CS3354 Software Engineering

Final Project Deliverable 2

Group 2

Document Management System

Ethan Arnold

Saksham Sangraula

Donovan Johnson

Arjun Balasubramanian

Steven Nguyen

Thinh Nguyen

Will Isaac

1. Delegation of Tasks

Ethan

- 2.1.1 Final Project draft description
- 2.1.2 An addressing to the feedback that we will have provided for the Final Project proposal.
- 2.1.3 List what you are doing / planning to do regarding the feedback provided for your project proposal.
- 2.2.5 Make another commit including a pdf/txt/doc file named "project_scope".
- 2.5.1 Functional requirements.
- 4.1 Project Scheduling
- 5. Test Plan for Software

Saksham

- 2.2.2 Create a GitHub repository named 3354-Group2.
- 2.2.3 Add all team members, and the TA as collaborators.
- 2.2.6 Include the URL of your team project repository into your project deliverable 1 report.
- 2.6 Use case diagram
- 2.7 Sequence diagram
- 4.2 Cost, Effort, and Pricing Estimation
- 4.3 Estimated Cost of Hardware Products
- 4.4 Estimated Cost of Software Products
- 4.5 Estimated Cost of Personnel
- 11. GitHub Requirement

Donovan

- 2.2.4 Make the first commit to the repository (i.e., a README file with [Group2] as its content).
- 2.5.2 Non-functional requirements
- 8. References

Thinh

- 2.4 Which software process model is employed in your project and why.
- 7. Conclusion

Arjun

- 2.8 Class diagram – Provide a class diagram of your project.
- 2.9 Architectural design – Provide an architectural design of your project.
- 9. Presentation Slides

Steven

- 2.8 Class diagram – Provide a class diagram of your project.
- 2.9 Architectural design – Provide an architectural design of your project.
- 6. Comparison of Work

Will

- 2.8 Class diagram – Provide a class diagram of your project.
- 2.9 Architectural design – Provide an architectural design of your project.

2. Project Deliverable 1 Content

2.1 Final Project Draft Description

Objective:

For this project, we are going to set up a VPS on Amazon Lightsail, install a LAMP stack, and OpenDocMan on the system. We will be implementing a docker and a version control system to make sure that each step of the project is well documented and repeatable in case there are steps that need to be re-visited. We will also use Trello and github to manage the progress of the project.

Motivation and real-life application:

We are building this ODMS from scratch because the company's (Naturae Oils) previous tool is not secure, well-documented, or built with a Software Engineering methodology. We are also going to be focusing on the overall security of the tool (ie. making sure that permissions are not unnecessarily given to files or directories).

This is a real-world project that is going to be implemented and used by Naturae to manage their documents securely while adhering to different manufacturing compliances. This project requires good team collaboration and will help us understand software process models better while allowing us to work in a professional environment, using different technologies including virtual private servers and document management systems.

Feedback:

"Great project topic with a fringe benefit, as it will help with a real life system eventually. In the final report, please make sure to include comparison with similar applications -if any- and make sure that you differentiate your design from those and explicitly specify how. Please share this feedback with your group members. You are good to go. Have fun with the project and hope everyone enjoys the collaboration."

Addressing Feedback:

There are many cloud-based document management systems currently in use at the moment such as Alfresco and Bit.ai. The biggest difference between tools like these and the Document Management System we're developing is the ease and freedom of customizability that creating it from scratch allows us to have. With tools that are already pre-built, we couldn't do things such as customizing the UI to cater to our needs or have full control over what packages are available to use – usually services like these require extra funds which isn't a feasible option. While there are benefits to using these paid systems, such as access to tools that help visualize the data or manipulate the files uploaded to the server, there are also many downsides – primarily the issue of requiring an increased budget. By creating this project purely from open source software, budget is not nearly as much of an issue and there are still a variety of different tools to choose from to perform specific tasks.

2.2 Github Repository

https://github.com/sakshamsangraulautd/3354-Group2

2.3 Delegation of Tasks

Saksham

- Create a GitHub repository named 3354-Group2.

- Add all team members, and the TA as collaborators. Include the URL of your team project repository into your project deliverable 1 report.

- Use case diagram – Provide a use case diagram for your project. Please note that there should be more than one use cases depending on the complexity of your project.

- Sequence diagram – Provide sequence diagrams for each use case of your project. Please note that there should be an individual sequence diagram for each use case of your project.

Ethan

- Make another commit including a pdf/txt/doc file named “project_scope”.

- An addressing to the feedback that we will have provided for the Final Project proposal. List what you are doing / planning to do regarding the feedback provided for your project proposal.

- Functional requirements.

Donovan

- Make the first commit to the repository (i.e., a README file with [Group2] as its content).

- Non-functional requirements

Thinh

- Which software process model is employed in your project and why.

Arjun, Will, Steven

- Class diagram – Provide a class diagram 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.

- Architectural design – Provide an architectural design of your project. Based on the characteristics of your project, choose and apply only one appropriate architectural pattern.

2.4 Which software process model is employed in the project and why

The Agile Software Process model is employed in the project. The Agile model is a plan-driven development that has a product at each stage of the development which makes it easier to accommodate changes to the requirements.

The software is developed in increments with the project's manager Ethan instead of the customer specifying the requirements and this suits our methods for developing the project. Agile is also suitable for small projects which makes it the best model for our project due to its simplicity and documentations are not needed, but efficient development of the software is focused. We also have small daily meetings to go over what has been done and the project manager Ethan will discuss what needs to be done till the next meeting.

Due to the characteristics of the Agile software process model that focuses on a quick and efficient development process, we choose to employ this software process model and use its methods in the development of the project.

2.5 Software Requirements

a. Functional Requirements

The system must allow users to log into their account on desktop, mobile, and tablet by entering their username and password.

The system must allow users to reset their passwords by clicking on the “Reset password” button

The system must allow users to logout of their account by clicking the “Logout” button

The system must allow users to upload documents via the “Add Document” tab

The system must allow users to download uploaded documents by clicking the corresponding links

The system must allow users with administrator privileges to Accept or Reject documents via the “Check-In” tab

The system must allow users with administrator privileges to Add, Remove, and Edit other users’ information/accounts via the “Admin” tab

b. Non-functional Requirements

Usability Requirement – The system should not require training to use and be straightforward to interact with

Performance Requirement – The system should operate consistently and be able to recover easily from crashes

Space Requirement – The system must accommodate 40 GB of data a month

Dependability Requirement – The system must ensure that documents are submitted completely and that no issues occur with documents during the submission process; files must be kept intact and backed up in case of crashes

Security Requirement – The system must have a secure process of logging in, and usernames/passwords should be encrypted through MySQL database

Environmental Requirement – The system must be able to process files that are submitted via desktop, laptop, mobile phone, etc., but it will mainly be used with tablets

Operational Requirement – The system will mainly be used by people within actual warehouses inputting documents, whereas engineers will be validating documents

Development Requirement – Should allow room for implementation of additional features (data scraping, compiling data into extractable form)

Regulatory Requirement – The system must be regulated on a monthly basis to ensure that security and organization are well maintained

Ethical Requirement – The ethical use of open-source programming must be complied with, and all related policies and laws must be considered and followed

Accounting Requirement – The system will have a monthly cost of $5

Safety/Security Requirement – The system must ensure that login is secure, and it must be able to store sensitive documents safely

2.6 Use Case Diagram

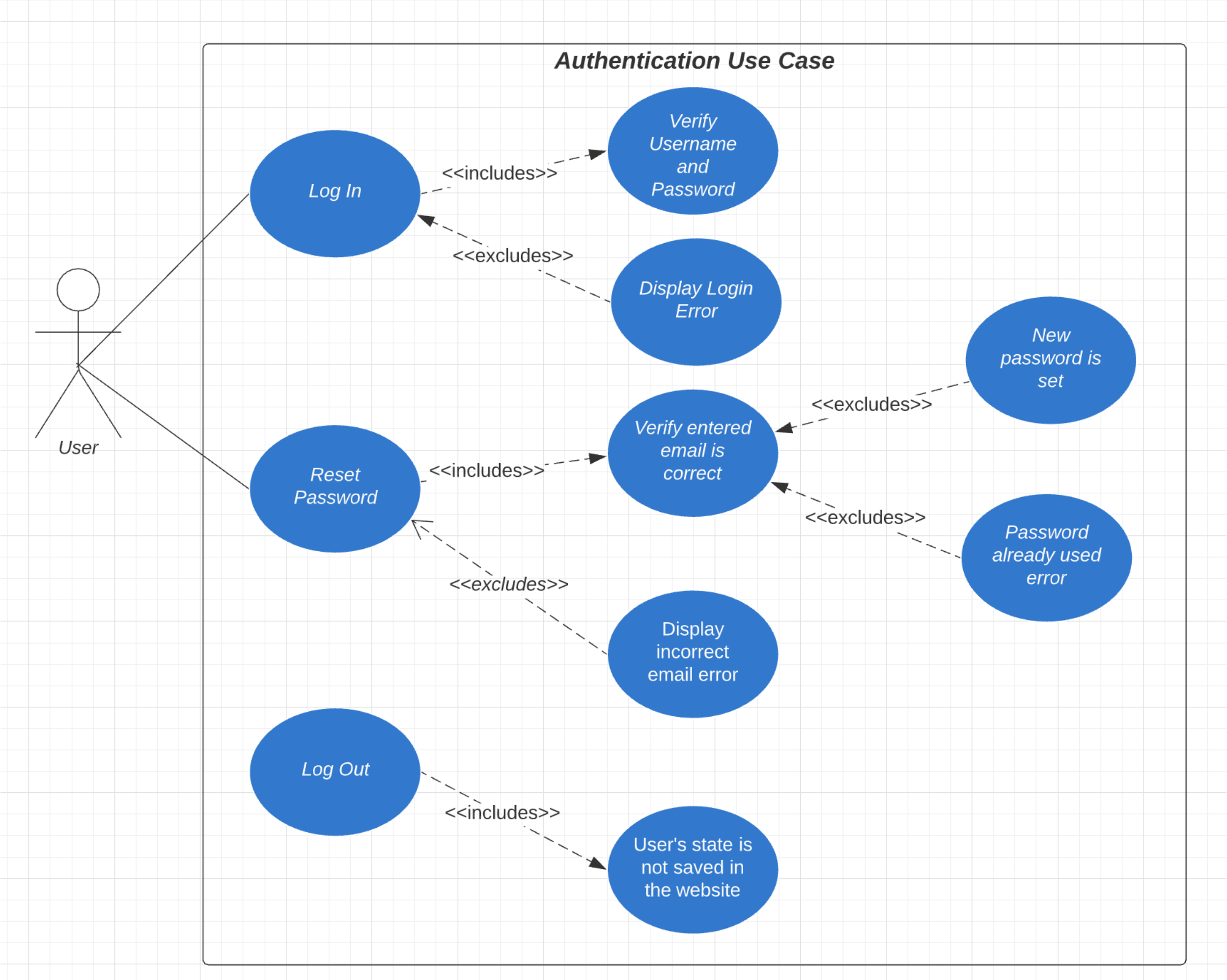
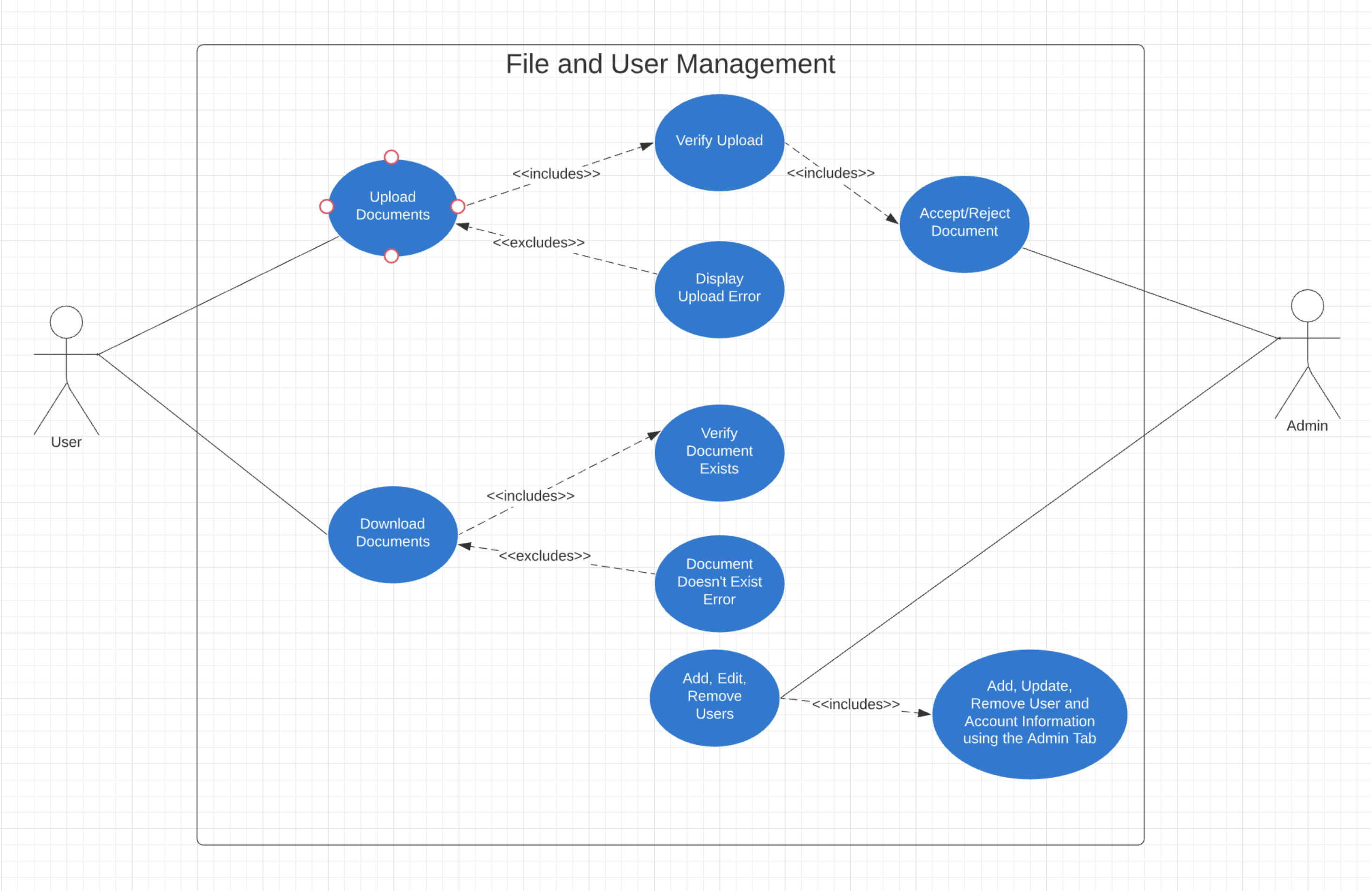


Figure 2.6.1

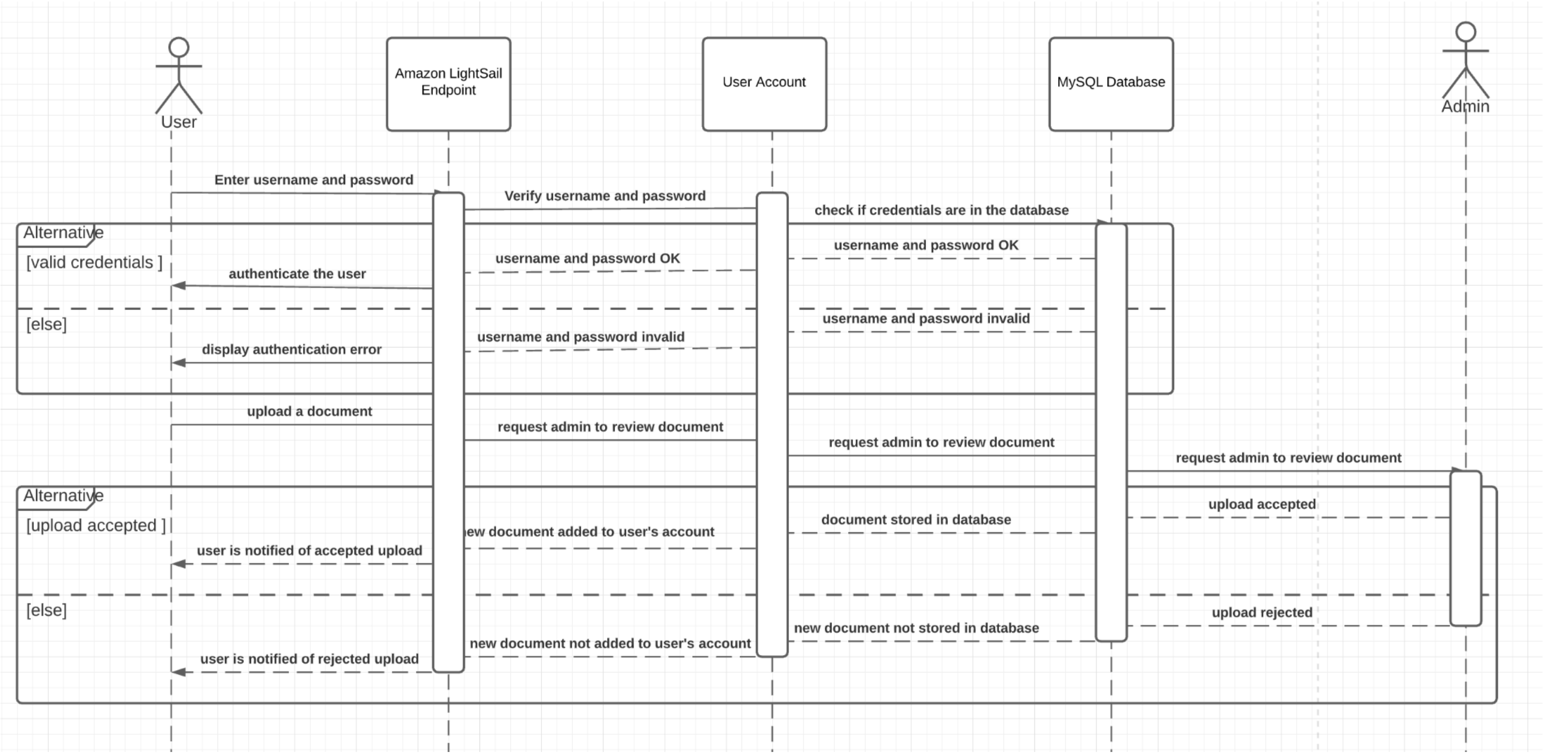
Figure 2.6.2


File and User Management
Upload Documents
Verify Upload
<<includes>>
<<includes>>
Accept/Reject Document
<<excludes>>
Display Upload Error
User
Admin
Verify Document Exists
<<includes>>
Download Documents
<<excludes>>
Document Doesn't Exist Error
Add, Edit, Remove Users
<<includes>>
Add, Update, Remove User and Account Information using the Admin Tab

2.7 Sequence Diagram

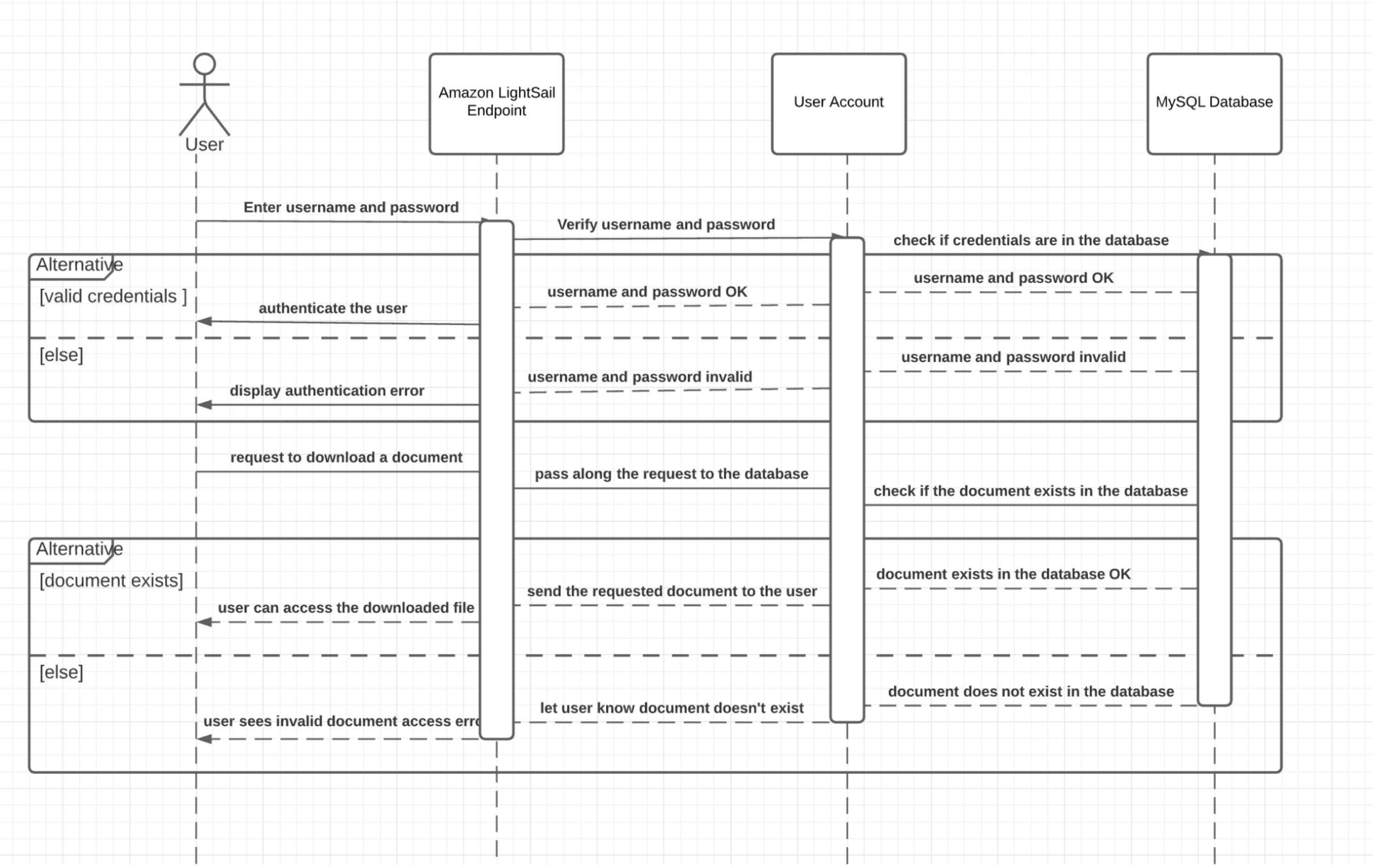
Sequence Diagram showcasing the process to upload documents

Figure 2.7.1


User
Amazon LightSail Endpoint
User Account
MySQL Database
Admin
Enter username and password
Verify username and password
check if credentials are in the database
Alternative
[valid credentials]
username and password OK
username and password OK
authenticate the user
[else]
username and password invalid
username and password invalid
display authentication error
upload a document
request admin to review document
request admin to review document
request admin to review document
Alternative
[upload accepted]
upload accepted
document stored in database
ew document added to user's account
user is notified of accepted upload
[else]
upload rejected
new document not stored in database
new document not added to user's account
user is notified of rejected upload

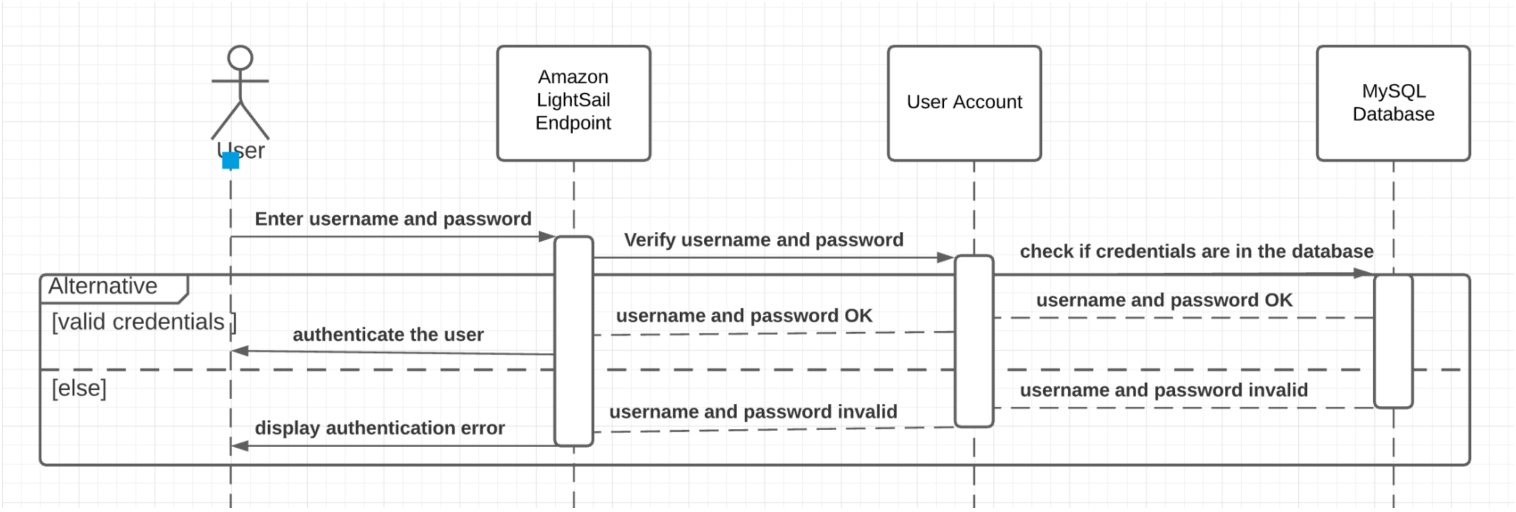
Sequence Diagram showcasing the process to download documents

Figure 2.7.2


User
Amazon LightSail Endpoint
User Account
MySQL Database
Enter username and password
Verify username and password
check if credentials are in the database
Alternative
[valid credentials]
username and password OK
username and password OK
authenticate the user
[else]
username and password invalid
username and password invalid
display authentication error
request to download a document
pass along the request to the database
check if the document exists in the database
Alternative
[document exists]
document exists in the database OK
send the requested document to the user
user can access the downloaded file
[else]
document does not exist in the database
let user know document doesn't exist
user sees invalid document access erro

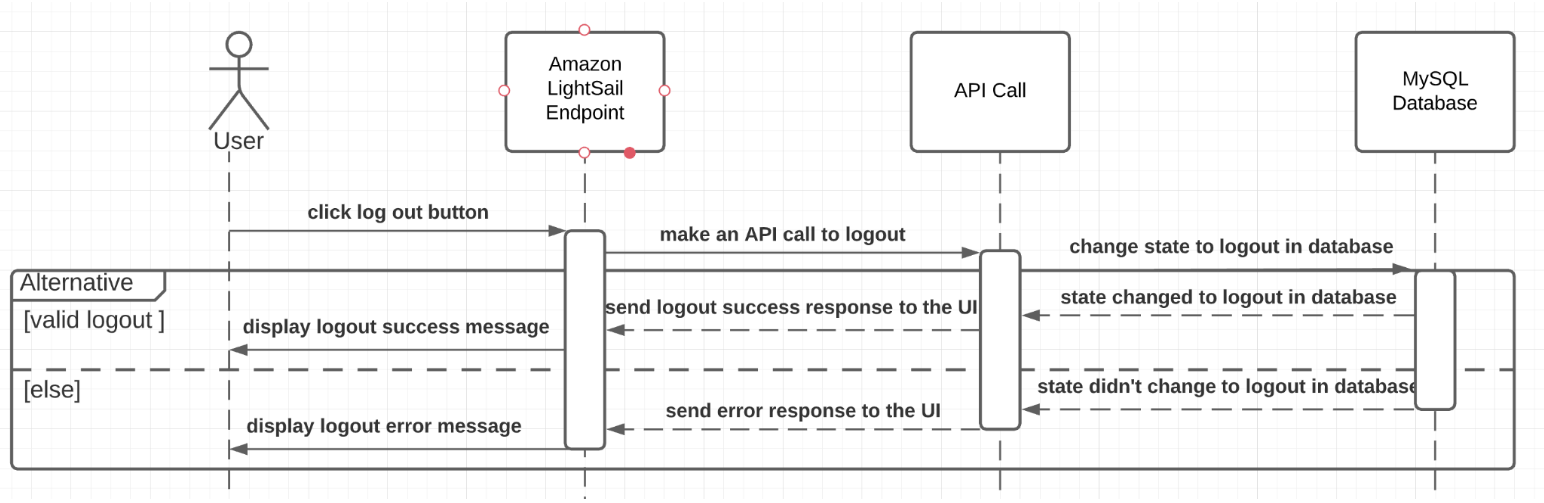
Sequence Diagram showcasing the process to login

Figure 2.7.3


User
Amazon LightSail Endpoint
User Account
MySQL Database
Enter username and password
Verify username and password
check if credentials are in the database
Alternative
[valid credentials]
username and password OK
username and password OK
authenticate the user
[else]
username and password invalid
username and password invalid
display authentication error

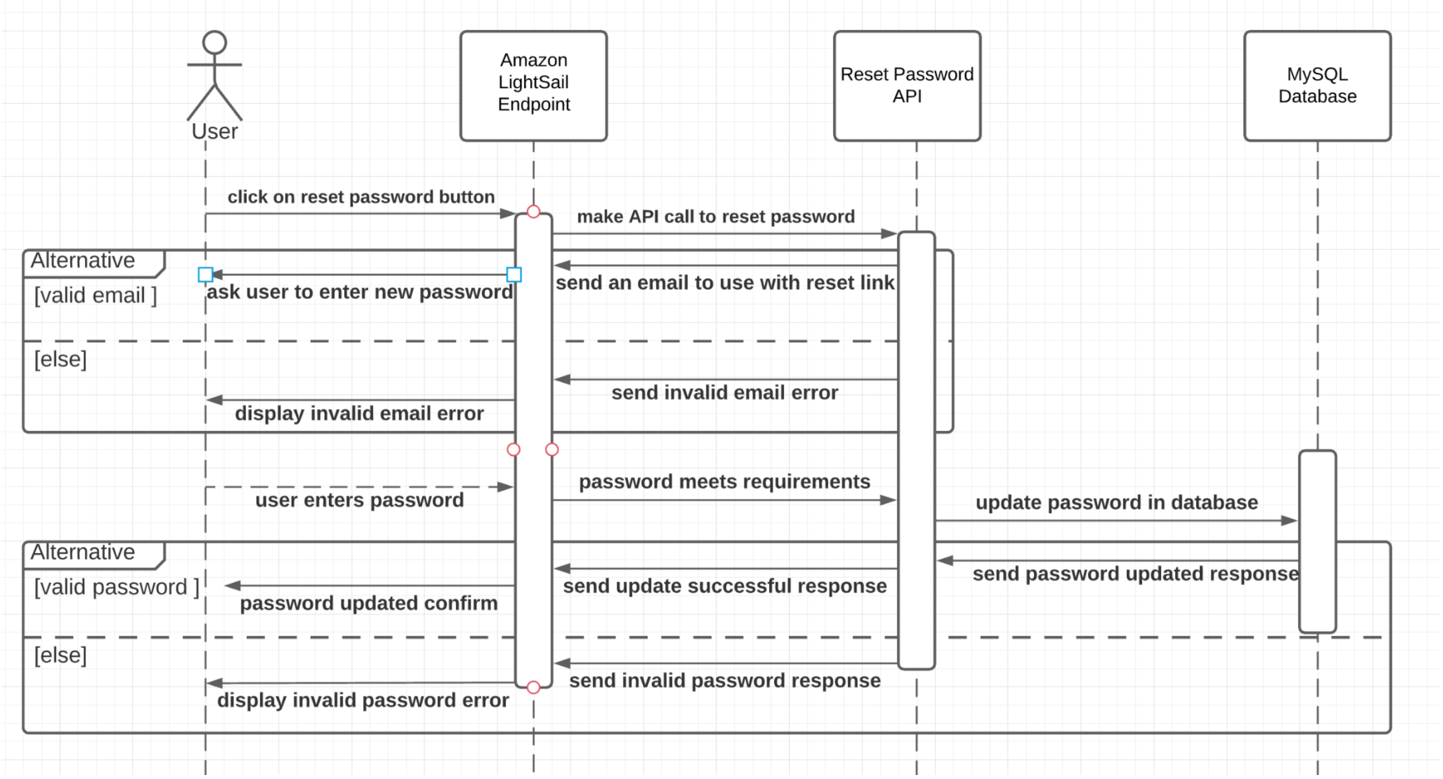
Sequence Diagram showcasing the process to logout

Figure 2.7.4


User
Amazon LightSail Endpoint
API Call
MySQL Database
click log out button
make an API call to logout
change state to logout in database
Alternative
[valid logout]
state changed to logout in database
send logout success response to the UI
display logout success message
[else]
state didn't change to logout in database
send error response to the UI
display logout error message

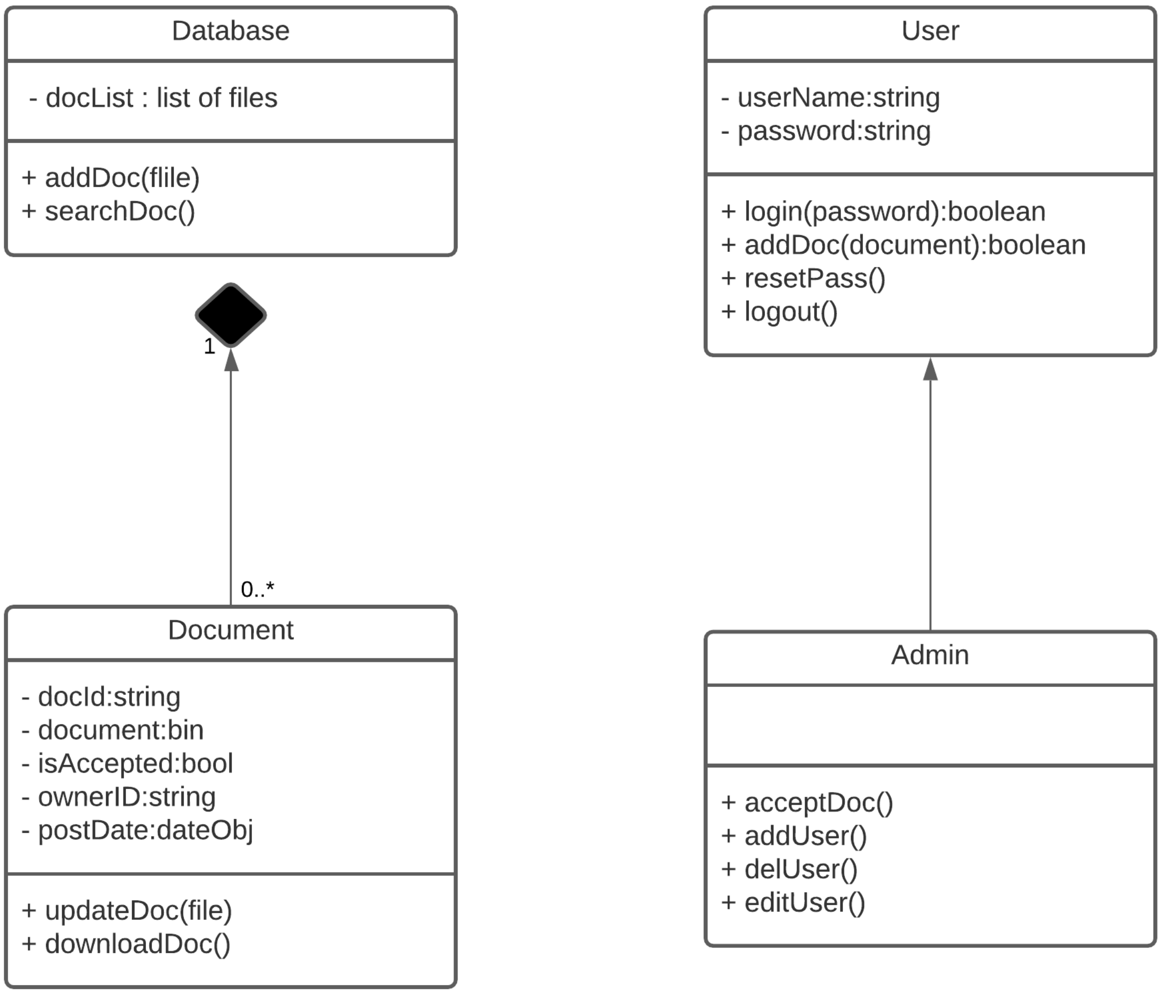
Sequence Diagram showcasing the process to reset password

Figure 2.7.5


User
Amazon LightSail Endpoint
Reset Password API
MySQL Database
click on reset password button
make API call to reset password
Alternative
[valid email]
ask user to enter new password
send an email to use with reset link
[else]
send invalid email error
display invalid email error
user enters password
password meets requirements
update password in database
Alternative
[valid password]
send password updated response
send update successful response
password updated confirm
[else]
send invalid password response
display invalid password error

2.8 Class Diagram

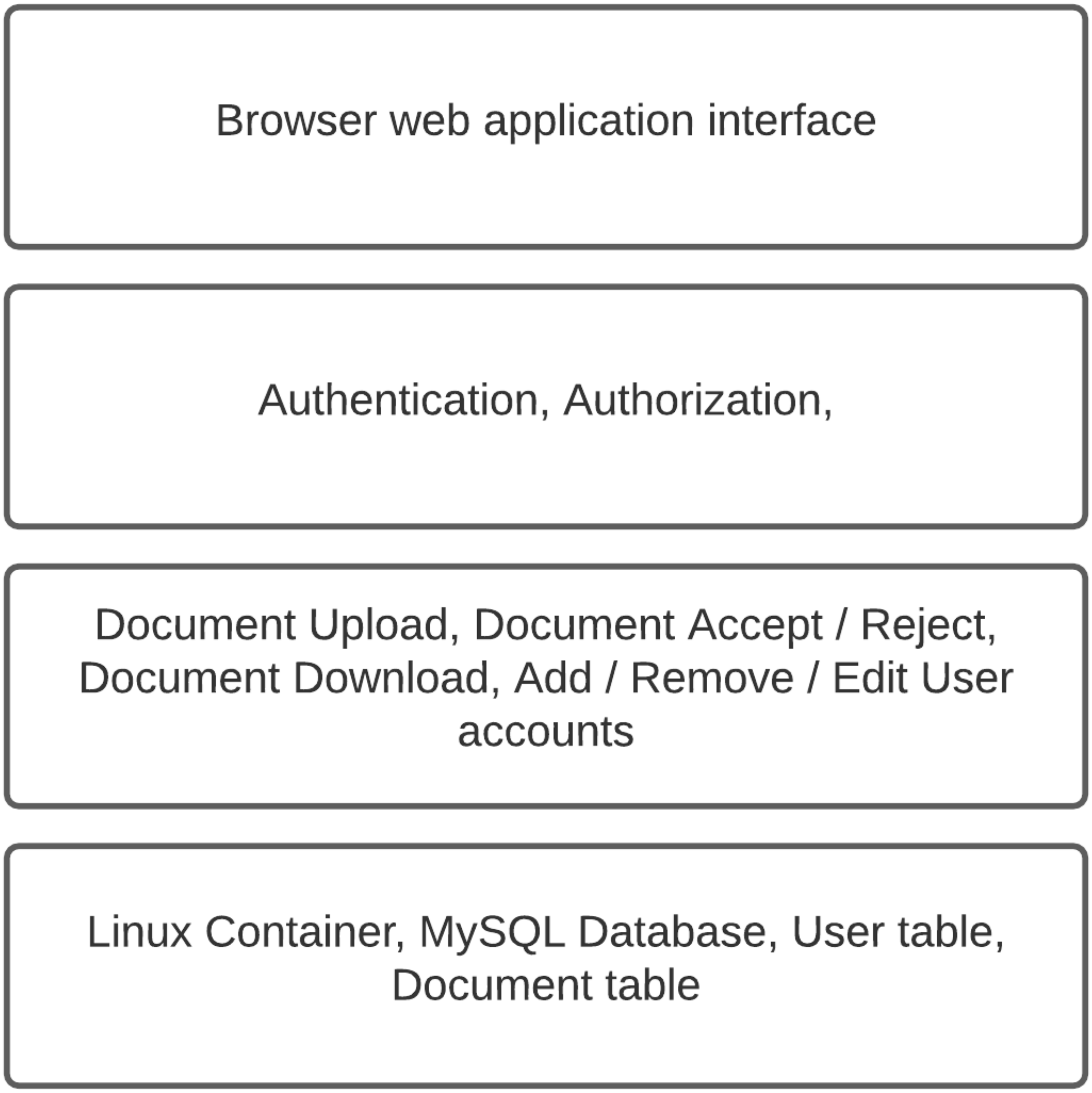
Figure 2.8.1



2.9 Architectural Design

Layered Architecture Pattern

Figure 2.9.1


Browser web application interface
Authentication, Authorization,
Document Upload, Document Accept / Reject, Document Download, Add / Remove / Edit User accounts
Linux Container, MySQL Database, User table, Document table

3. **Choose one of the options and specify clearly your choice**

    a. Present face to face

4. Project Scheduling, Cost, Effort, and Pricing Estimation

a. 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:

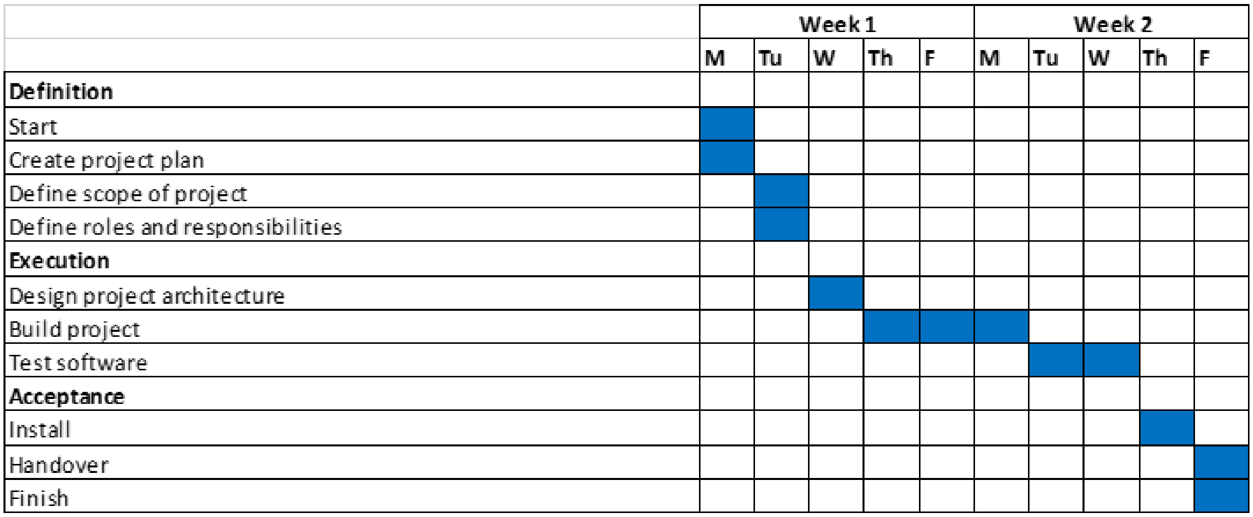
4.1 Project Scheduling

a. 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:

i. Whether weekends will be counted in your schedule or not

ii. What is the number of working hours per day for the project

The project will take approximately 2 weeks to complete with a team of 7 people. Weekends will not be counted in the project's schedule. We are also going to use an estimate of 4 hours/function point to give us a total of 280 function point hours per week (70 function points per week * 4 hours per point). Since there are seven members of the group participating in building this project, we divide the total function point hours per week by members to find how many work hours per week. 280/7 = 40, so with 5 day work weeks, each member will be spending 8 hours working each day for the project.



	Week 1					Week 2				
	M	Tu	W	Th	F	M	Tu	W	Th	F
Definition										
Start	■									
Create project plan	■									
Define scope of project		■								
Define roles and responsibilities		■								
Execution										
Design project architecture			■							
Build project				■	■	■				
Test software							■	■		
Acceptance										
Install									■	
Handover										■
Finish										■

Figure 4.1

4.2 Cost, Effort, and Pricing Estimation

a. 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:
   i. Function Point (FP)
   ii. Application composition

The algorithmic estimation technique that we agreed upon as a team is Functional Point. We agreed that the 14 questions to calculate PCA were relevant to our project and the overall implementation and would help us estimate our efforts better, so all of us in the team decided to use Functional Point as our estimation technique. The formula to calculate Function Point is: FP = GFP * PCA

To calculate the GFP, I need to take in account of 5 function categories which are:

1. Number of user input: Our estimated number was **10** taking in account for inputs from the end users as well as admin users and it will be categorized as a **simple** complexity.
2. Number of user output: Our estimated number was **5** taking in account for downloading of files and messages to end users and admins and it will be categorized as a **simple** complexity.
3. Number of user queries: Our estimated number was **10** taking in account for different queries users can execute from profile update to interacting with files and data and it will be categorized as a **simple** complexity.
4. Number of data files and relational tables: Our estimated number was **50** taking in account for the fairly large number of user files and data that will be stored per user, and it will be categorized as an **average** complexity.
5. Number of external interfaces: Our estimated number was **10** taking in account for some external tools we will be using for testing and deployment it will be categorized as a **simple** complexity.

Based on the estimates above, here is a table to calculate GFP:

	Function Category	Count	Simple Complexity	Average Complexity	Complex Complexity	Count * Complexity
1	Number of user input	10	3	4	6	30
2	Number of user output	5	4	5	7	20
3	Number of user queries	10	3	4	6	30
4	Number of data files and relational tables	50	7	10	15	500
5	Number of external interfaces	10	5	7	10	50
					Gross Function Unit (GFU)	630

Figure 4.2

Note: Complexity is highlighted in yellow above.

Hence, GFU is 630. Below is the calculation for PCA:

PCA = 0.65 + 0.01 (PC1 + PC2 +…. PC14) where 0.65 is an empirical constant and PC1…PC14 are the answers for the one of the 14 questions used to determine processing complexity. Each PC = a number in range from 0 to 5 inclusive

The 14 questions and their scores are outlined below based on what we decided together as a group:

1. **Does the system require reliable backup and recovery?** 5 - The system handles critical files for users so reliable backup and recovery is essential for our project
2. **Are data communications required?** 5 - Data communications are also essential for our project since our system will be responsible for transferring files, and following authentication and authorization processes using the data of the user.
3. **Are there distributed processing functions?** 2 - Multiple individual CPUs are not working on the same programs consistently.
4. **Is performance critical?** 3 - Our system does not have the requirement to process upload and download of files within seconds or faster, so performance is not significantly critical.
5. **Will the system run in an existing, heavily utilized operational environment?** 2 - The system is running on a server which can be remotely accessed and is not heavily utilized.
6. **Does the system require online data entry?** 5 - Users are using the system to authenticate and also interact with existing and new files in the system, so online data entry is essential for our system.
7. **Does the online data entry require the input transaction to be built over multiple screens or operations?** 0 - Multiple screens or operations are not relevant for our system.
8. **Are the master files updated online?** 5 - Records in master files are updated based on changes to user's data
9. **Are the inputs, outputs, files, or inquiries complex?** 4 - Our system needs to be able to handle many types and sizes of file so it is significant for our system to handle complex inputs, outputs, files and queries.
10. **Is the internal processing complex?** 2 - The internal processing of the system is moderately complex, but the complexity is reduced and managed with well-built APIs.
11. **Is the code designed to be reusable?** 2 - The codebase in our system is designed to be reusable within the system, so components within the system can be reused, but the entire codebase may not provide reusability when designing a new system with different functionalities.
12. **Are conversion and installation included in the design?** 2- Conversions are not included in the design and only some installations (linux, opendocman) are included in the design.
13. **Is the system designed for multiple installations in different organizations?** 1 - The system is designed to be accessed by multiple users online, so multiple installations in different organizations is not required as the system is easily accessible.
14. **Is the application designed to facilitate change and ease of use by the user?** 4 - Since this system is interactive and users are expected to heavily interact with the functionalities (like

upload/download files, login), it is significant to take in account for implementing changes and ease of use along with user feedback for our system.

PCA = 0.65 + 0.01 (5+5+2+3+2+5+0+5+4+2+2+2+1+4) = 1.07

FP = GFP * PCA = 630 * 1.02. So, the result of the Function Point algorithmic estimation technique is 674.1

Assuming that productivity of our team is 70 function points per week, the estimated effort is:

674.1/70 = 9.63 person-weeks

We have a team size of 7 so, the project duration is:

9.63/7 = 1.38 weeks

So, the overall project duration is 2 weeks (rounding up the estimate)

4.3 Estimated Cost of Hardware Products

a. Such as servers, etc

To set up a VPS on Amazon Lightsail, we need to create a LAMP instance in the AWS cloud. The server runs on a $10 per month Linux instance with 2 GB Memory, 1 Core Processor, 60 GB SSD Disk and 3 TB Transfer and contents like files are stored in Object Storage of 250 GB and Transfer of 500 GB for $5/month [1]. MySQL database is also implemented in Amazon Lightsail with the Standard plan for $15/month. The overall cost of setting up VPS on Amazon Lightsail is $30/month [1]. All of the functionality is supported with the Amazon Lightsail service and its individual components are easily integrated through this service. So, the total hardware costs associated with our service is **$30/month.**

4.4 Estimated Cost of Software Products

a. Such as licensed software, etc

Since we are mostly using open source softwares, we do not have a high cost for software products. We are using the Free version of Github plan so that would be $0/month. We are also using a free version of Trello project tracking software so that would also be $0/month. OpenDocMan Community edition is also free [2]. We are planning on using Docker to containerize our application and the Personal Plan they provide allows small projects like ours to use the platform for free [3].

So, the overall cost of software products is **$0/month** since we are only using free versions of softwares and most of them are open source as well.

4.5 Estimated Cost of Personnel

a. Number of people to code the end product, training cost after installation

We have a relatively small team of people developing and maintaining the product. We have 4 main developers coding the end product who are paid $5,000/month each. We have 2 QA (Quality Assurance) testers in the team who are responsible for the end to end testing of the product. The salary of each QA tester is $4000/month. We also have 1 product owner in the team responsible for setting the goals for the product and leading the overall design and architecture effort of the product. The product owner is compensated $8,000/month. We have 2 marketing staff in the Marketing team who are each given a salary of $3000/month. We also have 2 project managers responsible for the daily scrum meetings and helping the developers and the entire team be on track for their individual projects. Each project manager is given a salary of $3000/month. The entire team is led by the President of our startup who is responsible for partnering with other companies in the Industry and has a salary of $10,000/month. Since the documentation can be written by the developers who created the product, we will not have any additional training cost after the installation. The developers will also be responsible for resolving support tickets submitted by the users. The developers will also upload recordings showcasing the functionalities provided by our product in Youtube so users can easily understand the functionality we provide. So, the overall estimated cost of personnel is $58,000/month taking in account $20,000/month (4 * $5000/month) for all developers, $8000/month (2 * $4000/month) for all QA testers, $8000/month for the product owner, $6000/month (2 * $3000/month) for marketing staff, $6000/month (2 * $3000/month) for project managers, and $10,000 for the President.

4.6 Estimated Total Cost of Project

To estimate the total cost of the entire project, we are going to add our results from 4.3, 4.4, and 4.5 together.

Hardware: $30/month
Software: $0/month
Personnel: $58,000/month

Since the development of this project is estimated to take 2 weeks to complete, we can calculate the total costs of development to be ($58,000 + $30) / 2 = $29,015/month

5. Test Plan for your software:

a. Describe the test plan for testing minimum one unit of your software.

b. 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.

c. Clearly define what test case(s) are provided for testing purposes and what results are obtained (Ch 8).

d. Include your test code as additional document in your zip file submitted.

This project utilizes many different pieces of software that work together to create an online database management tool. Since we are working with a wide array of applications, it is useful to test some smaller components from each software to guarantee usability and minimize bugs.

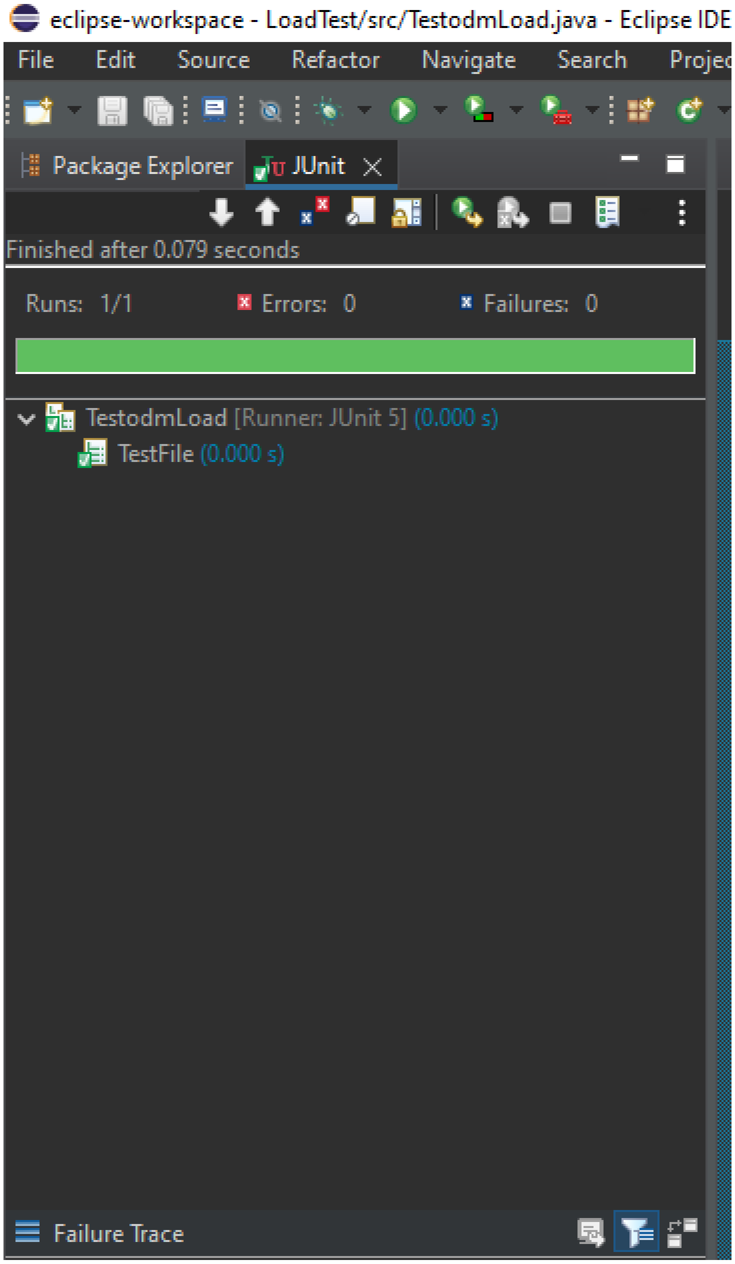
To test our project, we are going to use JUnit testing method, where we will analyze the class odmLoad. This class is a part of the OpenDocMan software and is used to set the absolute path of the project's directory by searching some of the more common directories for the configuration file. Once the configuration file is found, the variable ABSPATH is set. If the configuration file is not found, an exception error will be thrown and the user will be prompted to manually set the configuration file. Since the ABSPATH variable is a vital part of the system, this test will function as a Regression Test to ensure that no unintended behaviors or errors are introduced with any new changes to the software.

In the unit tests developed for this class, each of the directories are to be tested via JUnit. These directories include the following:

"config.php" – root folder
"docker-configs/config.php" – docker folder
"../config.php" – subfolder
"../../config.php" – plugin folder

If the "config.php" file exists in one of these directories, the JUnit test will assertTrue to determine whether or not the file does exist. If the test fails, there is something wrong with the programming logic and the class will need to be fixed since it's not recognized. If the configuration file does not exist in any of the directories, the program will default to a catch statement. To further test the reliability of this, assertions have been included to verify that the configuration file does not exist – if the Unit Test fails here, then an existing configuration file has slipped through the previous if statements.

In figure 5.1, we can see that the Test class has passed the included JUnit tests. This means that the program either correctly recognizes that the configuration file, “config.php” in the root folder, docker folder, subfolder, or plugin folder or the program correctly recognizes that the configuration file is missing and will throw an exception to let the user know. Since the program is run through consecutive statements, the root folder’s configuration file will have precedence (since it is called first) and that file is going to be picked up, rather than an existing configuration file located in the docker, subfolder, or plugin folder.



Finished after 0.079 seconds

Runs: 1/1 Errors: 0 Failures: 0

TestodmLoad [Runner: JUnit 5] (0.000 s)
TestFile (0.000 s)

Figure 5.1

6. Comparison of your work with similar designs.

a. This step requires a thorough search in the field of your project domain. Please cite any references you make.

Online document management systems (ODMS) are a very common and widely used software. There are many options available, from paid proprietary software as a service options to free and open source. Some paid or proprietary options include monday.com and google drive, while some open source options include OpenDocMan, what we chose, and Mayan EDMS.

Since the source code and architecture of the proprietary options are not available, a feature comparison will be made. Alongside document management, monday.com [4] provides more features than OpenDocMan such as project management, integrations, automations, and many pre-made templates to customize to a business's needs. It includes an easy to navigate UI that works properly on mobile and desktop. A feature that is missing, however, is the ability to review documents submitted by users to approve or deny.

Another proprietary option is Google Drive. Google Drive [5] provides an easy way for people to upload and download documents. There are native clients for Google Drive, which the other applications do not provide. Where it falls short is managing these documents since there isn't a way to set permissions and have a review system where an admin can review the documents that a user uploads without doing it manually.

An open source alternative is Mayan EDMS [6]. The product provides unique features that OpenDocMan does not, such as providing Application Program Interface (API) support to integrate with 3rd party software [7]. It provides the same functionality when it comes to permissions, where an access control list can be customized to grant permissions to a group of people. Mayan EDMS provides it's own Docker container for easy deployment [8], which deploys the entire software stack and is premade by the developers, which is similar to what we created to deploy OpenDocMan. It also provides duplicate document scanning as well as automatic OCR processing [9].

Between all the options, Mayan EDMS and OpenDocMan provide features that are most aligned with the requirements of the business. It might be worth exploring Mayan EDMS further, however the developers and users are more accustomed to OpenDocMan, which fulfills all the necessary requirements.

7. Conclusion

a. 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.

Overall, the project progress has all been going according to the planning, which took the team about 2 weeks of business days to complete. The workload was divided among all 7 members of the team and each had a 5 days work-week, 8 hours day to deliver the finished project in 2 week. We estimated the cost of hardware to be $30/month which was within our set budget. Weekly meeting goals were met except for one time some functions got delayed due to the complexity, but it still was within the planned deadline. We were able to use Trello to track our progress over the week and also communicate frequently with the team to resolve any issues

All the functions were tested thoroughly, and the performance was overall satisfactory. The system performed well and handled all the functions smoothly without any errors. The end product was delivered on time with one minor change which had docker implemented to standardize the executable components to be able to run on different environments.

8. References

a. Please include properly cited references in IEEE paper referencing format. Please review the IEEE referencing format document at the URL:

b. https://ieee-dataport.org/sites/default/files/analysis/27/IEEE%20Citation%20Guidelines.pdf

c. It means that your references should be numbered, and these numbers properly cited in your project report.

[1] "Amazon Lightsail Pricing," *Amazon*. [Online]. Available: https://aws.amazon.com/lightsail/pricing/. [Accessed: 09-Nov-2021].

[2] "Document Management System Pricing," *OpenDocMan*, 19-Jan-2019. [Online]. Available: https://www.opendocman.com/document-management-system-pricing/. [Accessed: 09-Nov-2021].

[3] "Docker Pricing & Monthly Plan Details," *Docker*. [Online]. Available: https://www.docker.com/pricing. [Accessed: 09-Nov-2021].

[4] "Work OS: The Visual Platform that Manages Everything," *monday.com*. [Online]. Available: https://monday.com/product/. [Accessed: 09-Nov-2021].

[5] "Google Workspace Learning Center," *Google Help*. [Online]. Available: https://support.google.com/a/users?hl=en#topic=9247638. [Accessed: 09-Nov-2021].

[6] R. Rosario, "Your Documents. No Surprises.," *Mayan EDMS*. [Online]. Available: https://www.mayan-edms.com/. [Accessed: 09-Nov-2021].

[7] "Mayan EDMS," *GitLab*. [Online]. Available: https://gitlab.com/mayan-edms/mayan-edms. [Accessed: 09-Nov-2021].

[8] "OpenSource DMS Features," *OpenDocMan*, 18-Jan-2019. [Online]. Available: https://www.opendocman.com/features/. [Accessed: 09-Nov-2021].

[9] “OpenDocMan - Free PHP Document Management System DMS,” *GitHub*. [Online]. Available: https://github.com/opendocman/opendocman. [Accessed: 09-Nov-2021].

9. Non-recorded (no-voice) presentation slides.

https://docs.google.com/presentation/d/1J45F3d4RlDfPnsWw89smza245AxHSDBMHXKaA6XEuug

10. Your program code (if fully implemented the project, not required otherwise). OPTIONAL PART.

Since this project is composed of multiple different modules (including Apache, MySQL, and OpenDocMan), these modules may not be able to execute on their own. The project is, however, up and running on a remote server that can be accessed through the http address given below. A username and password have been created to allow access to the tool:

http://3.224.10.20/
Username: cs3354
Password: deliverable2

A demo of the software has also been included in the link below as evidence of working code.

https://youtu.be/bZycO2jOS2E

11. GitHub requirement:

a. Make sure at least one member of your group commits everything for project deliverable 2 to your GitHub repository, i.e.

    i. Your final project deliverable2 report

    ii. Unit test code for a sample unit of your project

    iii. Implementation code (if you have implemented your project)

    iv. Non-recorded (no voice) presentation slides

    v. Still, one member of your team should also submit the required project deliverable 2 materials to eLearning

12. Figure References