CSE 360 Project Report Number 6

Team Tu57

Team Member Names:

1. Andrew Boban
2. Andy Vu
3. Vedant Kaushik
4. Venkata Sai Kaushik Pattela
5. Vikas Mejari

Table of Contents

Table of Contents

1. The Problem to be Addressed

1.1. Target Organizations

The first target organization was originally a manufacturing firm, but the current product has expanded to include any target organization that works on major projects and requires effort and defect logging for these projects. This could include most engineering firms, as well as software development firms. These firms could be small and be composed of small teams or could be large and composed of larger teams that may utilize collaborative work to complete their projects. The product should be designed to serve the needs of both the users as well as the supervisors in these teams.

1.2. The Problem

The problem can be separated into three categories: security and privacy, accurate and concise display of information, and scalability. Security and privacy are the most important categories. Users should be confident in the product's abilities to keep their sensitive information private from other users, their supervisors, and outside attackers. Users should also have the choice of sharing their information to people that they need to share information to, such as team members. Regarding supervisors, making sure that as much useful information is provided to supervisors as possible while ensuring that no identifying information is made available to supervisors is essential for ensuring the anonymity of users, which is a key feature of this product. Displaying information in a concise and accurate manner is also an important part of the problem. Right now, users report that searching for information using the currently implemented product is time-consuming. By streamlining this process and displaying the information in a quick-to-access and easy-to-understand format, users will experience a quality-of-life enhancement. Also ensuring that users can quickly filter relevant and irrelevant information is also helpful for the user experience. This is specifically helpful in situations such as planning poker, where users might need to find data from relevant historical projects to reference and use when planning future projects. Displaying relevant data concisely is also helpful to supervisors. Supervisors would use this information to make decisions on where to allocate resources and what approaches should be utilized, so having this information be more accessible and digestible would allow supervisors to perform their jobs far more effectively. Scalability is the last category and is also important in ensuring that the product will satisfy the needs of the customers. Ensuring that the product can be scaled up to serve the needs of larger firms as well as the needs of firms that have cross-team projects is integral in ensuring that the target customer base is larger and can include larger firms on top of the smaller firms that are already being supported. The success of the product hinges on its ability to address three primary categories: security and privacy, accurate and concise information display, and scalability. Among these, security and privacy take precedence. Users must trust that the product can safeguard sensitive information from other users, supervisors, and external threats. Enabling users to selectively share information with designated team members ensures a balance between privacy and collaboration. Anonymization techniques play a crucial role, allowing supervisors to access valuable insights without compromising the anonymity of individual contributors. User-managed sharing settings further empower users to control access, reinforcing confidence in the product. In terms of displaying information, the focus is on improving efficiency and user experience. Advanced search algorithms and intuitive filtering options reduce the time users spend navigating the system. Customizable dashboards provide a personalized view of relevant metrics, aligning with individual preferences and priorities. The inclusion of a historical data reference feature aids in situations like planning poker, enabling users to quickly retrieve and compare data from past projects. This not only benefits individual contributors but also assists supervisors in making informed decisions about resource allocation. Scalability is integral to accommodating a diverse customer base. Cloud-based infrastructure offers flexibility, allowing resources to scale based on demand. A modular architecture enhances adaptability, enabling the product to evolve alongside changing requirements. Cross-team collaboration features are crucial for firms engaged in collaborative projects, ensuring seamless communication and data sharing across teams. This comprehensive approach aims to create a product that not only meets current needs but anticipates and accommodates future challenges and growth. By addressing security concerns, refining information display, and ensuring scalability, the product strives to be a versatile and effective solution for a broad spectrum of users and organizations.

2. A Proposed Solution

2.1. Solution Overview

The application to solve the problems listed shall consist of two main parts, a login interface and an information display interface. The login interface will require the user to enter in a username and password, and then will check the username and password entered to see if it is a valid combination. If the username and password combination is not valid, the user shall not be allowed access. If the user and password combination is valid, the application shall then move to the information display interface. Additionally, the user can create a username password combination to register as a new user. To bolster the security of the application, the login interface should automatically lock if the user attempts to log in unsuccessfully too many times. The information interface displays information about the historical projects related to the signed in user, and allows the user to add, edit, delete, and search for data. The user will only have access to the projects they have been assigned to or the ones they have worked on; all other projects are hidden from the user and are inaccessible.

2.2. Major Solution Scenarios

2.2.1 Enhanced Security and Privacy Measures

User Authentication: Implement robust user authentication mechanisms to ensure only authorized individuals have access to sensitive information.

Selective Data Sharing: Develop a user-controlled data sharing feature, allowing users to selectively share information with team members while maintaining privacy from supervisors and other users.

Encryption: Employ advanced encryption techniques to safeguard data during transmission and storage, enhancing overall security.

2.2.2 Optimized Information Display

Streamlined Search Functionality: Redesign the search functionality to be more intuitive and time-efficient, addressing user concerns about the current time-consuming process.

Concise Data Presentation: Optimize the user interface to present information in a concise and easy-to-understand format, improving the overall user experience.

Quick Filtering: Implement quick filtering options for relevant data, especially during planning poker sessions, enabling users to access historical project data swiftly.

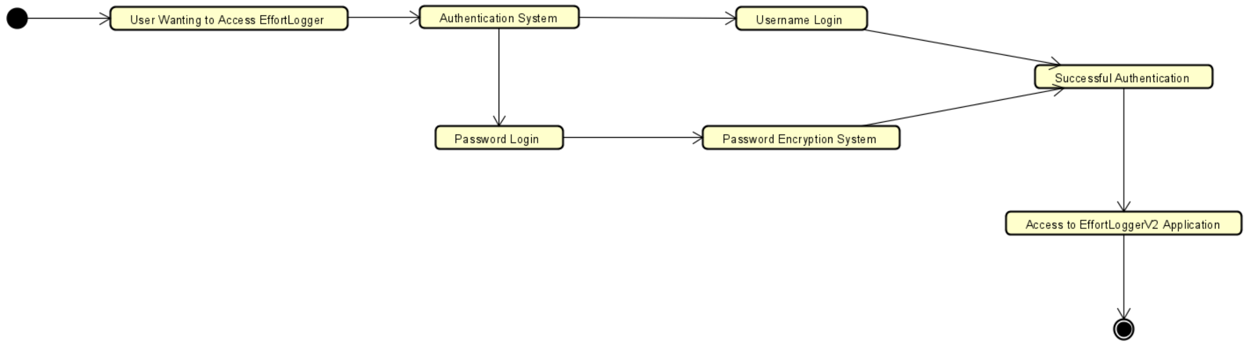
2.2.3 Scalability and User Anonymity

Scalable Architecture: Design the product architecture to accommodate the varying needs of different organizations, ensuring scalability for both small and large teams.

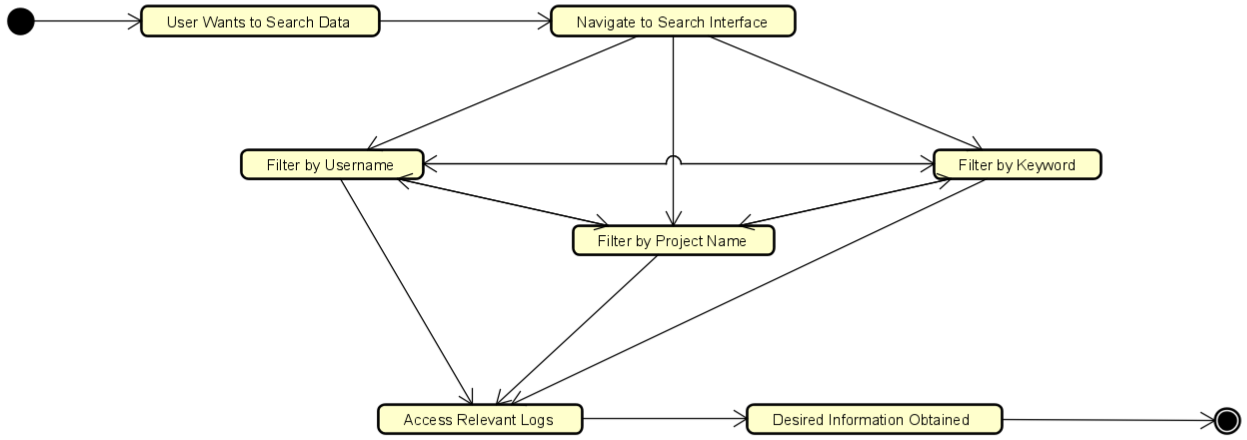
Supervisor Decision Support: Develop features that provide supervisors with relevant project information without compromising user anonymity, aiding effective decision-making.

2.3. Activity Diagram

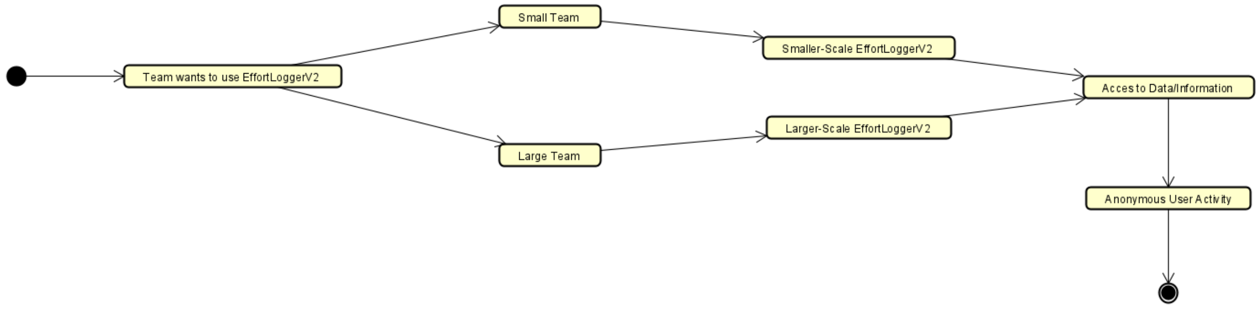
2.3.1 Enhanced Security and Privacy Measures


User Wanting to Access EffortLogger
Authentication System
Username Login
Successful Authentication
Password Login
Password Encryption System
Access to EffortLoggerV2 Application

2.3.2 Optimized Information Display


User Wants to Search Data
Navigate to Search Interface
Filter by Username
Filter by Keyword
Filter by Project Name
Access Relevant Logs
Desired Information Obtained

2.3.3 Scalability and User Anonymity


Small Team
Team wants to use EffortLoggerV2
Smaller-Scale EffortLoggerV2
Acces to Data/Information
Larger-Scale EffortLoggerV2
Large Team
Anonymous User Activity

3. Requirements

3.1. User Stories

3.1.1. As a user, I should be able to access information regarding historical projects that I have worked on, such as the user stories related to the historical project, the story points assigned to each items, the estimated effort and list of defects associated with the project, etc.

3.1.2. As a user, I should be able to label my historical projects with keywords that are relevant to the historical project and should be able to edit or delete the keywords associated with a historical project if the understanding or information regarding that project changes.

3.1.3. As a user, I want to be able to search through historical projects based on keywords to find projects similar to the current project I'm working on. By searching for a specific keyword, only historical projects I've worked on that are associated with that keyword should be displayed.

3.1.4. As a user, I want to be able to enter in a written description of the historical projects I've worked on, and for those written descriptions to be saved. I should be able to access those descriptions later and should also be able to edit them if necessary.

3.1.5. As a user, I want specific information related to the effort of a historical project. This information should include things such as the overall amount of effort, how the effort was distributed among the various categories, etc.

3.1.6. As a user, I also want information related to the defects associated with a historical project. This information should include things such as which part of the product contained the defect, when the defect was introduced, when the defect was discovered, the amount of effort required to address the defect, etc.

3.1.7. As a user, I want to be able to share information about my historical projects to other users. I should be able to choose which users I want to share this information with, and I should also be able to choose what information I share to the chosen users.

3.1.8. As a user, I should be able to hide historical projects that are irrelevant to the current project I'm working on. If I choose to hide a historical project, it should no longer be displayed. I should also be able to undo this action and redisplay any historical projects I've hidden.

3.1.9. As a user, I should be able to login to my account using a username password combination. If I try to log in with an invalid username password combination, I should be denied. I should also be guaranteed that no one is able to access my account and the information inside my account without knowing my username password combination.

3.2. Operational Requirements

Operating System Compatibility:

The Effort Logger program should be compatible with major operating systems, including Windows (7 and above), macOS, and Linux, to ensure accessibility for a wide range of users.

Hardware Requirements:

Minimum hardware requirements should include 4 GB RAM and a dual-core processor to ensure smooth operation of the program. Adequate free disk space should also be specified for data storage.

Network Requirements:

The program should be designed to work efficiently over standard internet connections with a recommended minimum speed of 5 Mbps to ensure quick and reliable data entry and retrieval for users.

Security and Authentication:

User authentication must be implemented using a secure username-password combination. Failed login attempts should result in denial of access. The program must ensure that user accounts and associated information are secure, and unauthorized access is prevented.

Backup and Recovery Procedures:

Regular automated backups of user data should be performed, and users should be assured that their historical project information is securely stored. A clear process for data recovery in case of system failure or data loss should be outlined.

Performance Metrics:

The program should be designed to handle a specified number of simultaneous users, ensuring responsive performance during peak usage times. Response times for data entry and retrieval operations should be within defined limits.

User Interface and Experience:

The user interface should be intuitive, allowing users to easily access, edit, and delete historical project information. The program should provide a seamless experience for labeling projects with keywords, searching based on keywords, and hiding or displaying relevant historical projects.

Sharing and Privacy Controls:

Users should have the ability to share information about their historical projects selectively. The program should provide granular control over shared information, allowing users to choose specific users to share with and specifying the information shared.

Historical Project Information Storage:

Written descriptions of historical projects entered by users should be stored securely. Users should have the capability to access, edit, and delete these descriptions as needed.

Defect Tracking and Effort Information:

The program should efficiently store and retrieve information related to defects associated with historical projects, including details such as defect location, introduction, discovery, and effort required for resolution. Effort distribution among project categories should also be tracked.

3.3. Quality Requirements

Proper Data Display: Ensuring that data is displayed accurately is one of the most important requirements for ensuring the quality of the product. This means that even as the data being displayed becomes more comprehensive and includes more information about the historical projects, the data should always be properly displayed. The formatting of the display should also be easily readable and intuitive so that users can access the information quickly.

Security and Privacy: Making sure that the security and privacy of the product is maintained is also important in ensuring the quality of the product. Making sure that the product maintains the privacy of users through rigorous testing is necessary for quality assurance. This testing can be done by ensuring that invalid username password combinations do not allow a user to access any information, that every username password combination only goes to the corresponding account and not to any other account, and that information from accounts are separate from one another.

Proper Data Flow: Making sure that the user inputs do exactly what they are supposed to and allowing users to add, delete, and edit data while maintaining the integrity of the data is key to ensuring that users can utilize the product. This means that when the user interacts with the product in order to add, delete, and edit data, the functions should always behave as expected, and no unusual behaviors should occur. The application should also give proper outputs to the user so that the user is informed that the action is successful.

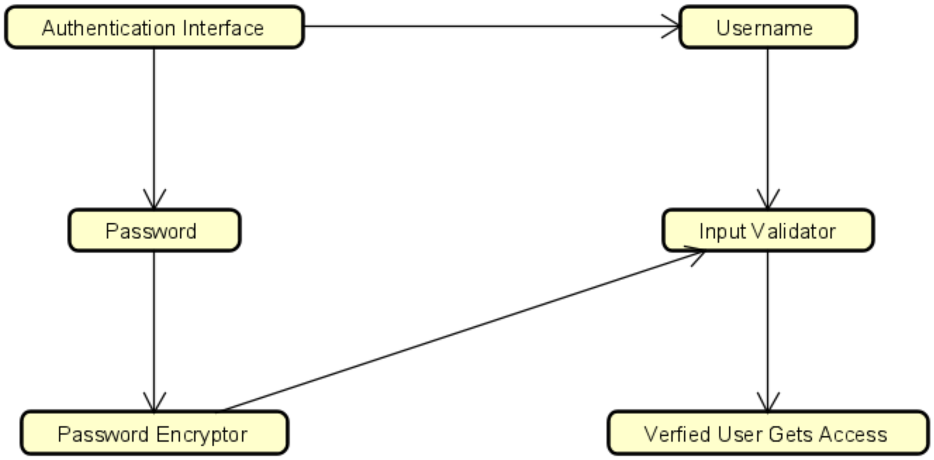
4. Architecture

4.1. Architectural Overview

The proposed architecture is designed to address the multifaceted challenges identified in the problem statement. It integrates key elements to enhance security, streamline information display, and ensure scalability. The architecture emphasizes user control over data sharing and maintains user anonymity while providing supervisors with valuable decision-making insights.

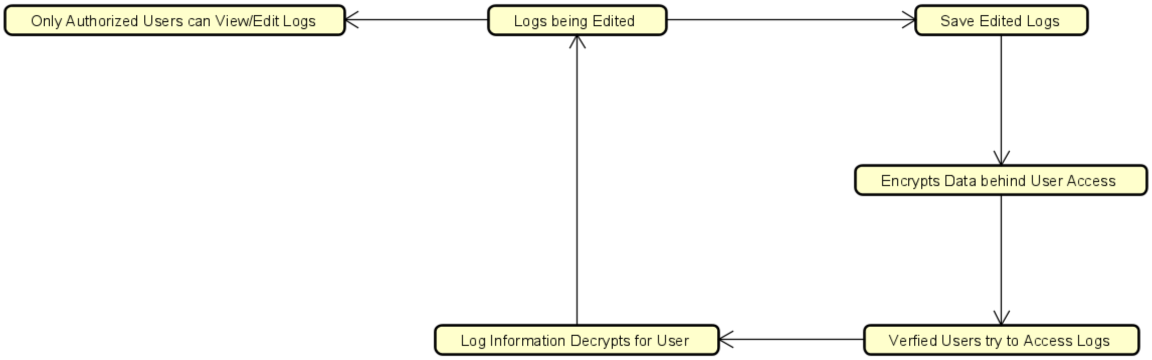
4.2. Architectural Elements and Rationale

User Authentication Module: This is a foundational component in EffortLogger's architecture, ensuring secure access, user accountability, and compliance with data privacy standards. It establishes a robust security perimeter, contributing to the overall reliability and trustworthiness of the system.



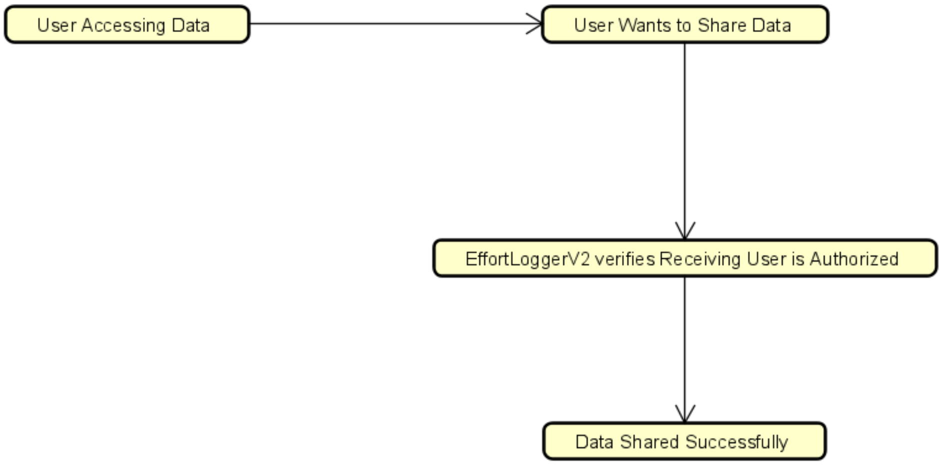
Rationale: Ensures secure access to the system by implementing precise user authentication mechanisms. This is vital for protecting sensitive information from unauthorized access.

Data Encryption Component: This component is pivotal in fortifying the security posture of EffortLogger. It safeguards sensitive data, ensures compliance with privacy regulations, and enhances the overall trustworthiness of the system, contributing to its reliability and integrity.


Only Authorized Users can View/Edit Logs
Logs being Edited
Save Edited Logs
Encrypts Data behind User Access
Log Information Decrypts for User
Verfied Users try to Access Logs

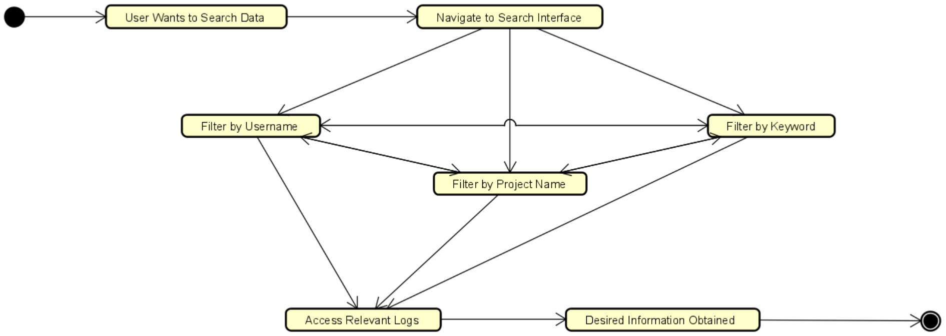
Rationale: Incorporates advanced encryption techniques to secure data during transmission and storage. This element safeguards sensitive information, maintaining the privacy and integrity of user data.

Selective Data Sharing Feature: This will be a strategic addition to EffortLogger's architecture, offering a flexible and controlled approach to data access. It ensures that organizations can adapt to the unique needs of their projects and teams while prioritizing data privacy, security, and compliance with industry regulations.



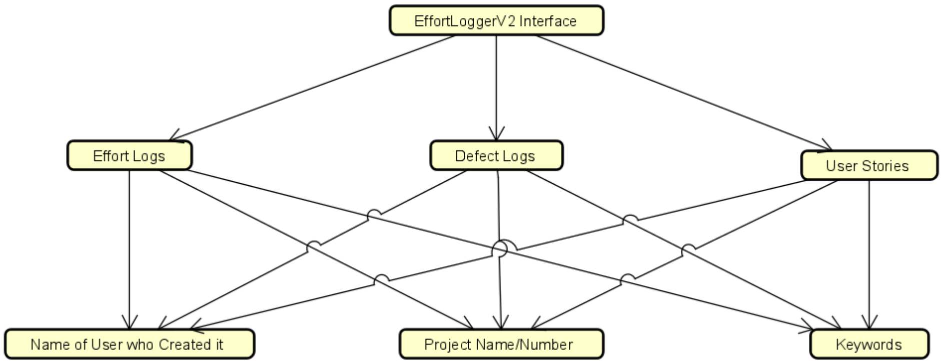
Rationale: Empowers users to control the sharing of their information. Users can selectively share data with team members, striking a balance between collaboration and individual privacy.

Streamlined Search Functionality: Key Element that provides users with an efficient and user-friendly means of accessing project information. This also contributes to the overall effectiveness of the platform by optimizing data retrieval processes and enhancing the user experience.


User Wants to Search Data
Navigate to Search Interface
Filter by Username
Filter by Keyword
Filter by Project Name
Access Relevant Logs
Desired Information Obtained

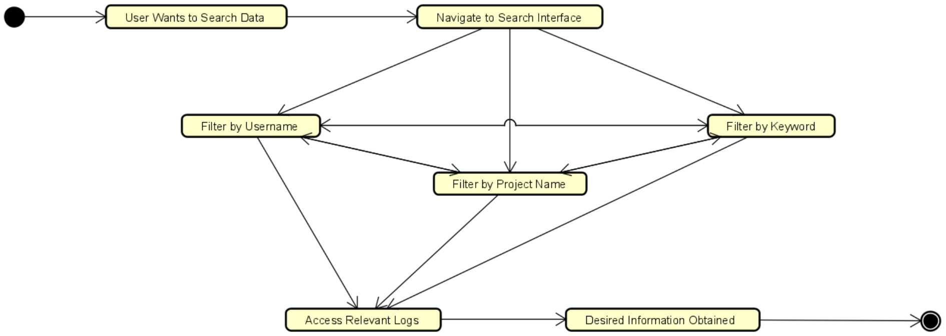
Rationale: Addresses user concerns about time-consuming searches. The streamlined search functionality ensures that users can quickly and efficiently retrieve relevant information, improving the overall user experience.

Concise Data Presentation Interface: This will be optimizing the way project-related information is displayed and facilitating efficient user interaction while also enhancing user comprehension, decision-making, and engagement, ultimately contributing to the effectiveness and user satisfaction with the platform.



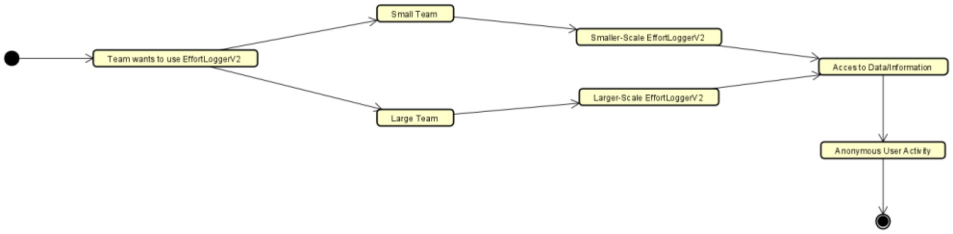
Rationale: Optimizes the user interface to present information in a clear and concise manner. This enhances user comprehension and facilitates quick decision-making during planning poker sessions.

Quick Filtering Options: This promotes efficient data exploration and empowers users to focus on the information that matters most to them. This feature also aligns with user-centric design principles and enhances the overall effectiveness and usability of the platform.


User Wants to Search Data
Navigate to Search Interface
Filter by Username
Filter by Keyword
Filter by Project Name
Access Relevant Logs
Desired Information Obtained

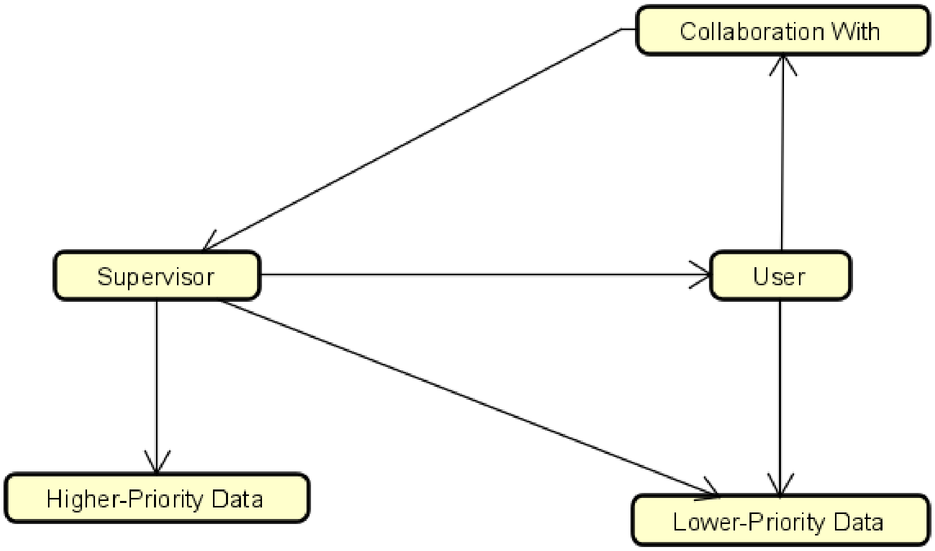
Rationale: Implements quick filtering options for relevant data. Especially beneficial during planning poker sessions, this feature allows users to access historical project data swiftly, contributing to more informed estimations.

Scalable Architecture Design: Foundational to EffortLogger's ability to grow, adapt, and deliver a reliable and responsive experience to users. It supports the platform's evolution, ensures efficient resource utilization, and positions EffortLogger to meet the dynamic demands of project management environments.


Team wants to use EffortLoggerV2
Small Team
Large Team
Smaller-Scale EffortLoggerV2
Larger-Scale EffortLoggerV2
Accesto Data/Information
Anonymous User Activity

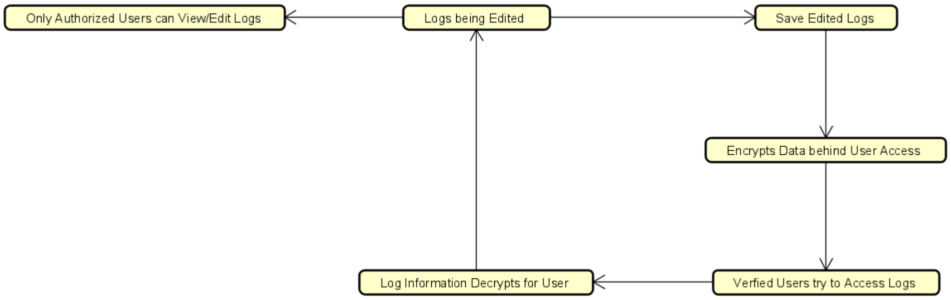
Rationale: Ensures scalability to accommodate the diverse needs of different organizations. The architecture is flexible enough to support both small teams in manufacturing firms and larger teams in engineering or software development firms.

Supervisor Decision Support Features: This fosters data-driven decision-making and enhances the overall effectiveness of project management. These features contribute to a proactive and adaptive approach, allowing supervisors to lead projects with confidence and optimize outcomes based on real-time insights.



Rationale: Develops features that provide supervisors with valuable project information without compromising user anonymity. This supports supervisors in making informed decisions about resource allocation and project approaches.

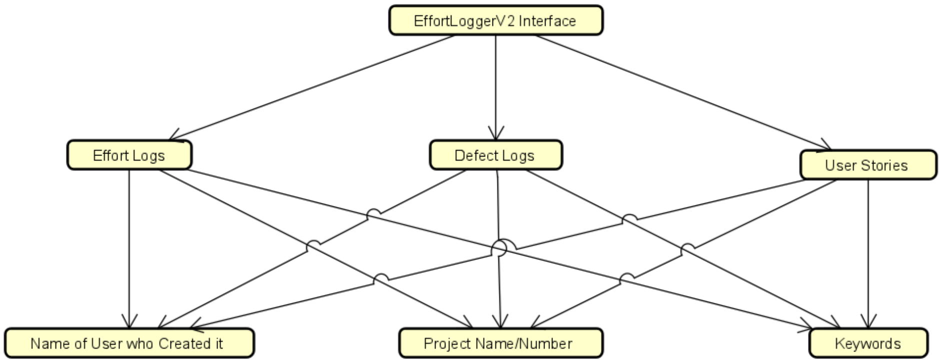
Activity Logging and Monitoring: Logging activities promotes accountability, transparency, security, and continuous improvement efforts. These features contribute to the overall reliability, performance, and user satisfaction with the platform, aligning with best practices for robust project management solutions.


Only Authorized Users can View/Edit Logs
Logs being Edited
Save Edited Logs
Encrypts Data behind User Access
Log Information Decrypts for User
Verfied Users try to Access Logs

Rationale: Implements comprehensive activity logging and monitoring to track user actions within the system. This contributes to security measures and enables auditing for accountability.

Integration with Planning Poker Module: The Planning Poker Module promotes collaborative and accurate effort estimations. This integration aligns with agile principles, enhances user engagement, and contributes to transparent and efficient project planning processes.

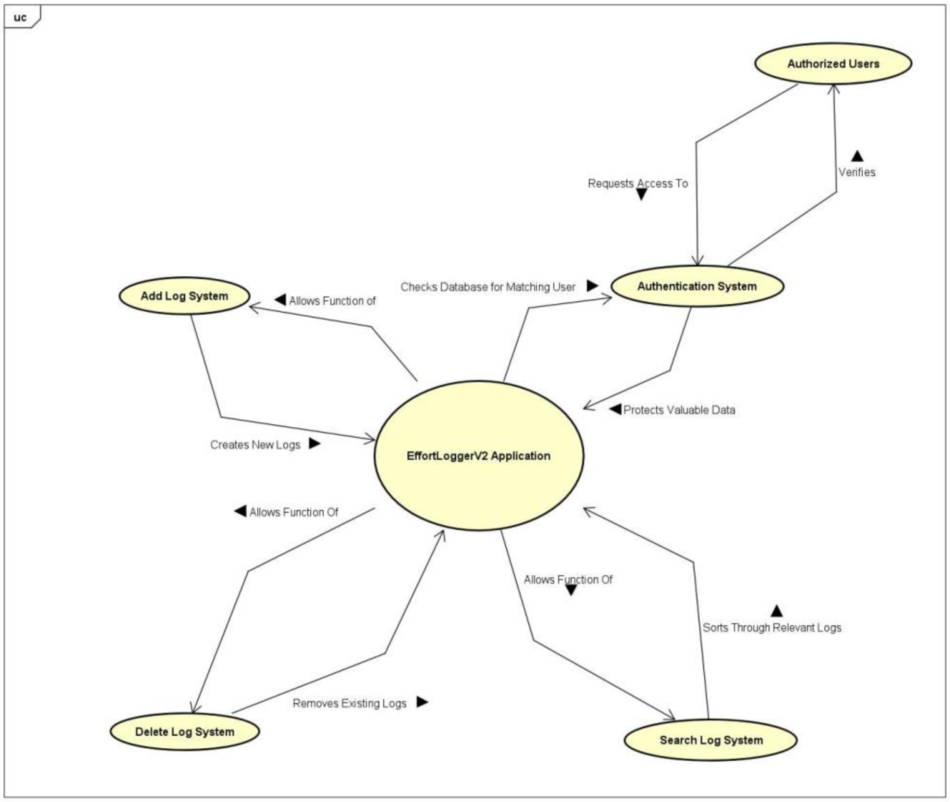
Architecture



Rationale: Seamless integration with the Planning Poker module to ensure that historical project data is readily available during estimation sessions, contributing to more accurate and informed estimations.

5. Detailed Design

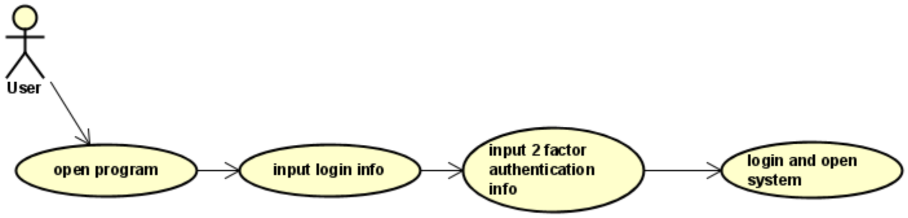
5.1. System Context and Interactions



The EffortLoggerV2 System works on the principle of three main functions, those being the Add Log System, Delete Log System, and Search Log System. In addition, the authorization system is a core component of EffortLogger, and allows us to address those issues of privacy and security when it comes to valuable user data. The architectural elements are all addressed under this model, and flow into our detailed design process. EffortLoggerV2 employs robust user authentication mechanisms to ensure that only authorized users can access the system. Also taking place within the authentication system is the encryption of data such as login passwords. User data, project details, and other sensitive information are encrypted, safeguarding them from unauthorized access and ensuring confidentiality. The model includes features that allow selective sharing of project data based on user roles and permissions, which fall under the authorized users segment. Users can also use the Search Log System to efficiently search for specific projects, tasks, or team members, enhancing the overall user experience and facilitating data retrieval. EffortLoggerV2's interface, which is displayed upon successful authentication, ensures that project-related data is presented in a concise and user-friendly manner, improving readability and supporting effective decision-making. Within the Search Log System itself, users can apply filters based on various criteria such as project name, team member name, or keywords, enabling them to quickly isolate and analyze relevant information. The scalable architecture ensures that EffortLoggerV2 can handle varying workloads and adapt to changing demands without compromising performance. Supervisors have the power to dictate who is authorized to have access to EffortLoggerV2, as well as make informed decisions, identify trends, and optimize project management strategies. Finally, EffortLoggerV2 includes activity logging in the form of the Add and Delete Log Systems in order to monitor components to track user actions, project activities, and system performance.

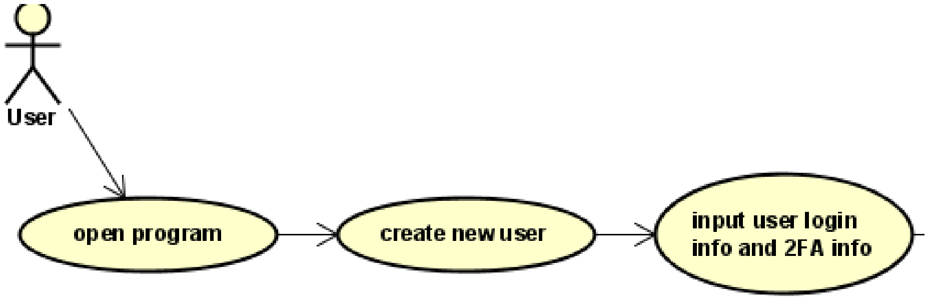
5.2. Use Cases

Use case 1: logging into system



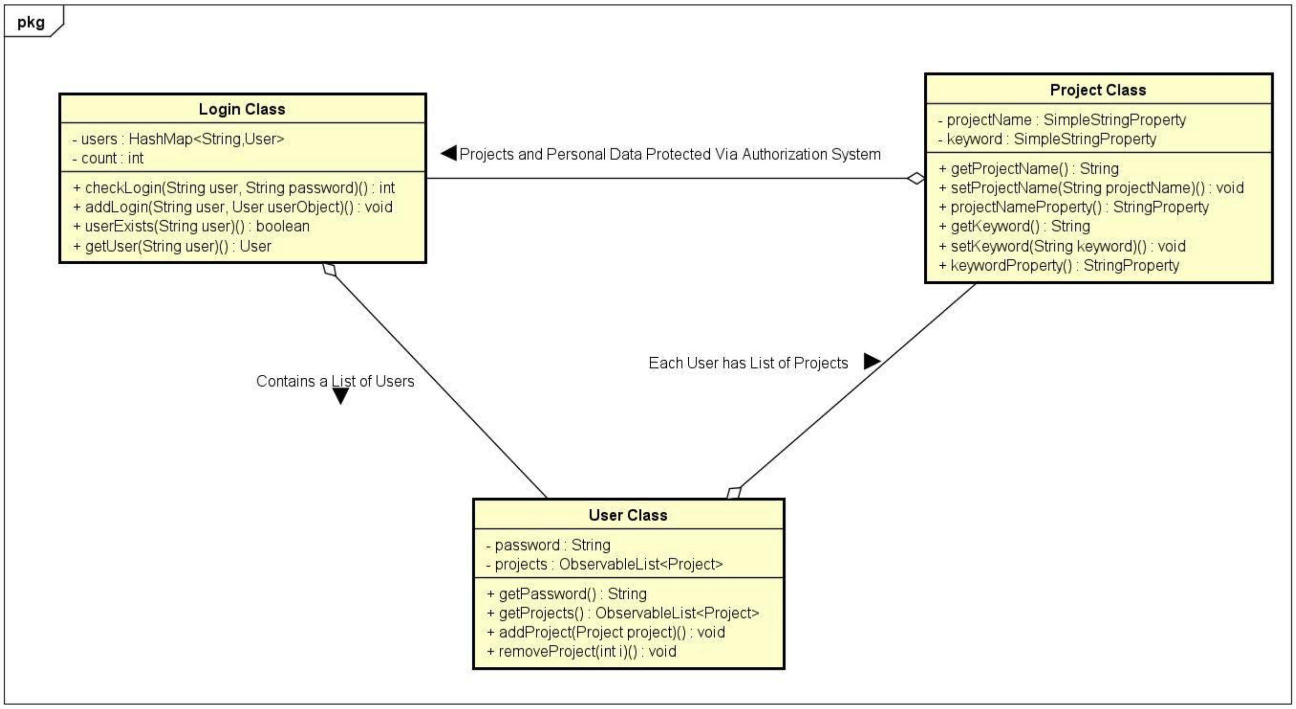
In this diagram a user opens the effort logger program and then inputs their info. Then a window to the 2 factor authentication opens up. The user then inputs the answer to their personal question linked to the user. Once a user hits login they are welcomed to the system.

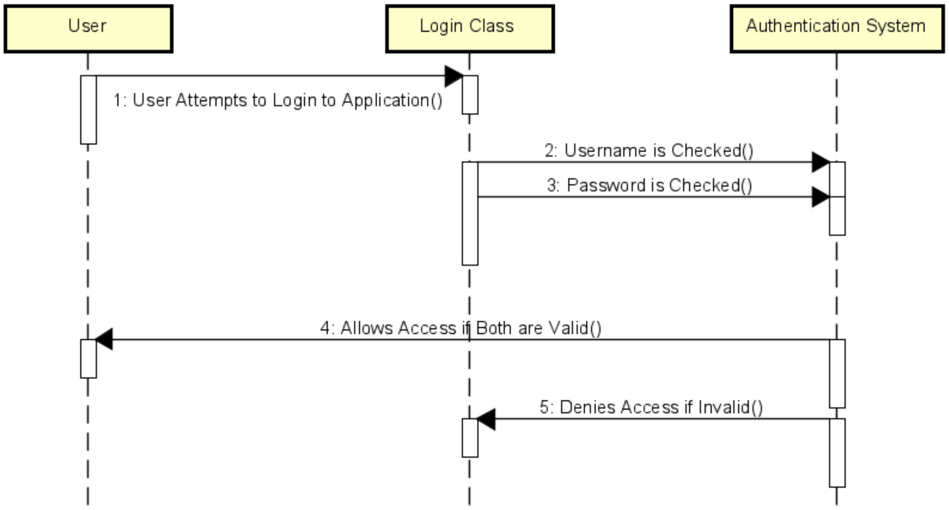
Use case 2: creating new user

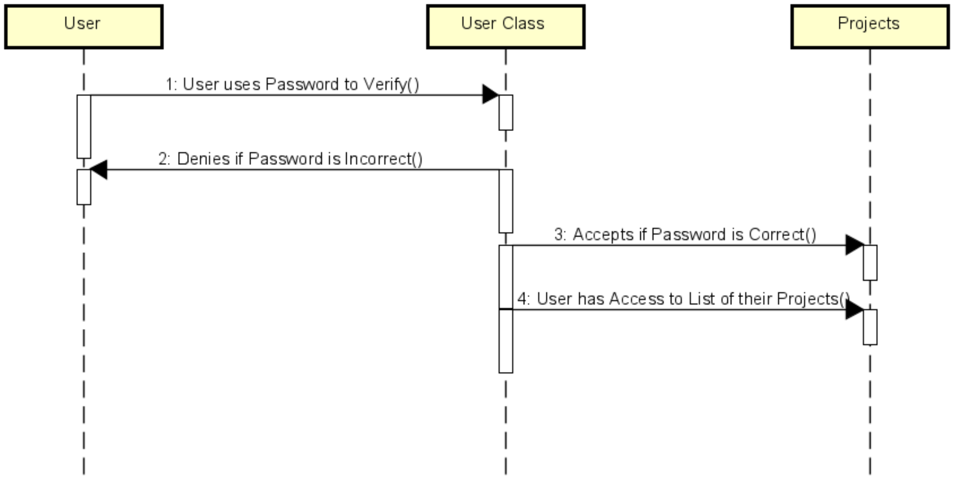


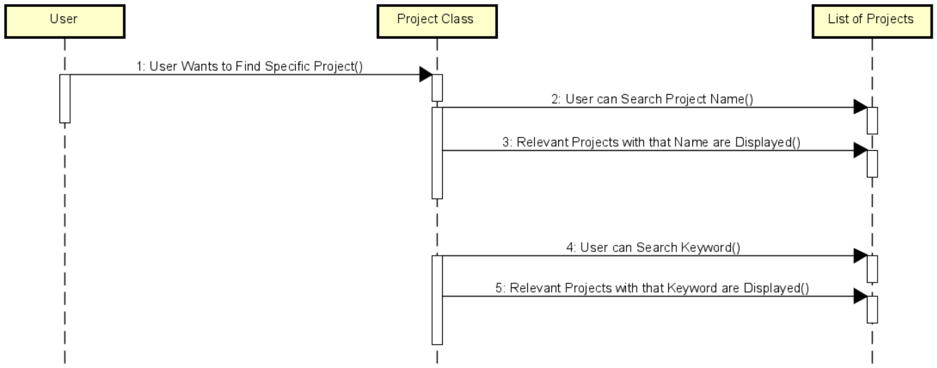
In this diagram a user is creating a new account in effort logger. The user opens the program and then hits a create user button. They are then prompted to input login info and 2 factor authentication info.

5.3. Class Diagram

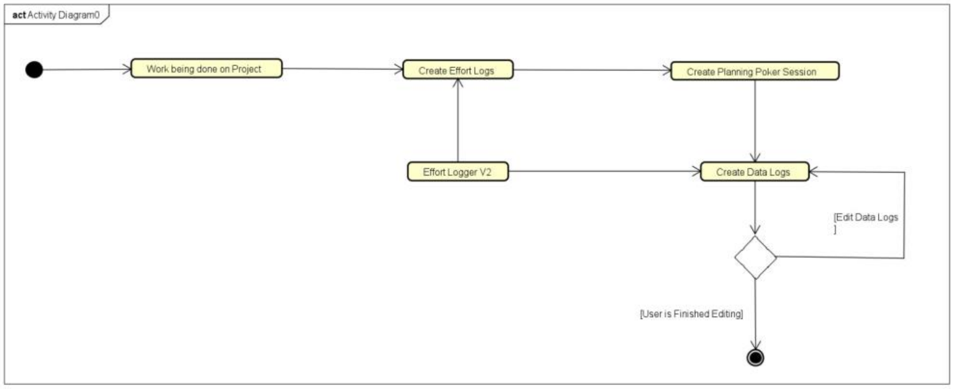

pkg
Login Class
- users : HashMap<String,User>
- count : int
+ checkLogin(String user, String password)() : int
+ addLogin(String user, User userObject)() : void
+ userExists(String user)() : boolean
+ getUser(String user)() : User
Projects and Personal Data Protected Via Authorization System
Project Class
- projectName : SimpleStringProperty
- keyword : SimpleStringProperty
+ getProjectName() : String
+ setProjectName(String projectName)() : void
+ projectNameProperty() : StringProperty
+ getKeyword() : String
+ setKeyword(String keyword)() : void
+ keywordProperty() : StringProperty
Contains a List of Users
Each User has List of Projects
User Class
- password : String
- projects : ObservableList<Project>
+ getPassword() : String
+ getProjects() : ObservableList<Project>
+ addProject(Project project)() : void
+ removeProject(int i)() : void

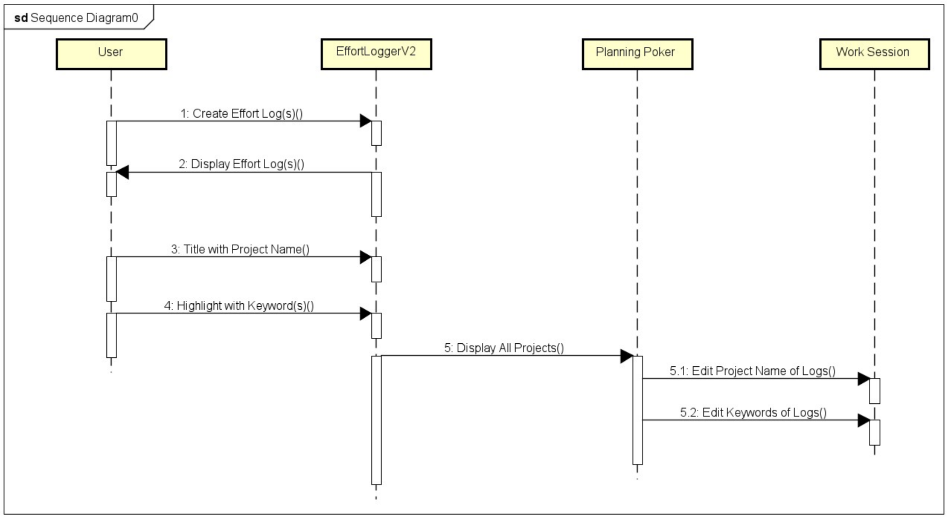

User
Login Class
Authentication System
1: User Attempts to Login to Application()
2: Username is Checked()
3: Password is Checked()
4: Allows Access if Both are Valid()
5: Denies Access if Invalid()


User
User Class
Projects
1: User uses Password to Verify()
2: Denies if Password is Incorrect()
3: Accepts if Password is Correct()
4: User has Access to List of their Projects()


User
Project Class
List of Projects
1: User Wants to Find Specific Project()
2: User can Search Project Name()
3: Relevant Projects with that Name are Displayed()
4: User can Search Keyword()
5: Relevant Projects with that Keyword are Displayed()

5.4. Supporting UML Models and Diagrams




sd Sequence Diagram0
User
EffortLoggerV2
Planning Poker
Work Session
1: Create Effort Log(s)()
2: Display Effort Log(s)()
3: Title with Project Name()
4: Highlight with Keyword(s)()
5: Display All Projects()
5.1: Edit Project Name of Logs()
5.2: Edit Keywords of Logs()

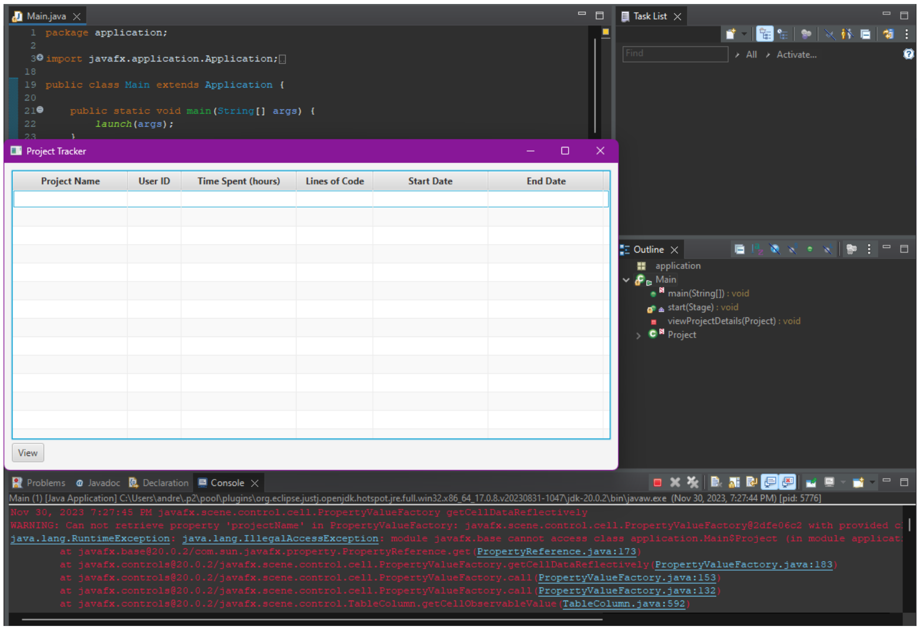
6. Implementation

6.1. Structure and Naming

The application consists mainly of two scenes, the login scene and the display scene. The login scene simply consists of a username text field and a password text field. The user should input their information into the text field and press the button to get to the next scene, which is the display scene. This is the majority of the application. The display screen consists of a table of projects associated with the logged-in user. There is a project name text field and a keyword text field, which will allow for users to add, delete, and search for historical projects. There are a series of buttons that will allow the user to choose which function is activated. The structure is relatively simple. There is a project object, which basically consists of information relating to the project. Then there is the user object, which consists of the user's password, as well as a list of project objects that the user has worked on. The last part of the product is the login interface, which has a list of users, as well as buttons and text fields which allow for the user to input information and interact with the system. This part is the front end of our system, and is the only part that the user should see.

6.2. Verification, Validation, and Testing

While testing our program, we ran into multiple problems, shown below is one of the largest problems that we overcame. As you can see, the program although creating the box and showing the titles and sections of each cell, is completely invisible, and you can see that the view button is fine, the cells are produced normally, but we were not getting the output that we wanted for this part of the program. We ended up overhauling the whole section of the program, to include a way to not only insert, but edit and delete AND search for projects based of keywords and such. The program now works completely seamlessly and we do not have to worry about visibility or anything else.



```
package application;

import javafx.application.Application;

public class Main extends Application {

    public static void main(String[] args) {
        launch(args);
    }
```

Project Name	User ID	Time Spent (hours)	Lines of Code	Start Date	End Date

```
Main (1) [Java Application] C:\Users\andre\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.8.v20230831-1047\jdk-20.0.2\bin\javaw.exe (Nov 30, 2023, 7:27:44 PM) [pid: 5776]
Nov 30, 2023 7:27:45 PM javafx.scene.control.cell.PropertyValueFactory getCellDataReflectively
WARNING: Can not retrieve property 'projectName' in PropertyValueFactory: javafx.scene.control.cell.PropertyValueFactory@2dfe06c2 with provided c
java.lang.RuntimeException: java.lang.IllegalAccessException: module javafx.base cannot access class application.Main$Project (in module applicat
	at javafx.base@20.0.2/com.sun.javafx.property.PropertyReference.get(PropertyReference.java:173)
	at javafx.controls@20.0.2/javafx.scene.control.cell.PropertyValueFactory.getCellDataReflectively(PropertyValueFactory.java:183)
	at javafx.controls@20.0.2/javafx.scene.control.cell.PropertyValueFactory.call(PropertyValueFactory.java:153)
	at javafx.controls@20.0.2/javafx.scene.control.cell.PropertyValueFactory.call(PropertyValueFactory.java:132)
	at javafx.controls@20.0.2/javafx.scene.control.TableColumn.getCellObservableValue(TableColumn.java:592)
```

6.3. GitHub Repository and Contents

https://github.com/Narukam-1/CSE360-Program

We used the GitHub repository to pass around and work on our code for the Effort Logger Project. In GitHub you will find various push and pull requests where we have made changes to incorporate, change or even delete some functions within our code.

Our utilization of the GitHub repository has been instrumental in the collaborative development of the Effort Logger Project. Within the repository, a meticulous history of our team's contributions unfolds through a series of push and pull requests. These requests encapsulate a spectrum of modifications, ranging from the seamless incorporation of new features to meticulous alterations and, on occasion, the removal of redundant functions within our codebase.

GitHub's branching mechanism has afforded our team the ability to concurrently work on diverse aspects of the project, utilizing branches to isolate and implement specific features or bug fixes. The ensuing pull requests encapsulate the meticulous process of code review, where proposed changes undergo scrutiny before integration into the main branch. This meticulous approach not only streamlines the integration of code but also serves as a robust quality assurance checkpoint.

The issue tracking system within GitHub has proven invaluable in the organizational aspect of our workflow for the Effort Logger Project. Task identification and prioritization are systematically managed through the creation and assignment of issues, fostering a structured approach to project management. The transparent nature of this system ensures that each team member is cognizant of the project's overarching goals and the specific tasks assigned to them.

Furthermore, the repository's integration with continuous integration (CI) tools automates our testing processes, ensuring that each code modification undergoes rigorous automated testing before integration. This pre-emptive identification and resolution of potential issues contribute significantly to the overall reliability and stability of our codebase. In essence, GitHub has not merely served as a code repository; it has functioned as an intricate and indispensable ecosystem that has propelled the collaborative development of the Effort Logger Project to new heights.

7. Demonstration

7.1. Overview

User authentication is implemented using secure mechanisms to ensure access to the platform by prioritizing privacy and security. This includes a login system with a username and password validator that checks if input is valid or not. Sensitive information, such as user credentials and project details, is encrypted both during transmission and storage using encryption algorithms. Activity Logging and Monitoring components are integrated to track user actions, project activities, and system performance. The Effort Logs are stored securely, and monitoring tools provide real-time insights into the system's status. Additionally, the Search Functionality is implemented using search field system, allowing users to search for projects based on the project names or keywords related to a certain project.

7.2. Key Features

Andrew Boban - https://drive.google.com/file/d/1O_brHWMOlohx9PXY5hPg-8bMykv6bPcs/view?usp=sharing (overhauled display system)

Vedant Kaushik -
https://drive.google.com/file/d/1zYLeHPQuIzQ7896vEuiRJTmUM5YCNygw/view?usp=sharing

Andy Vu - https://drive.google.com/file/d/1iARUo24GyCZXH5kBQsPqtv_WsNOHrVwJ/view?usp=sharing

Venkata Pattela- https://asu.zoom.us/rec/share/I_8Acuu2GB71mhtYmhuebOfKtLx_tV2wGfcf1no7j-mSFg-tHCDnASe6uCq_GTU3.NrqQuKp923CX4grf?startTime=1701407890000
Passcode: DJrN5#eY

Vikas Mejari - https://drive.google.com/file/d/1CYKGin6yqaogqDGKx3HZYb8dI7-KiOMJ/view?usp=sharing

8. Conclusion

8.1. Overview

The most important aspect of the project was understanding how to make the product secure, how to ensure that the data flow behaves as expected, and how to keep testing to ensure that these properties are maintained as the product is being developed and improved. These aspects of the project can be applied to other projects as well, regardless of the nature of the project. Also, proper documentation is another important aspect of the project. Documentation is essential for allowing others to understand the functions of the product and how it is implemented and is also important for making sure the team working on the project understand what is currently being worked on, what needs to be improved, and the priority of the tasks that are currently required.

8.2. Lessons Learned

Throughout the course of this project, one of the pivotal lessons we gleaned was the significance of writing code with clear and easily understandable documentation. As we integrated our individual code contributions into the collaborative project, it became evident that some sections needed to be reworked from the ground up to seamlessly align with the overarching structure of the final product. This realization underscored the importance of comprehensive documentation, as code that lacked clarity and transparency proved to be more challenging to integrate. The effort expended in rewriting certain portions highlighted the critical role of well-documented code in facilitating collaboration and streamlining the integration process. Moreover, this experience emphasized the need for thorough testing methodologies, surpassing our initial expectations. Rigorous testing became an indispensable practice to identify and rectify discrepancies arising from the integration of diverse code modules, ensuring a cohesive and robust product. This project underscored the enduring value of writing code with meticulous documentation, not only for individual comprehension but also for the seamless amalgamation of diverse code sections.

8.3. Recommendations for Improvement

In considering recommendations for future improvement, a notable enhancement could involve revisiting the two-factor authentication (2FA) process. Instead of relying on a personal question, implementing a system where the application calls a phone number linked to the user's account and prompts them to press "#" could potentially enhance security. This modification adds an additional layer of verification by leveraging a device tied directly to the user, making it more challenging for unauthorized access. This method not only aligns with contemporary 2FA practices but also reduces the risk associated with personal questions that might be susceptible to social engineering. This improvement not only enhances the security posture of the authentication process but also aligns with evolving industry standards, ensuring a robust and reliable means of user verification for future iterations of the application.

9. Appendix A: Credit Sheet

Team Member Name	Contributions
Andy Vu	Worked on the problem introduction, the solution, and the conclusion. Also worked on implementing the login function and ensuring the code worked for that function.
Andrew Boban	Completed parts 6.2 and 6.3 Worked on the problem Completed parts 3.1 and 3.2
Vikas Mejari	Completed the activity diagrams for 2.3, as well as the Architecture and Detailed Design portions of the document
Vedant Kaushik	Did use case diagrams Worked on conclusion Line 4 Line 5 Line 6
Venkata Sai Kaushik Pattela	Worked on archetecture,Worked on 2.1, 5.1,6.2 Made the database implementation

10. Appendix A: Current Team Norms

This appendix is an updated version of the original Team Norms contract the team has created as well as a URL to an ASU Google Drive PDF of the signed copy. The graders must be able to access that PDF!

Google docs: https://drive.google.com/drive/folders/1Hv4OvU6nwKkErbRh6-vAesYKOrZsj7vy?usp=sharing