions which allows for collaboration and effective communication.

5. Detailed Design

5.1. System Context and Interactions

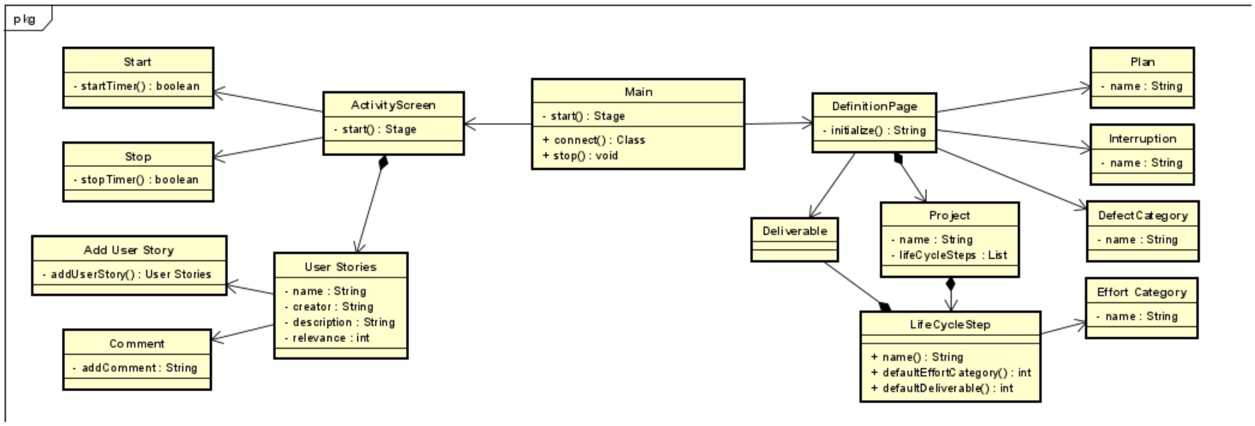
1. **User Authentication:**
   - For new users, the EffortLoggerV2 will initially ask for a new, unique username and password where each of them has specific requirements. Upon being created, the new user's data will be stored in the database.
   - For returning users, they will have their own unique private password. Upon entering the password, it will be run through a SHA-256 hash before comparing with other hashes in the database.
2. **Project Planning:**
   - Users can utilize the EffortLoggerV2 tool to specify a variety of project components through defined customization, e.g: a large pool of unique life cycle steps.
   - Moreover, in order to closely keep track of the project, users can also add in different parts such as plans, deliverables, interruptions, and defect categories.
3. **Activities Log:**
   - Along with the Project Planning, EffortLoggerV2 also provides a customized UI system for clients where all the discussions and comments made before, during, and after the project will be recorded.
   - Team members can add in User Stories prior to the meeting and it will be saved into the user's database, where he/she can have the option to edit, delete, share to others.
4. **Database:**
   - The backend of the tool utilizes a relational database, MySQL, to store all the information. As any information regarding username, password, recordings, and projects will be stored securely in the database and can be accessed at any time.
   - Users will be able to access the database with limited access, having only what the user is shared or created by him/herself visible. Only those holding the roles of Director can access the full database.

5.2. Use Cases

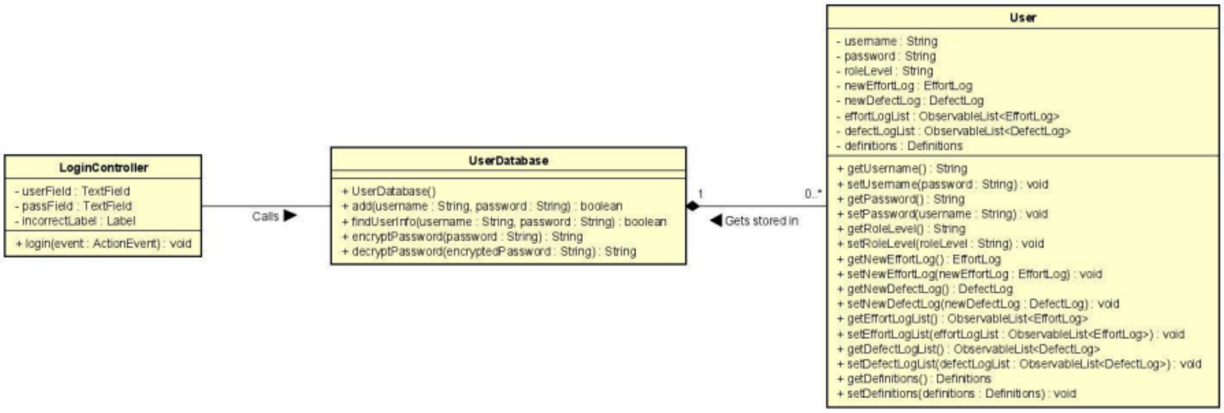
- Use Case 1: A new intern of the company tries to register in EffortLogger V2 after joining an on-going project. The intern inputs a username over 16 characters and password over 16 characters with at least one number and one special character. The tool runs a check with the database to see if the username and password fit the requirements and that the user doesn’t already exist. Once passed, the program will encrypt the intern’s password and then add the information into the database.

- Use Case 2: The same intern, after successfully creating a new account then wants to see the details of the project so he/she clicks the Definitions Page and finds the project by searching from a drop-down list. Upon seeing the project, the intern double-click on it and below will be the details of the project, including life cycle steps, plans, deliverables, defects, and interruptions.
- Use Case 3: The project leader wants to conduct a planning poker session, so the team starts to put in user stories, including the intern. The intern accesses the Activity Screen and chooses to create a new User Story. After defining it, the intern hit create and view others user stories from the list. Each one he/she tries to leave a comment to note down.

5.3. Class Diagram



5.4. Supporting UML Models and Diagrams


LoginController
- userField : TextField
- passField : TextField
- incorrectLabel : Label
+ login(event : ActionEvent) : void
Calls ▶
UserDatabase
+ UserDatabase()
+ add(username : String, password : String) : boolean
+ findUserInfo(username : String, password : String) : boolean
+ encryptPassword(password : String) : String
+ decryptPassword(encryptedPassword : String) : String
1
0..*
◀ Gets stored in
User
- username : String
- password : String
- roleLevel : String
- newEffortLog : EffortLog
- newDefectLog : DefectLog
- effortLogList : ObservableList<EffortLog>
- defectLogList : ObservableList<DefectLog>
- definitions : Definitions
+ getUsername() : String
+ setUsername(password : String) : void
+ getPassword() : String
+ setPassword(username : String) : void
+ getRoleLevel() : String
+ setRoleLevel(roleLevel : String) : void
+ getNewEffortLog() : EffortLog
+ setNewEffortLog(newEffortLog : EffortLog) : void
+ getNewDefectLog() : DefectLog
+ setNewDefectLog(newDefectLog : DefectLog) : void
+ getEffortLogList() : ObservableList<EffortLog>
+ setEffortLogList(effortLogList : ObservableList<EffortLog>) : void
+ getDefectLogList() : ObservableList<DefectLog>
+ setDefectLogList(defectLogList : ObservableList<DefectLog>) : void
+ getDefinitions() : Definitions
+ setDefinitions(definitions : Definitions) : void

6. Implementation

6.1. Structure and Naming

The project was made in the application package and uses a variety of entity classes to populate the application database which is controlled by the ApplicationController and LoginController classes. The methods within the controller classes are controlled by the GUI attributes with buttons, combo boxes, text fields, and etc.

6.2. Verification, Validation, and Testing

In order to verify, validate and test our application, we had to do extensive debugging especially on the GUI since it's a bit hard to automate GUI. Lots of problems stemming from the GUI were either just logic problems within the action methods or simply something needing to be updated in scenebuilder. But we were also able to do some automated unit testing with JUnit, especially when verifying valid usernames and passwords.

6.3. GitHub Repository and Contents

GitHub Link: https://github.com/RAdams-RST/CSE360-Th64/tree/main

Contents: Contains the main branch that holds our final version of EffortLogger V2 as well as our individual branches that we used to work on the project.

7. Demonstration

7.1. Overview

The demonstration of EffortLoggerV2 includes several key functionalities showcased through various screencasts, providing a visual and interactive representation of the system capabilities.

https://asu.zoom.us/rec/share/TIvVD7UoKFX2m30CBx0tKRUiceJl8zI9Llg1zrOd2Zpsd0hpyT3gJB3rmNf_J97c.HS1ZP69BoCbWxjDJ

Passcode: y!ixM9zL

7.2. Key Features

1. **Log-in Page:**
    a. https://asu.zoom.us/rec/share/iQV-6knB8PmP16PIUTckDVp1m7elmcawRBXiTdCZlQJ2vHmOsfJgDrKW2P6LcvLc.FSUqNZs0yPLfW7nL?startTime=1701411722000

    Passcode: +%eDjeg5

2. **Effort Log Console:**
    a. https://asu.zoom.us/rec/share/Km8CwwLxNeMvwOo0vQIYwfkZfuSeCARstFbXJPXk3UTBkFat6EB0yv6cwbj_5VGz.jzLgYk5_H8gtwErw

    Passcode: f#tZ@L48

3. **Definitions Page:**
    a. https://app.screencast.com/fj3pquprsKJJi

8. Conclusion

8.1. Overview

The EffortLoggerV2 project successfully integrated key functions to enhance both security and usability, focusing on user-friendly navigation and efficient project management. This includes improved security through the use of password hashing and user data management, as well as features for collaborative comment and user story management.

8.2. Lessons Learned

While there were many lessons learned along the way there are a few highlights that come to mind with a project of this proportion.

1. **Enhanced Security Measures:** The importance of aligning with modern privacy standards through functions like password hashing was a key learning point. This not only prevents unauthorized access but also enhances the overall system security. This was an excellent learning experience.
2. **Challenges with Collaboration Tools:** Throughout the course of this project our team had various difficulties working with GitHub as our main collaboration tool, this aspect of the project was more challenging than anticipated. The difficulties of managing a project of this size improved our communication as a team and helped us move forward in our careers as software engineers.
3. **Data Management and Navigation:** Managing large volumes of data and ensuring efficient navigation through large amounts of entries was a significant challenge. The development of prototypes to mitigate risks associated with data display and navigation was a critical part of the learning experience.

8.3. Recommendations for Improvement

1. **Refinement of User Interface:** Continued focus on user interface design to enhance the overall user experience and usability. This includes improving

user interactions with the systems core functionalities for enhanced productivity.

2. **Further Security Enhancements:** Ongoing review and improvement of security practices, especially in light of evolving cyber threats. This may include the integration of more advanced encryption methods and regular security audits.
3. **Expanding Data Management Capabilities:** Developing more sophisticated methods for handling large volumes of data, ensuring data is displayed to the user correctly and can be navigated efficiently.

9. Appendix A: Credit Sheet

Team Member Name	Contributions
William Adam	Coded Effort Log Page and Creation Planning Poker
David Williams	Coded Defect Log Creation Wrote section 6
Quan Nguyen	Wrote Section 5 Created the UML Diagrams
Jaden Thomas	Wrote sections 1, 2, 3, 4, and 8 Designed the Definitions Page within EffortLoggerV2
Evan Gittens	Setup Database Coded Definitions Page

10. Appendix A: Current Team Norms

Team Norms (Th64)

https://drive.google.com/file/d/1Qnkm8RHfKUr1U7AdqQGyj2LmRGIHDsVK/view?usp=drive_link

Team Norms (Th64)

Introduction

This contract specifies the terms by which group **Th64**, formed of the five members:

1. **William Adams**
2. **David Williams**
3. **Evan Gittens**
4. **Quan Anh Nguyen**
5. **Jaden Thomas**

agree to work as a team through the development process of the CSE 360 Fall 2023 Group Project. This contract consists of 5 sections:

1. **Goals**
2. **Communication Norms**
3. **Work Norms**
4. **Decision-Making Norms**
5. **Signatures**

Goals

1. Score high on deliverables.
2. Improve individual communication and collaboration skills.
3. Apply topics learned in class in a practical environment.
4. Build a cohesive and communicative team.
5. Maintain respect between all members of the team.

Potential Obstacles to Goals, With Solutions

1. *One or multiple members become incapacitated*

- Acute Solution: The work originally placed upon that member will be distributed across the remaining available members to lessen the impact.
- Preemptive Solutions:
    - All members of the team are expected to openly communicate any blocks in their schedule where their availability will decline from the status quo.
    - All members of the team are expected to work on their assigned section of an assignment ahead of the due date. This will lessen the impact if a member is incapacitated near the due date of an assignment.

2. *There is a significant decline in team morale*

- Solution: The team will regularly meet in person as well as online to maintain internal relationships.

3. *Communication is poor or sparse*

- Solution: Communication will be made as frictionless as possible with frequent meetings and communication through Discord/Slack.

4. *Meetings or individual work is plagued with distractions*

- Solution: Meetings will be scheduled ahead of time so members may remove any distractions they know of ahead of time.

5. *The team experiences harsh internal conflicts*

- Solution: Internal conflicts will be raised early. Team members are expected to maintain respect with each other in the event of a conflict. If it is not able to be handled individually, the team will put it up to a vote.

Communication Norms

1. There will be a fifteen minute informal in-person meeting after class each Thursday.
2. Supplementary meetings, averaging about thirty minutes each, will be scheduled as needed throughout the week between members collaborating on their assigned section of an assignment.
3. Communication will primarily take place over discord, in a designated server to parse through relevant discussions and maintain a log of our communications.
4. Members are expected to show up to meetings as they are available.
5. Members are also expected to contribute significantly at these meetings for a full grade in both the “attended meetings” and “contributes to group discussions” category.

Work Norms

1. The number of individuals assigned to a section in a deliverable will vary based on the complexity of the deliverable.
    - The "complexity" of the deliverable will be determined by the length in pages, as well as how many section prerequisites it requires at a given point in time. Complexity is greater the greater the length is in pages. Complexity is lower when it has more incomplete prerequisites.
        - For example, a twelve-page section would have many people assigned to it, unless it could only be completed at the end, in which case it would gradually have more people assigned as sections got completed.
2. The informal due date we will hold ourselves to is Saturday at twelve pm (noon). Any changes made past this due date must not affect the work of those who have declared they're done working. In order to get full points in the “work completed on time” category, team members need to complete their work by this time or communicate ahead of time why it will take them until later.
3. If any member(s) has an issue with another member(s) lack of communication or the speed at which they're getting an assignment done, they will bring it up through three stages:
    1. Individually
    2. With the team as a whole
    3. With a TA

Escalation from stage two to stage three will only occur if no explanation or improvement has been provided after the next two deliverables.

4. Members are expected to make significant contributions to the project while maintaining a positive attitude in communications with the team in order to receive full points in the "contributions were significant to the team's success," "maintained a positive and cooperative attitude with teammates" and the "work met or exceed the team's agreed-to norms and expectations" categories.

Decision Making Norms

1. Conflicts require the consensus of everyone who will be directly affected by the issue at hand.

- e.g. If one member has an issue with the spacing in a section, they should discuss it with their section collaborators. If they have an issue with the format of the deliverable as a whole, it must be brought up to the team as a whole.

2. If one person cannot let go of an individual idea, it will go to a group vote.