CS3354 Software Engineering
Final Project Deliverable 2

Asset Management Software

Anna Miranda
Chelsea Heredia
Vincent Nguyen
Pooja Ganapathy
Jacob Guiang
Mason Kuehne

Due: 18 November 2022

1. [5 POINTS] **Well described delegation of tasks**, i.e. who did what in the project. Now that your project is complete, you are required to submit the delegation of tasks from beginning of the project until the end. Please make sure to fairly distribute tasks in the team and remember that in the end of the semester, each member of a team will receive the same grade. See grading policy below for more detail. If no/poor contribution by a member, please specify clearly so that we can grade each student fairly.

Anna Miranda: Sequence Diagrams and Class Diagram (deliverable 1); Test Plan and Unit Test Code (deliverable 2); Objective, Sequence Diagram, and User Interface Demo (presentation)

Chelsea Heredia: Software requirements list - 7 functional, all non-functional types and class diagram (deliverable 1); Conclusion and software requirements presentation slides (deliverable 2).

Mason Kuehne: what software process model to use and why, class diagram (deliverable 1); cost, effort, and pricing estimation (deliverable 2).

Vincent Nguyen: creating multiple use case diagram for users (deliverable 1); comparison of software (deliverable 2); use case diagram slides (presentation)

Jacob Guiang: Architectural design (deliverable 1 and presentation); comparison of software to other designs (deliverable 2)

Pooja Ganapathy: Project Scope document and class diagram (deliverable 1), project scheduling & project timeline (deliverable 2)

2. [10 POINTS] **Everything required and already submitted in Final Project Deliverable 1**. Please specify this part as “Project Deliverable 1 content”.

BEGIN PROJECT DELIVERABLE 1 CONTENT

1. [5 POINTS] Please attach here the **Final Project draft description (that contains the instructor feedback)**. It is ok to include a picture of the original document.
Address the feedback provided for your proposal by listing what you did / plan to do to comply with those proposed changes and or requests for additions to your project.

Project Description:
We will be developing asset management software. Some of its features will be handling different user roles with different access levels, linking files together with a tagging system, and capability to invite external users to modify certain files. The software will organize and store files by a hierarchical system. The system will support importing and exporting data with different file types. There will be a search tool that would allow searching by keywords & phrases. The system will also be able to support SQL queries. Another feature would be implementing a customizable data visualization tool within the system.

Instructor Feedback:

Good choice for a topic! Asset management software is a promising tool that will help users wisely and confidently. Lots of useful inherent parts are mentioned such as database search feature, hierarchical storage, ability to allocate a multi-user system which are all making your design very powerful and unique.

A great detailed task delegation, thank you for that.

In the final report, please make sure to include comparison with similar applications -if any-, make sure that you differentiate your design from those, and explicitly specify how.

2. [10 POINTS] Setting up a **Github repository**. Please use your utdallas email accounts only for each group member.

https://github.com/AMirandaUTD/3354-AssetManagement/

3. [5 POINTS] **Delegation of tasks**: Who is doing what. If no contribution, please specify as it will help us grade each group member fairly.

Create Project Scope document (similar to examples in project info doc) - Pooja Ganapathy
What software process model to use & why - Mason Kuehne
Software requirements list (5-7 functional, all non-functional types) - Chelsea Heredia
Use case diagrams (multiple) - Vincent Nguyen
Sequence diagrams (multiple) - Anna Miranda
Architectural design - Jacob Guiang
Class diagram (one only) - Mason, Anna, Pooja, and Chelsea

4. [5 POINTS] Which **software process model** is employed in the project and why. (Ch 2)

We should use the **incremental process model** because our project will end up with many different features that should be implemented at different times. For example, our first iteration may simply store user files, but iterations after that may implement a database search feature, hierarchical storage, and the ability to allocate a multi-user system. Additionally, this process model is not too hard to work with, which will benefit our team considering this is our first time working together.

5. [15 POINTS] Software Requirements including
5.a.) [5 POINTS] **Functional requirements**. To simplify your design, please keep your functional requirements in the range minimum 5 (five) to maximum 7 (seven). (Ch 4)

7 Functional Requirements that describe functionality and system services.

1. The system shall store assets and data files in a hierarchical manner, allowing the storage to be sorted by file name or data modified.
2. The system shall organize data into categories tagged by category labels.
3. A user shall be able to search the database and storage of all assets within the system by specifying a keyword or phrase.

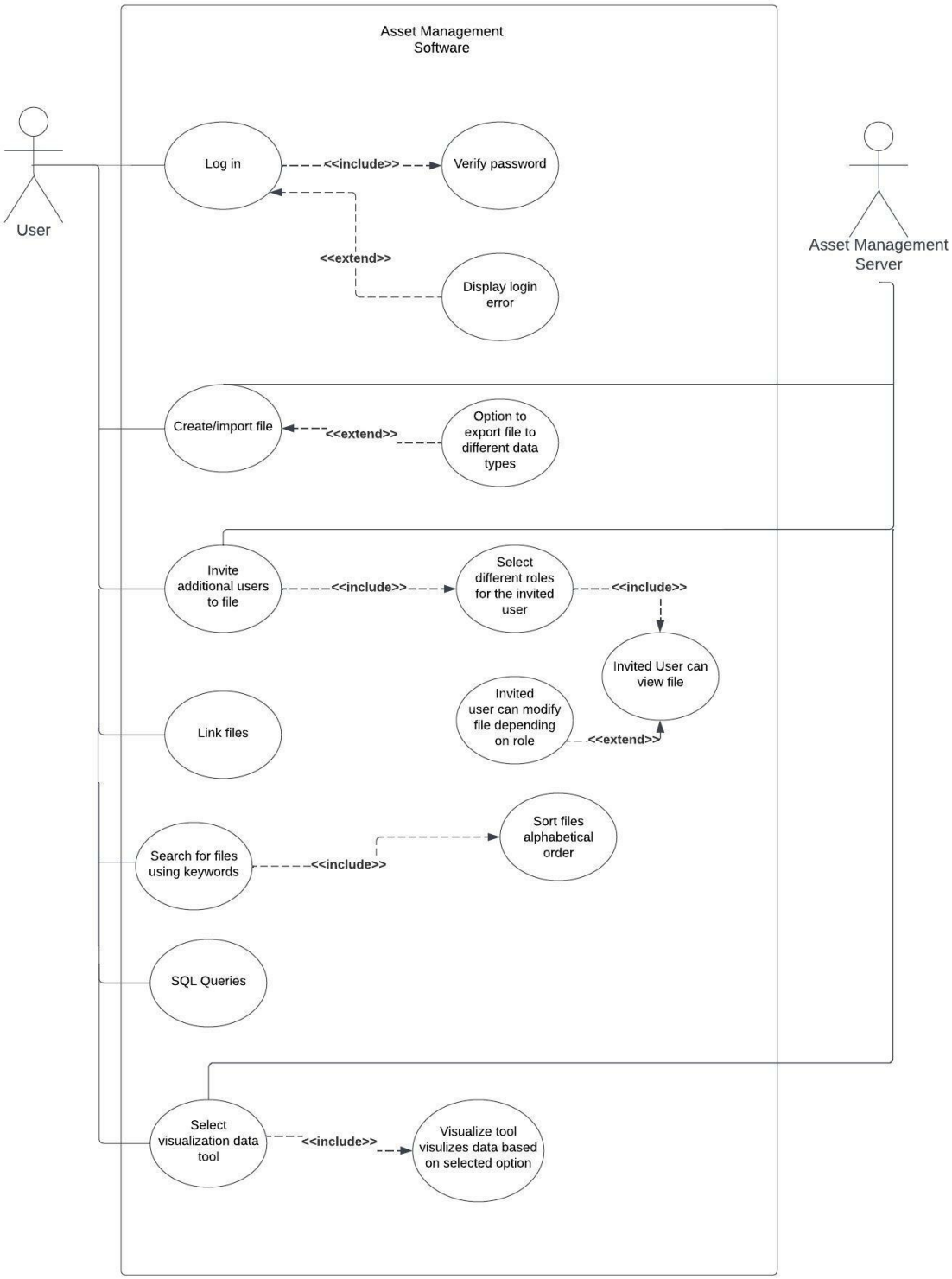
4. Each person using the system shall be uniquely identified by their identification number and login, specifying their access level.
5. The system shall only generate available files based on the cleared access level of the user.
6. The system shall support SQL queries.
7. The system shall support importing and exporting data files of different types, such as .txt, .xlsx, .pdf, .doc, etcetera.

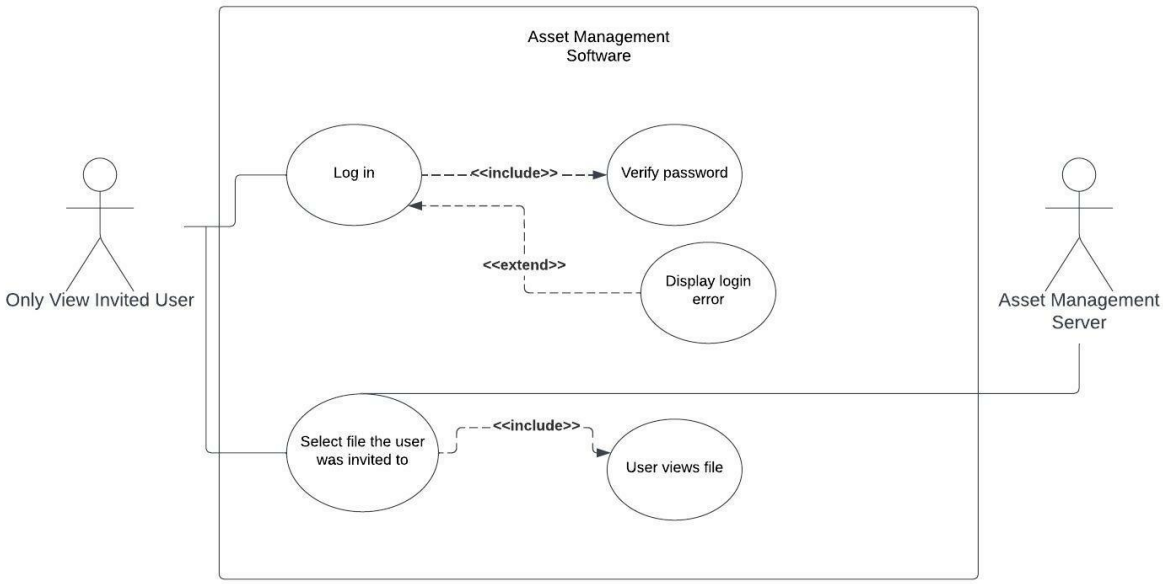
5.b.) [10 POINTS] **Non-functional requirements** (use all non-functional requirement types listed in Figure 4.3 - Ch 4. This means provide one non-functional requirement for each of the leaves of Figure 4.3. You can certainly make assumptions, even make up government/country based rules, requirements to be able to provide one for each. Please explicitly specify if you are considering such assumptions.)

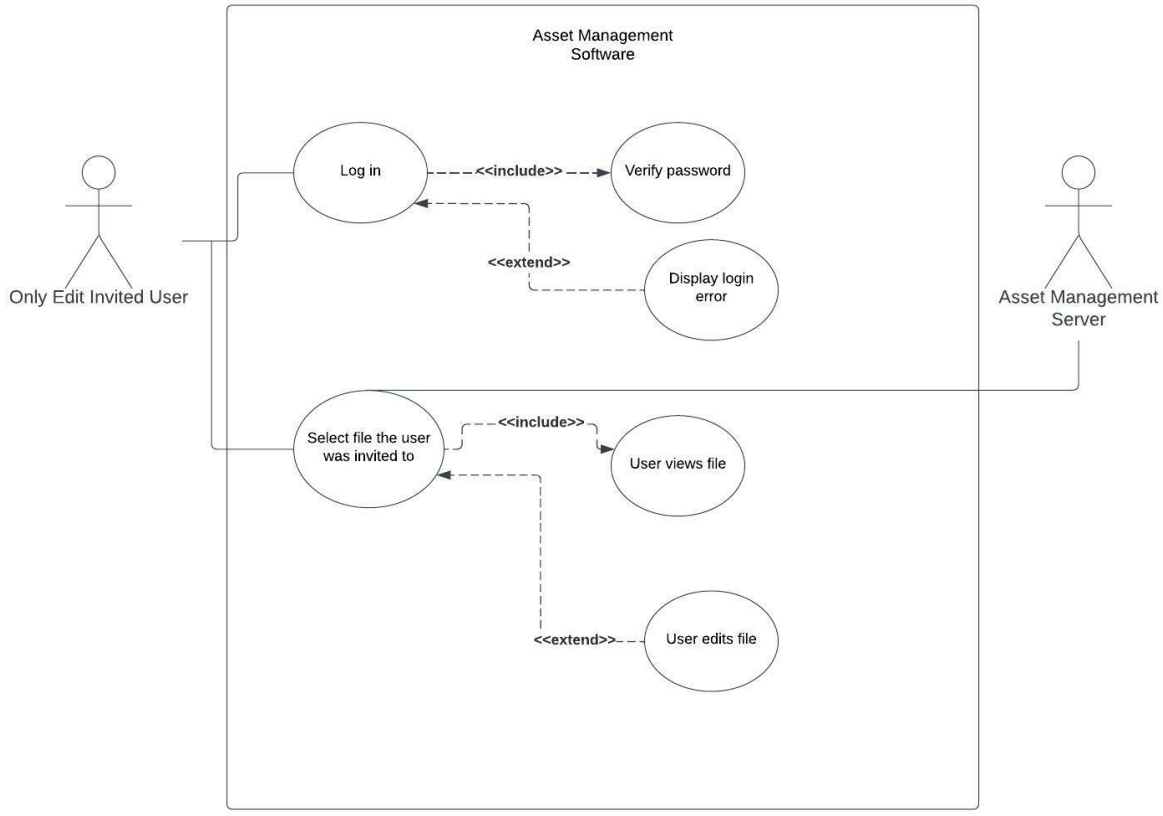
Non-Functional Requirements that define system properties and constraints.

- **Usability Requirement**: Users shall be able to customize data visualization, such as by prioritizing files or selecting a viewing theme.
- **Performance Requirement:** The system shall generate search results of data queries within five seconds of search time.
- **Space Requirement:** The asset management system shall be able to store a maximum of 1TB storage, approximating a storage of 200,000 documents.
- **Dependability Requirement:** The system shall be available to all users during normal waking hours each weekday (Mon-Fri, 05:00-23:59). Downtime shall not exceed five seconds in any one day outside of maintenance hours each week (1x/week, 00:00-04:59).
- **Security Requirement:** The system shall contain encrypted file storage and sharing.
- **Environmental Requirement:** The team shall use the Visual Studio Code IDE for local environment development, and they shall push changes to the remote repository hosted on GitHub.
- **Operational Requirement:** Users shall authenticate themselves using a secure company login, including identification PIN.
- **Development Requirement:** The product shall be developed using common web framework technologies, including but not limited to: React.js, Node.js, jQuery, and/or Express.
- **Regulatory Requirement:** The system shall allow efficient and effective modifications to development of existing datasets and files without compromising system quality or other assets, as approved by regulators in upper management.
- **Ethical Requirement**: The system shall be implemented in the highest standard acceptable to user expectations and to the general public, as abiding by the IEEE Code of Ethics.
- **Accounting Requirement:** The system's user data collection will abide by national laws enforced by the U.S. Federal Trade Commission, including but not limited to: The Children's Online Privacy Protection Act (COPPA), the Gramm Leach Bliley Act (GLBA), and the Fair Credit Reporting Act (FCRA).
- **Safety/Security Requirement:** The system shall allow user authentication and password management for user account safety and security.

6. [15 POINTS] **Use case diagram** – Provide a use case diagram (similar to Figure 5.5) for your project. Please note that there can be more than one use case diagrams as your project might be very comprehensive. (Ch 5 and Ch 7)

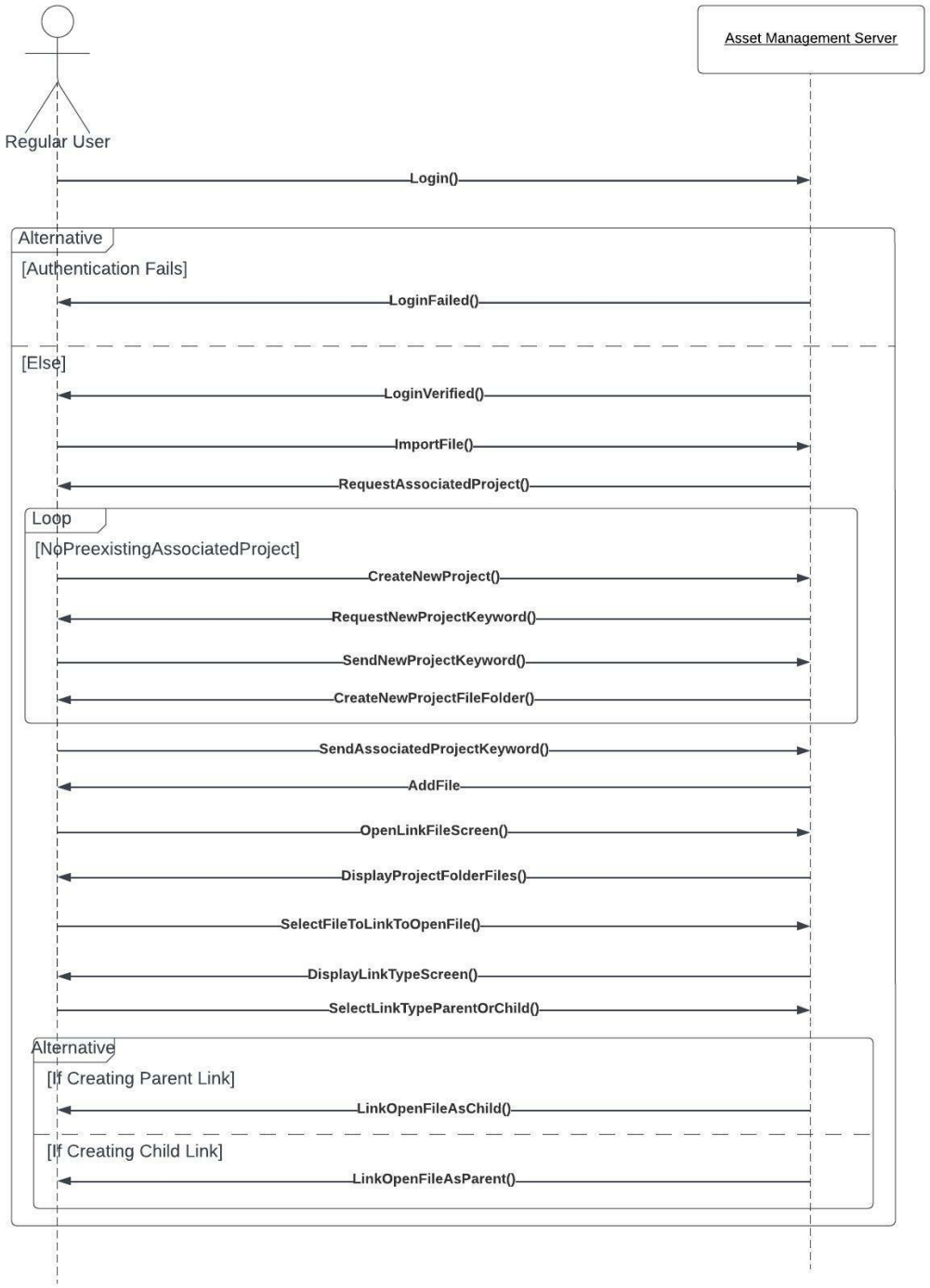


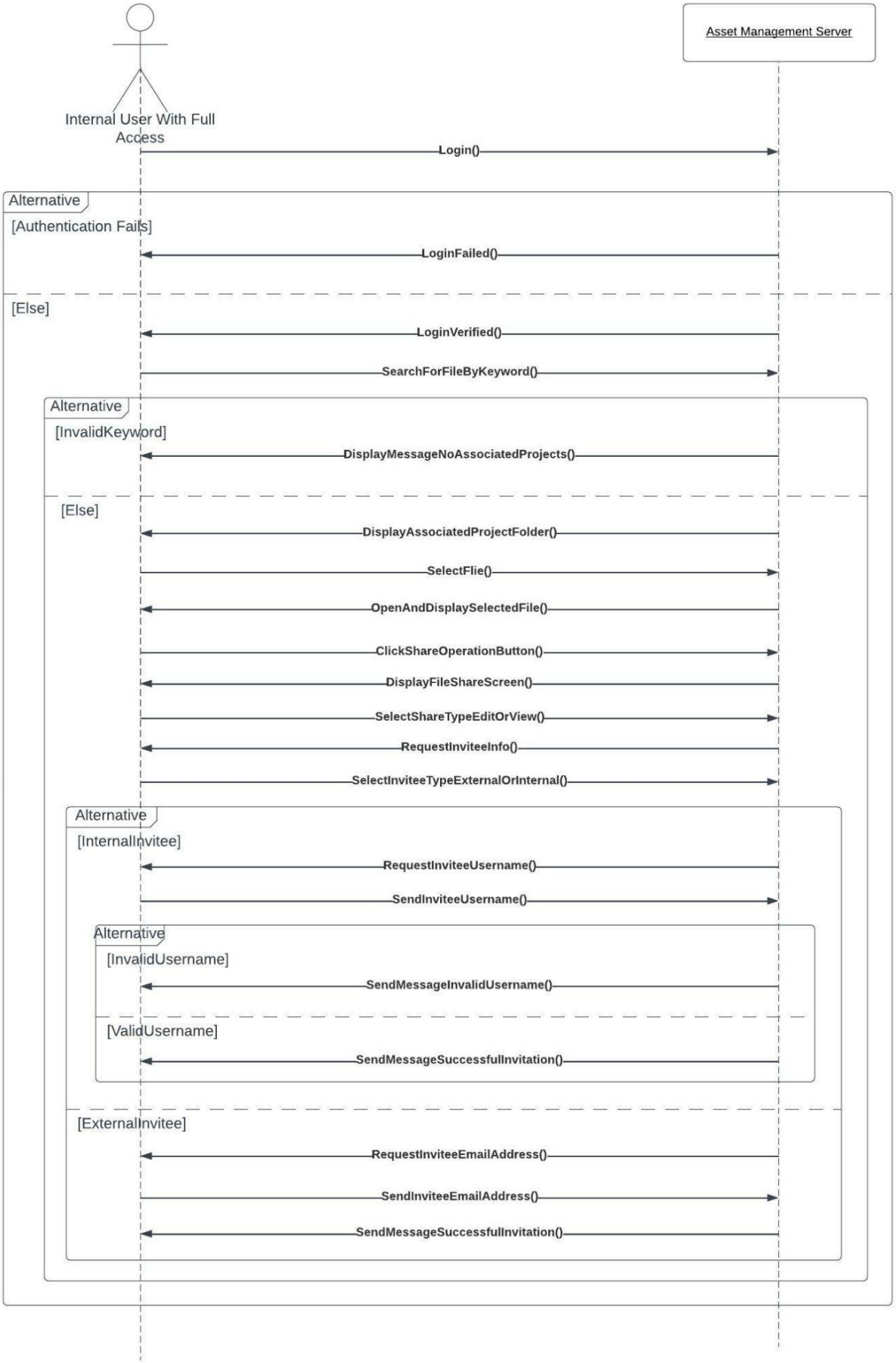

Asset Management Software
Log in
<<include>>
Verify password
<<extend>>
Display login error
Only View Invited User
Asset Management Server
Select file the user was invited to
<<include>>
User views file

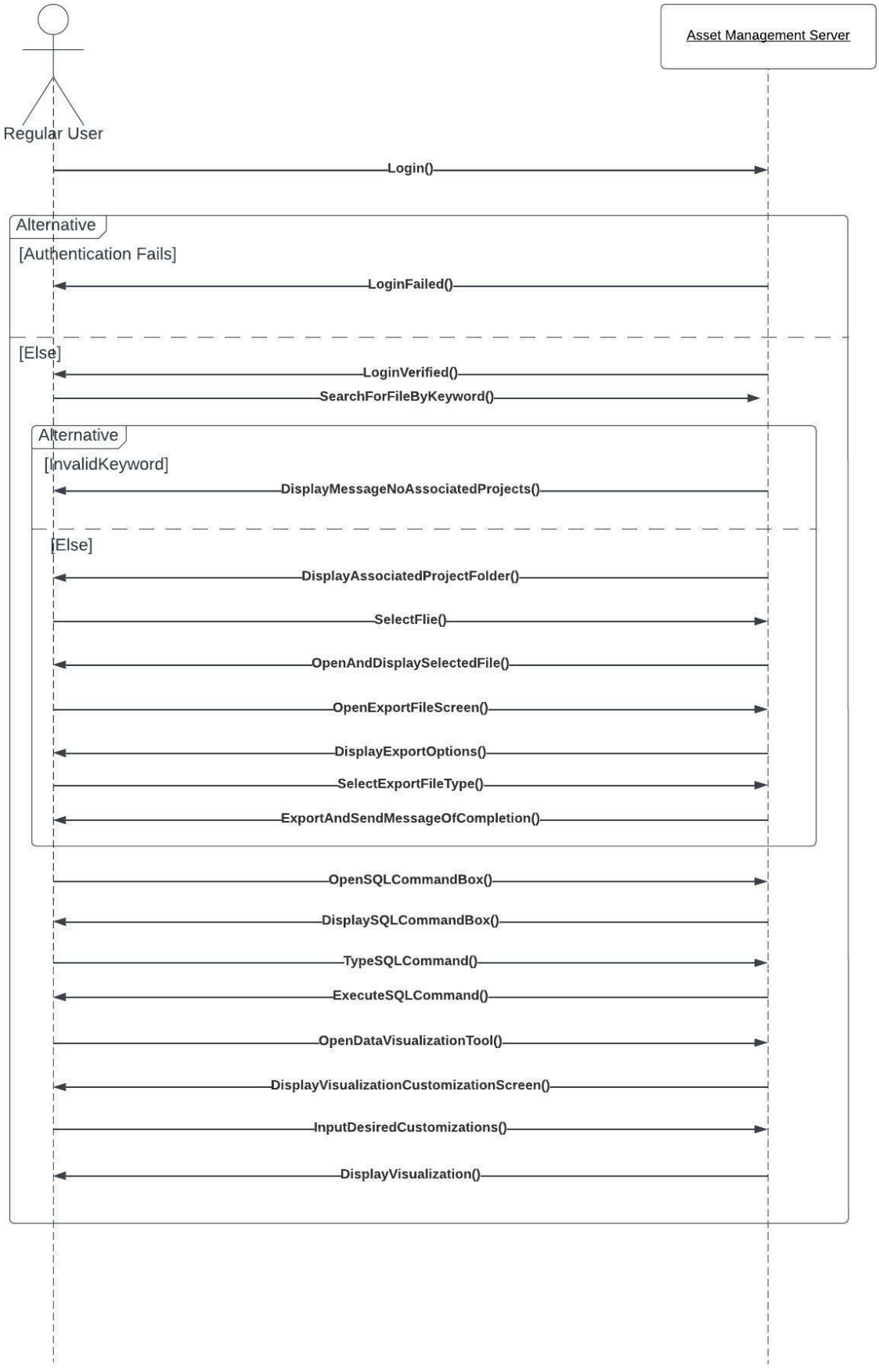

Asset Management Software
Log in
<<include>>
Verify password
<<extend>>
Display login error
Only Edit Invited User
Asset Management Server
Select file the user was invited to
<<include>>
User views file
<<extend>>
User edits file

7. [15 POINTS] **Sequence diagram** – Provide sequence diagrams (similar to Figure 5.6 and Figure 5.7) for each use case of your project. Please note that there should be an individual sequence diagram for each use case of your project. (Ch 5 and Ch 7)

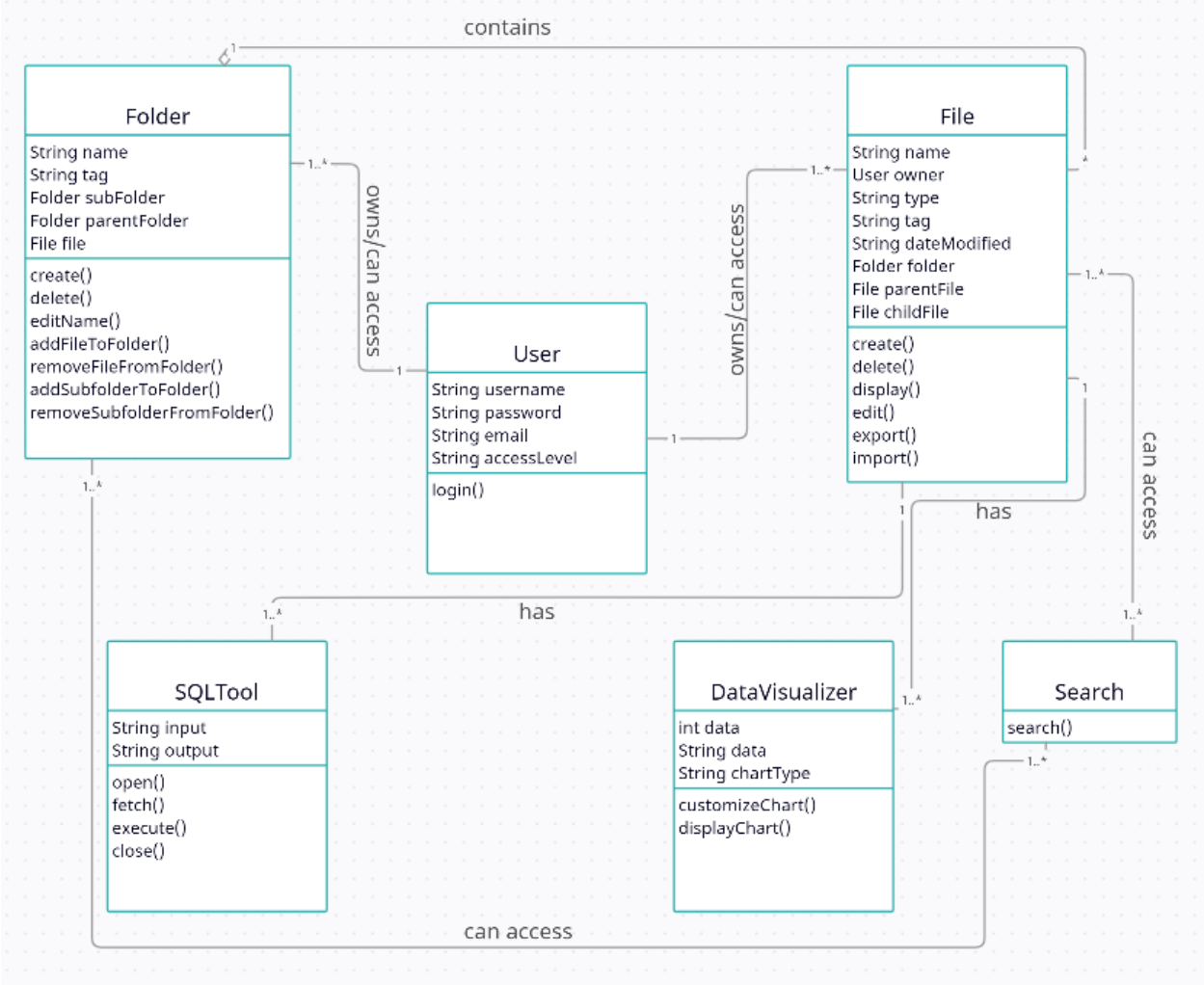
Three sequence diagrams have been provided below. The first includes the actions for importing and linking files. The second shows the actions for inviting collaborators on files and how the search function would work. The third shows how to export files, use SQL commands, and use the data visualization tool. Each "project" would be a different asset in the system, each of which has its own designated keyword or tag.



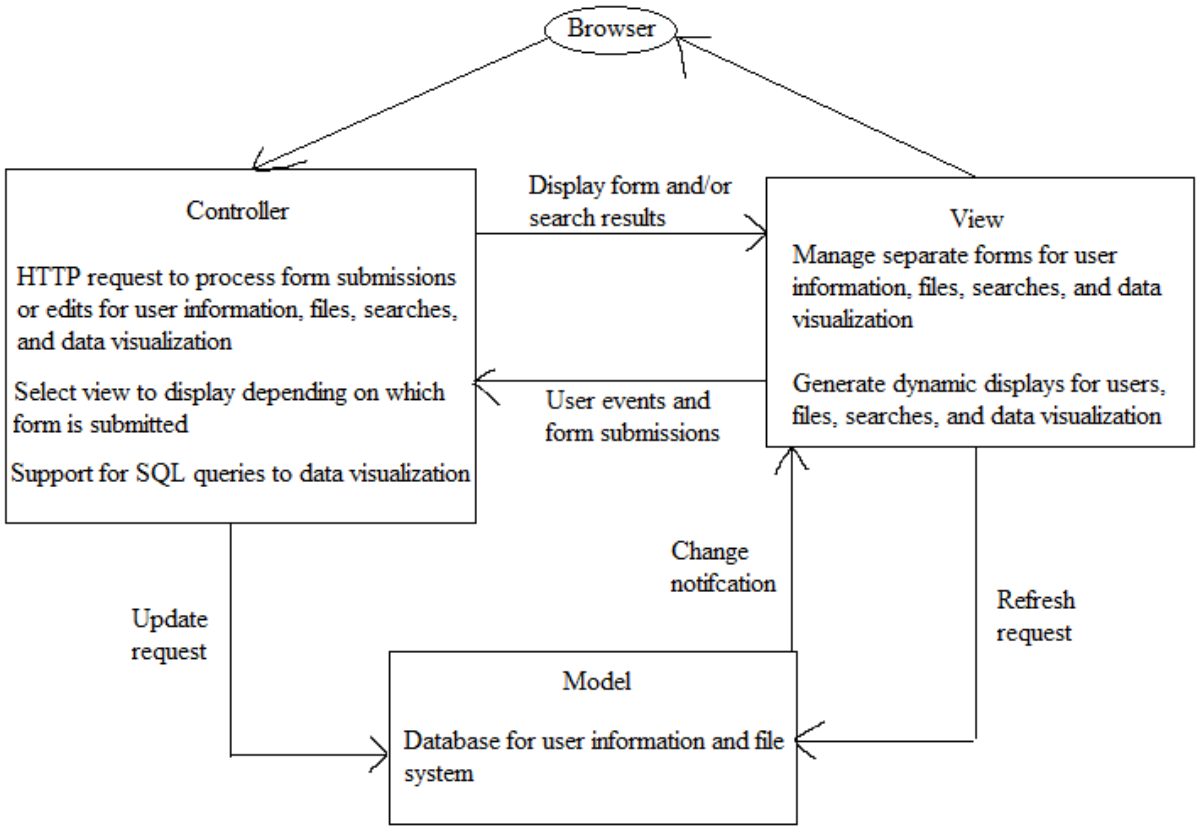

Asset Management Server
Internal User With Full Access
Login()
Alternative
[Authentication Fails]
LoginFailed()
[Else]
LoginVerified()
SearchForFileByKeyword()
Alternative
[InvalidKeyword]
DisplayMessageNoAssociatedProjects()
[Else]
DisplayAssociatedProjectFolder()
SelectFlie()
OpenAndDisplaySelectedFile()
ClickShareOperationButton()
DisplayFileShareScreen()
SelectShareTypeEditOrView()
RequestInviteeInfo()
SelectInviteeTypeExternalOrInternal()
Alternative
[InternalInvitee]
RequestInviteeUsername()
SendInviteeUsername()
Alternative
[InvalidUsername]
SendMessageInvalidUsername()
[ValidUsername]
SendMessageSuccessfulInvitation()
[ExternalInvitee]
RequestInviteeEmailAddress()
SendInviteeEmailAddress()
SendMessageSuccessfulInvitation()


Asset Management Server
Regular User
Login()
Alternative
[Authentication Fails]
LoginFailed()
[Else]
LoginVerified()
SearchForFileByKeyword()
Alternative
[InvalidKeyword]
DisplayMessageNoAssociatedProjects()
[Else]
DisplayAssociatedProjectFolder()
SelectFlie()
OpenAndDisplaySelectedFile()
OpenExportFileScreen()
DisplayExportOptions()
SelectExportFileType()
ExportAndSendMessageOfCompletion()
OpenSQLCommandBox()
DisplaySQLCommandBox()
TypeSQLCommand()
ExecuteSQLCommand()
OpenDataVisualizationTool()
DisplayVisualizationCustomizationScreen()
InputDesiredCustomizations()
DisplayVisualization()

8. [15 POINTS] **Class diagram** – Provide a class diagram (similar to Figure 5.9) of your project. The class diagram should be unique (only one) and should include all classes of your project. Please make sure to include cardinalities, and relationship types (such as generalization and aggregation) between classes in your class diagram. Also make sure that each class has class name, attributes, and methods named (Ch 5).



9. [15 POINTS] **Architectural design** – Provide an architectural design of your project. Based on the characteristics of your project, choose and apply only one appropriate architectural pattern from the following list: (Ch 6 section 6.3)
9.1. Model-View-Controller (MVC) pattern (similar to Figure 6.6)



END PROJECT DELIVERABLE 1 CONTENT

IMPORTANT NOTE: The following items will all need to be calculated / worked on based on the project you are designing. As an example, if a team of 7 students in CS3354 class is working on the development of a hospital information system, this group will prepare the project scheduling, cost, effort and pricing estimation calculations based on the hospital information system design, NOT based on their 7 student team. Think of the analogy to the "Inception" movie: What you will be working on is the dream in a dream, i.e. the dream in the second level, NOT in the first level.

3. [35 POINTS] Project Scheduling, Cost, Effort and Pricing Estimation, Project duration and staffing: Include a detailed study of project scheduling, cost and pricing estimation for your project. Please include the following for scheduling and estimation studies:

3.1. [5 POINTS] **Project Scheduling**. Make an estimation on the schedule of your project. Please provide start date, end date by giving justifications about your estimation. Also provide the details for: - Whether weekends will be counted in your schedule or not - What is the number of working hours per day for the project

Our group plans to start the project after the presentation in December (December 9th). The engineers will only work weekdays. After using the Function Point algorithm, we concluded that this project should take about one week assuming the team works on it for eight hours per day. Adding seven business days to December 9th results in the finish date being December 19th.

3.2. [15 POINTS] **Cost, Effort and Pricing Estimation**. Describe in detail which method you use to calculate the estimated cost and in turn the price for your project. Please choose one of the two alternative cost modeling techniques and apply that only: **- Function Point (FP)** - Application composition

Our group used the Function Point algorithm. We had 4 user inputs, 4 user outputs, 6 user queries, 1 data file/relational table, and 1 external interface. We said our user inputs and outputs would be simple, and everything else would be average, leaving us with a GFP of 68. We then determined that our PCA was 1.12. Based on these two numbers our FP was 76 giving us 2 person-weeks. This process should take one week in total with two software engineers.

3.3. [5 POINTS] Estimated cost of hardware products (such as servers, etc.)

We will need to buy at least one server so that we could host our website. Hosting can range from 3 to 50 dollars per month [1]. Assuming we are within that price range, it would cost us around 25 dollars per month to host the server.

3.4. [5 POINTS] Estimated cost of software products (such as licensed software, etc.)

We would not need any assistance from software products, most of the work would be done through personal IDEs and GitHub.

3.5. [5 POINTS] Estimated cost of personnel (number of people to code the end product, training cost after installation)

According to ZipRecruiter, the average weekly cost of a software engineer is $1,752 [2]. If it takes one week to finish our project and we hire two different software engineers, we would be paying out a total of $3,504 for personnel.

4. [10 POINTS] A **test plan** for your software: Describe the test plan for testing minimum one unit of your software. As an evidence, write a code for one unit (a method for example) of your software in a programming language of your choice, then use an automated testing tool (such as JUnit for a Java unit) to test your unit and present results. Clearly define what test case(s) are provided for testing purposes and what results are obtained (Ch 8). Include your test code as additional document in your zip file submitted.

The unit that will be demonstrated in the test plan is the login verification method. This method will accept a user's username and password as input, check if the user is present in a database of verified users, and then output a boolean that indicates whether the user is part of the system or not. Since the rest of the software has not been implemented yet, the database of users will simply be a locally saved text file. In the completed version of the software, the password will be stored after passing through a hashing function. However, it will require some additional time to determine the best and most secure hashing function, so this will not be included in the unit test. The two test cases included in the code are for a true user and a false user. The unit tests demonstrate that the login method works successfully.

5. [10 POINTS] **Comparison of your work with similar designs**. This step requires a thorough search in the field of your project domain. Please cite any references you make.

- ManageEngine AssetExplorer - ManageEngine has a desktop and mobile application [3]. On the other hand, our software will only have a desktop application. Our software and the ManageEngine can import files and have data visualization tools. The ManageEngine software has a paid subscription model [3]. Our asset management software does not have a subscription model.
- Invgate - Invgate includes some features that our software does not have such as: software license management, financials, network discovery, software metering, and software deployment [4]. Features that our software includes that Invgate does not have is a hierarchical file storage, the ability to grant users different levels of access, and performing SQL queries on the asset data. Both our software and Invgate include detailed data visualization options including a variety of graphs and charts to display data.
- Asset Panda - Our software and Asset Panda both include features of inviting multiple users to view asset data, and the ability to control levels of access among users [5]. Asset Panda is different from our software since it includes a mobile app and a barcode scanner. Our software includes a customizable data visualization feature that Asset Panda does not.

6. [10 POINTS] **Conclusion** - Please make an evaluation of your work, describe any changes that you needed to make (if any), if things have deviated from what you had originally planned for and try to give justification for such changes.

A major strength of our submission for this deliverable includes the function point cost estimation procedures. Our function point evaluation is thorough because we considered the function category count and determined key characteristics of the project planning, such as the complexity, gross function point, and processing complexity adjustment. Changes that we have conducted from plans in deliverable 1 are demonstrated in the complexity analysis. We originally assumed that all function categories were simple but have since changed the user queries, number of data files, and external interfaces to more average and complex levels. We justify these complexity changes because our non-functional space requirement obliges the product to store a minimum of 200,000 documents and the magnitude of this storage increases the complexity of user queries and data files.

A potential weakness in our project concerns the original project planning from deliverable 1. We have addressed this by modifying the project schedule. Our original plan was to begin development this week on November 14, 2022 after the original due date of deliverable 2 plans and continue for approximately 1 week. However, due to unavailability of developers and the extended deadline of deliverable 2 planning we have changed this start date. The new date will commence after the presentation in early December and continue for one week into mid-December.

In conclusion, we are discussing the cost estimation, project schedules, test plans, and competitor comparisons in the submission of this deliverable. We look forward to hearing your feedback.

7. [5 POINTS] **References**: Please include properly cited references in IEEE paper referencing format. Please review the IEEE referencing format document at the URL: https://ieeedataport.org/sites/default/files/analysis/27/IEEE%20Citation%20Guidelines.pdf). It means that your references should be numbered, and these numbers properly cited in your project report.

[1] H. Whitfield, “How Much Does It Cost to Host a Website? | Compare 2022 Prices.” WebsiteBuilderExpert. https://www.websitebuilderexpert.com/web-hosting/cost-to-host-a-website/ [Accessed 11/9/2022]
[2] “How Much Do Computer Engineer Jobs Pay per Week?” ZipRecruiter. https://www.ziprecruiter.com/Salaries/Computer-Engineer-Salary-per-Week [Accessed 11/9/2022].
[3] L. Creamer, “ManageEngine Assetexplorer Review,” PCMAG, 2019. [Online]. Available: https://www.pcmag.com/reviews/manageengine-assetexplorer. [Accessed: 18-Nov-2022].
[4] “It assets management software: Invgate assets,” IT Management Software. [Online]. Available: https://invgate.com/assets/. [Accessed: 09-Nov-2022].
[5] “Asset management software & asset tracking,” Asset Panda, 04-Aug-2022. [Online]. Available: https://www.assetpanda.com/. [Accessed: 09-Nov-2022].