UNIVERSITY "POLITEHNICA" OF BUCHAREST
FACULTY OF ENGINEERING IN FOREIGN LANGUAGES
ELECTRONIC ENGINEERING, TELECOMINICATIONS AND INFORMATION TECHNOLOGIES –APPLIED ELECTRONICS

DIPLOMA PROJECT

Project coordinator:
Conf. dr. ing. Iuliana MARIN

Student:
Ionuț-Alin CONSTANTIN

Bucharest 2023

UNIVERSITY "POLITEHNICA" OF BUCHAREST
FACULTY OF ENGINEERING IN FOREIGN LANGUAGES
ELECTRONIC ENGINEERING, TELECOMINICATIONS AND
INFORMATION TECHNOLOGIES – APPLIED ELECTRONICS

COLLABORATIVE PLATFORM FOR SOFTWARE DEVELOPERS

Project coordinator:
Conf. dr. ing. Iuliana MARIN

Student:
Ionuț-Alin CONSTANTIN

Bucharest 2023

UNIVERSITY "POLITEHNICA" OF BUCHAREST
FACULTY OF ENGINEERING IN FOREIGN LANGUAGES
ELECTRONIC ENGINEERING, TELECOMINICATIONS AND INFORMATION TECHNOLOGIES – APPLIED ELECTRONICS

PLATFORMĂ COLABORATIVĂ PENTRU INGINERII DE SOFTWARE

Project coordinator:
Conf. dr. ing. Iuliana MARIN

Student:
Ionut-Alin CONSTANTIN

Bucharest 2023

"POLITEHNICA" UNIVERSITY OF BUCHAREST
FACULTY OF ENGINEERING IN FOREIGN LANGUAGES
ELECTRONIC ENGINEERING, TELECOMMUNICATIONS AND INFORMATION TECHNOLOGIES - APPLIED ELECTRONICS

Approved
Dean:
Prof.dr.ing.Cristian DRAGOMIRESCU

DIPLOMA PROJECT THEME FOR:
Ionuţ-Alin CONSTANTIN

1. Theme title:
   Collaborative Platform for Software Developers

   Platformă Colaborativă pentru Dezvoltatorii de Software

2. Initial design data:
   The goal of this project is to develop an application that can help the software developers which can't finish a project. They can upload the unfinished code and other programmer can take it, solve the problems and finish it.

3. Student contribution:
   - summarize relevant articles
   -create the database
   - design graphical user interface
   - implement required code in VS Code

4. Compulsory graphical material:
   Explanatory diagrams, code snippets

5. The paper is based on the knowledge obtained at the following study courses:
   Software Engineering

6. Project development environment:
   Visual Studio Code, C#, HTML, CSS, JavaScript, React, NodeJS, NoSQL

7. The project serves as:
   Didactic purposes

8. Paper preparation date:
   June 2023

Project coordinator

S.L. Dr. Ing. Iuliana MARIN

Student:

Ionuţ-Alin CONSTANTIN

Academic Honesty Statement

I, Constantin Ionuț-Alin, hereby declare that the work with the title "Document Management System", to be openly defended in front of the diploma theses examination commission at the Faculty of Engineering in Foreign Languages, University "Politehnica" of Bucharest, as partial requirement for obtaining the title of Engineer is the result of my own work, based on my work.

The thesis, simulations, experiments and measurements that are presented are made entirely by me under the guidance of the scientific adviser, without the implication of persons that are not cited by name and contribution in the Acknowledgements part.

The thesis has never been presented to a higher education institution or research board in the country or abroad.

All the information used, including the Internet, is obtained from sources that were cited and indicated in the notes and in the bibliography, according to ethical standards. I understand that plagiarism is an offense and is punishable under law.

The results from the simulations, experiments and measurements are genuine. I understand that the falsification of data and results constitutes fraud and is punished according to regulations.

Ionuț-Alin CONSTANTIN 15/06/2023

Table of Contents

Table of Figures

1 Introduction

1.1 Description

Coherence and knowledge sharing are the major drivers of innovation and progress in today's rapidly evolving software development landscape. I am introducing ProjectVault in order to recognise the importance of developing a platform that facilitates an easy exchange of ideas, solutions and projects. ProjectVault is an innovative storage platform to be devoted exclusively to software developers that enables them to post and share their work with a passionate community of likeminded individuals. ProjectVault has been designed to change the way developers operate, learn and work in order to foster collaboration and provide a repository of diverse projects. ProjectVault is at the forefront of collaborative development platforms, offering a user-friendly interface, high features and commitment to security and reliability that unlock new opportunities for growth, education or innovation.

ProjectVault is revolutionizing the way software developers upload, share. And search for projects through open development platforms that prioritize security and inclusivity. By fostering seamless collaboration and discovery of new knowledge. ProjectVault accelerates innovation in the software development process and creates meaningful connections among developers. With a sophisticated set of features user-friendly interface and unwavering commitment to security ProjectVault is poised to become the go-to platform for developers seeking new possibilities and reshaping collaborative programmings future. In terms of uploading projects ProjectVault service

makes it easy for developers to distribute their software creations with the worldwide software development community. Whether complex or simple. Developers can showcase their work on ProjectVault with relative ease thanks to an intuitive interface that allows them to provide greater detail about their projects' purpose, functionality, dependencies, and documentation. This comprehensive overview makes it easier for developers to express themselves effectively while attracting partners.

1.2 Seamless Upload and Showcase

Experience greater ease in uploading your software projects using ProjectVault's streamlined service. The platform's seamless process allows developers worldwide to distribute all sorts of creations- from simple scripts to high-level applications- without difficulty. The user interface is not only intuitive but also informative, providing details about the project such as its purpose, functionality, dependencies and documentation for better visualization. Developers can now articulate effectively when presenting worthwhile ideas and attract partners at every turn by offering thorough overviews of each creation.

1.3 Discover and Explore

One of ProjectVault’s most notable features is the wealth of diversity contained within its vast collection of projects. From frameworks to libraries to applications, developers worldwide have contributed solutions that span multiple areas of interest providing a multitude for programmers possess unlimited potential within reach. Using such powerful resources becomes more feasible with ProjectVault's robust filtering mechanisms designed assist determined users in locating relevant programs through use parameters such as programming language or project type while making their individual development sequences faster by tapping into pre-existing techniques. By empowering peer collaboration and knowledge exchange, ProjectVault serves as both an incubator for innovative ideas while at once propelling forward advancements in technology.

1.4 Collaboration and Connection

The aim of ProjectVault is to encourage collaboration amongst developers by providing them direct access to one another for social interaction initiatives. This can be achieved through coordinated utilization of collaborative tools such as project specific forums or code snippet sharing facilities or even the more complex features involved in integrated project management mechanisms Thus helping team members across vast geographic areas work together more

cohesively in order reach desired outcomes as a cohesive group unit. By maximizing connectivity, virtual teams are able brought into existence with clearly designated focuses on idea coordination feedback provision among other important things while ultimately supporting each other's growth within the space.

1.5 Security and Privacy

Keeping developer projects secure and private is of paramount importance at ProjectVault. I ensure data contained within said projects is safeguarded through impenetrable encryption algorithms preventing any illegal access attempts. Furthermore, project backups and redundant measures provide us with adequate means to guarantee project availability and maintained integrity. Our strict privacy policy aligns with our values in respecting an individual’s intellectual property rights, allowing them full transparency as their work is shared or used by others. At each developer's discretion, a license can be created outlining certain terms relating to usage rights; we at ProjectVault fully respect this decision-making process which informs our security & privacy guarantees fostering trust within our community.

1.6 Motives for using a ProjectVault

ProjectVault is highly valued by developers who appreciate its many benefits: collaboration opportunities, increased visibility, easy access to diverse projects, simplified work and project management processes, skill acquisition opportunities through targeted learning experiences. It also provides feedback loops that boost performance levels while encouraging innovative approaches in software development practices. Participating in this thriving community means contributing toward shared knowledge while leveraging collective intelligence that results in accelerated progress in software development. Simplifying workflows for more efficient project development is at the core of what ProjectVault strives for as it provides a central platform for collaboration and management purposes in software development. The resourceful platform equally serves both newbies and experienced professionals within its community.

One exceptional advantage that ProjectVault presents stems from its high-value insight-sharing capabilities through shared projects that allow programing experts imbibe various techniques proffered not only by best practices but also by industry trends thereby amplifying skill level positively.

The substantial emphasis ProjectVault places on knowledge sharing creates an atmosphere that encourages lifelong learning across technical know-hows just as much as professional rites of passage for software engineers.

Furthermore, ProjectVault's continual strive towards excellence includes constructive feedback mechanisms designed to create continuous improvement opportunities for problem-solving such as bug fixes or even suggestions because it helps refine individual developer projects.

1.7 Advantages of storing projects and share unfinished projects

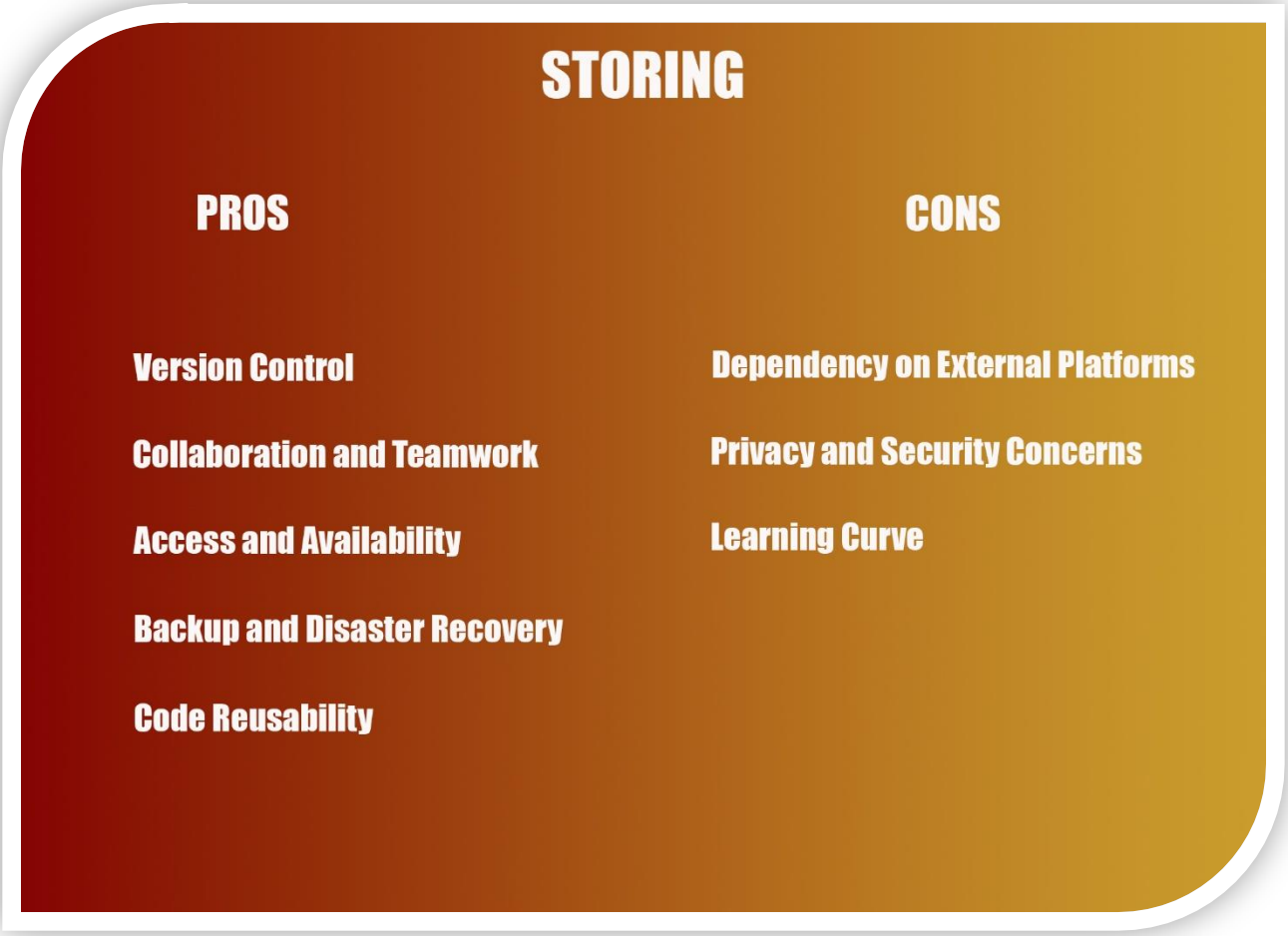

Figure 1: Storing projects pros&cons

To foster innovative software development practices effectively; it's crucial to share ongoing projects and maintain storage for completed ones. Sharing our work as developers ensures that we tap into a collective pool for better collaboration, increased innovation and faster learning.

Thanks to ProjectVault's ability to secure project storage; developers have a trustworthy means of preserving intellectual property with easy access when needed in future developments or updates. In this light; promoting collaborative sharing of ongoing developer work while organizing repositories for completed programming tasks is vital for significant contributions by all towards continued growth within our collective software development communities.

Collaboration on incomplete projects presents a valuable opportunity for exchange of knowledge among professionals with varying perspectives. By extending an invitation to others, we can incorporate new insights into the development process that can improve project outcomes beyond what we could achieve alone.

Through this sharing process, developers learn from one another in interdisciplinary ways which fosters innovation while producing improved outcomes aided by collective feedback.

Storing these projects safely is critical for continuity purposes; protecting against possible loss scenarios centralizes data accumulated throughout previous stages of project developments providing access so developers may reuse code fragments or take advantage of past experiences thereby improving future metrics.

With the increasing use of centralized platforms for project storage comes an opportunity for enhanced collaboration among developers irrespective of their location or background. By breaking down geographical barriers that once separated potential collaborators, these platforms facilitate knowledge exchange that drives collective professional advancement. Additionally, records kept through these channels enable smooth transition when moving between activities while at the same time providing opportunities for reference, when necessary, at later times.

Figure 2: Not storing projects pros&cons

1.8 Structure of the paper

The purpose of this paper is to present an approach of storing and sharing projects by proposing a web-based application, called ProjectVault.

To describe the project this document is structured into seven main chapters as follows:

- Introduction
- State of the Art
- Requirements Analysis
- Design
- Implementation
- Conclusion

Where it is needed additional UML diagrams, tables and figures are present to help in understanding the features and functionalities of the project. At the end of the documentation the references are mentioned and a glossary that describes every abbreviation used.

The figures are created by using Canva (https://www.canva.com/) and Draw.io (https://app.diagrams.net/)

2 State of the art

2.1 Scientific state of the art

Continuous evolution characterizes current trends within the scientific community concerning optimal storage platforms for software developers. Recently, there has been a clear emphasis on boosting cloud-based storage solution performances through integrating distributed systems and parallel processing techniques into existing structures. Furthermore, experts continue investigating how new encryption and redundancy innovations could promote more secure management practices aimed at increasing overall resilience.

Distributed file systems: The development of effective distributed file systems for managing large-scale software projects is a priority in ongoing research. To enhance performance and scalability, researchers are exploring a range of techniques including sharding, replication, and caching.[1]

Data deduplication: The exploration of data deduplication methods aims to optimize software project performance by reducing storage requirements through the identification and deletion of redundant data. This technique is expected to boost overall system efficiency while also saving valuable space.[2]

Security and Access Control: I aspire to build safer storage platforms by researching how encryption technology, an access control mechanism, and secure sharing protocols can boost overall platform security standards. To prevent any data breaches or compromises in information confidentiality or integrity we explore various methods such as Attribute Based Access Control (ABAC), as well as Hierarchical Access Control techniques. ProjectVault prioritize establishing a trust-worthy environment where users feel confident that their sensitive data remains safeguarded from potential threats.[3]

Scalability and Performance Optimization: The pursuit of boosting scalability and performance in storage platforms involves various approaches studied by several researchers. Employing methods like caching, load balancing, or distributed processing is among them while streamlining data retrieval or indexing through optimization algorithms is another important line of inquiry in this domain.[4]

ProjectVault serves as a dependable repository designed specifically for storing data related to your software projects in one central location. Software development isn't always an individual process, and the goal was to create an environment where efficient transfer between team members could take place effortlessly.

The advanced feature set will help to achieve the objectives by facilitating improved productivity through innovation, fostering idea sharing through stronger collaborations among co-workers, and providing effective tools which streamline how tasks are handled, resulting in effective project planning. Since the success is the foremost priority at ProjectVault The result a smooth operational flow revolutionizing how to conduct successful software project management strategies, opening a world of possibilities for your breakthrough success.

2.2 Collaboration and teamwork

Everyone should believe that working closely with developers and teams is key to success-resulting in a strong focus on facilitation within the application's design. To make collaboration on joint projects as smooth as possible, ProjectVault provides numerous functions for code review, commenting and task assignment within our system. These tools help promote clearer communication amongst colleagues - resulting in more effective teamwork overall. [5]

2.3 Knowledge Sharing and Reusability

ProjectVault values the importance of knowledge sharing and collaboration among software developers. As such, the platform serves as a hub for sharing project templates, libraries, or code snippets between individuals in the industry.[6]

2.4 Security and Protection of Projects

ProjectVault knows how crucial it is for you to keep your project files secure and protected. The platform follows a rigorous security regime that involves sophisticated encryption methods, access controls along with safe protocols to prevent any kind of data breach or tampering while ensuring utmost confidentiality and integrity. By providing an unwavering commitment towards project safety and reliability.

2.5 Commercial state of the art

Storage platforms play a vital role in the software development ecosystem by providing efficient tools and infrastructure necessary for managing projects effectively. To keep up with today's competitive landscape demands, commercial-grade storage facilities require state-of-the-art technology features alongside improved business offerings to deliver better productivity gains while being cost-effective.

As a result of such demands from developers worldwide seeking innovative solutions catering specifically to their needs; many businesses have invested significant funds towards research whilst introducing secure code repositories integrated seamlessly into collaborative workflows through other tools; thus, creating more efficient working environments.

To evaluate these advancements further within storage platform providers ultimately shaping pivotal Industry expectations - one must scrutinize their essential components deeply, such as features; benefits resulting from competitive advantages offered within current markets through detailed analysis regularly revealing distinct patterns revolutionizing software developer practices over time persistently adapting towards business growth requirements.

2.5.1 GitHub:

The ethos at GitHub revolves around empowering developers by offering them a platform that's tailor-made for their unique needs. They've achieved this by centering their business strategy around promoting collaboration, fostering community engagement, and driving innovation in open-source software development. The overarching aim of GitHub is to facilitate an environment where developers can seamlessly manage their code repositories while also having access to several useful features.

Open-Source and Community Focus:

The availability of free public repositories on GitHub acts as a catalyst for creativity and ingenuity among developers globally. These repositories allow an individual or group to share their code efficiently with others in the open-source community while enabling them to contribute toward different projects simultaneously. Accordingly, this fosters innovation by promoting continuous learning through interaction with peers possessing diverse skills sets and talents.[7]

Developer-Centric Approach:

GitHub has made it a top priority to provide them with an exceptional array of potent version control functionalities, code review tools, issue tracking mechanisms and project management features- all available on an accessible interface which is highly intuitive.

Collaboration and Community Engagement:

With pull requests, project wikis, and discussions among its many features, GitHub empowers developers to work together efficiently while sharing their expertise.[8]

Integrations and Ecosystem:

With its broad array of integrations with mainstream development tools, CI/CD platforms, project management systems, and code editors, GitHub enables developers to enhance their productivity while gaining increased flexibility.

Enterprise Solutions:

For businesses seeking heightened data protection and control over their code repositories, GitHub offers GitHub Enterprise. With both cloud-based and on-site deployment options available, this product ensures flexibility while also providing improved security protocols. Furthermore, sophisticated administrative functions enable precise configuration of user access and project workflows.[9]

2.5.2 GitLab:

GitLab stands out as a comprehensive solution that delivers end-to-end capabilities for source code management, continuous integration/continuous deployment (CI/CD), and project collaboration. With the aim of helping organizations achieve faster software development and application delivery, GitLab's all-encompassing business strategy centers around offering a universal platform that boosts efficiency throughout development processes.

All in one DevOps Platform:

GitLab strives towards providing users with a single solution addressing all aspects of software development lifecycle. It achieves this by integrating features such as source code management, CI/CD pipelines, issue tracking, project management and collaboration tools. This not only simplifies toolchain complexity but also helps enterprises follow streamlined DevOps processes while promoting coordinated efforts within teams during various stages of the application development cycle.[10]

Open-Core Model:

GitLab adheres to an open-core business model, wherein GitLab's primary features are freely accessible as open-source, but supplementary enterprise-grade characteristics and utilities

are extended through commercial licenses. This approach cultivates a robust open-source community for GitLab while simultaneously generating income for its enterprise clients.[11]

Emphasis on cooperation and visibility:

Collaboration and visibility are essential during the development phase in GitLab. The platform places a strong emphasis on these values to facilitate efficiency among teams. With merging requests, code revalidations, inline feedback, and activity feeds available as features, project progress can be tracked effortlessly.[12]

Focus on Security and Compliance:

GitLab takes the issue of software security very seriously indeed. That's why the product has a range of powerful features designed specifically with this in mind - including custom-built scanning tools for both codes analysis and compliance management. With cyber threats increasing in severity every year, it's more important than ever before for organizations to have access to these kinds of tools when developing new software products - which is why Gitlab has made them a top priority for our users' peace of mind.[13]

2.5.3 Bitbucket:

The prime focus of Bitbucket's business strategy is geared towards constructing an adaptable and robust platform that can accommodate both team-based as well as enterprise-level needs with its unlimited capacity and flexibility features.

Flexibility and adaptability:

Bitbucket's mission is to deliver the ideal Scalable Platform for all teams irrespective of their size – whether it's for Small or Medium sized enterprises or Large Enterprises. Bitbucket offers an extensive selection of pricing plans and deployment alternatives such as cloud hosting or self-hosted instances.[14]

Integrating into Atlassian's ecosystem:

Within the extensive Atlassian Ecosystem, Bitbucket occupies a crucial position alongside various renowned instruments including Jira that aids in issue management and Confluence that promotes efficient collaboration. With its seamless integration with these tools, Bitbucket provides users with an excellent platform to work in tandem on diverse development and project management tasks.[15]

Focus on collaboration:

For software development teams looking to optimize their workflow by streamlining collaboration efforts, Bitbucket is the perfect tool for the job. As one feature-packed platform that includes Pull Requests, Code Reviews and Inline Comments among other things - all aimed at building effective workflows - collaborating around coding changes has never been more comfortable or straightforward for your team members.[16]

Security and access controls:

At Bitbucket, security is given utmost importance when it comes to code repositories. There are various measures in place to guarantee the protection and compatibility of source codes with industry standards: this includes Branch Permissions, two factor Authentication, and Encryption.[17]

2.5.4 Microsoft Azure DevOps:

Empowering agile practices in application development & delivery is the core focus at Microsoft's Azure DevOps - previously named as Visual Studio Team Services (VSTS) and Visual Studio Online. The platform offers efficient capabilities in every aspect of software planning, development, testing & deployment that significantly improves team efficacy. The business strategy heavily emphasizes the adoption of modern-day techniques within a cloud environment for an optimal customer experience.

Integration with the Microsoft ecosystem:

Azure DevOps makes use of the vast Microsoft ecosystem and blends well with widely used tools like Azure Cloud Services, Visual Studio IDE, and Microsoft Teams collaboration platform. This enables businesses to have a smooth development and deployment journey within this ecosystem.[18]

End-to-end DevOps platform:

With its extensive set of tools and services that cover every aspect of the DevOps lifecycle right from source controls up until release management capabilities for testing options alongside work tracking applications - Azure DevOps provides an all-inclusive solution aimed at streamlining organizational processes for delivering top-notch quality software with minimum hassles involved. By implementing this end-to-end approach organizations can experience greater efficiency in their overall developmental operations while maintaining excellent standards for software quality. [19]

Cloud native and hybrid capabilities:

Enterprises today need agile tools that can keep pace with rapidly shifting technological demands - and that's where Azure DevOps comes in. Using advanced cloud technologies that are available only through this game-changing platform, users get access to a whole range of extraordinary services designed specifically for modern-day development needs.[20]

Collaboration and Team Productivity:

To achieve enhanced teamwork as well as maximum productivity levels across diverse operations; Azure DevOps offers integrated features such as agile planning tools, efficient task management capabilities alongside real-time communication functionalities. In leveraging well-known Agile methodologies like Scrum & Kanban. Teams can gain greater insights into the intricacies of task progress whilst allocating resources optimally.[21]

Security and Compliance:

With the rise of cyber-attacks targeting critical business applications or online infrastructure companies mustn't overlook security issues when working with development tools like Azure DevOps. Recognizing this fact Azure DevOps employs numerous techniques to help foster a safe environment for organizations concerning their sensitive information or intellectual property. [22]

2.5.5 AWS CodeCommit:

AWS CodeCommit is a fully managed service designed to host private Git repositories in secure and scalable conditions. Supporting Amazon's aims to deliver seamless solutions that support the growth potential of developers in their sights, it's considered heavily available and completely integrated.

Integration with AWS ecosystem:

Coordinating well with various other tools such as: AWS CodePipeline used for continuous integration/deployment; AWS CodeBuild applied in both building and testing-together they operate under one harmonious body where code sources autonomously make their way across specific pipelines until they are ultimately deployed by means of AWS CodeDeploy. With this level of intuitiveness, it's simple for developers to harness all resources provided by Amazon throughout software development along with its subsequent distribution cycle.[23]

Security and Compliance:

AWS CodeCommit prioritizes these values for its clientele using several measures. These measures include encryption protocols implemented at points of transmission or storage

respectively to secure client information. Also, repository permission management is enabled via access control features to ensure that only authorized individuals can engage with sensitive data. AWS Identity and Access Management (IAM) further facilitates secure user authentication/authorization practices in partnership with AWS CodeCommit's specialized service offerings; essential for clients' adherence to compliance requirements - such as HIPAA and PCI DSS regulations.[24]

Scalability and performance:

Organizations today have diverse workloads, and AWS CodeCommit is built precisely for such scenarios. What sets it apart is the capacity to manage enormous repositories effectively. With this tool, developers can enjoy a scalable infrastructure that supports easy handling, storing, and collaborating on any repository size imaginable.[25]

Global availability and reliability:

Its source code repositories provide lasting durability and high availability, resulting in minimal downtime when accessing and working collaboratively on code from anywhere.[26]

Developer productivity and collaboration:

AWS CodeCommit follows an all-encompassing approach to improving developer effectiveness while striving to create an environment of teamwork. Its comprehensive list of features such as customized workflows for specific industries alongside pull requests, code reviews plus notification systems empowers teams to communicate efficiently amongst themselves. Moreover, integrating seamlessly with established AWS services like CodeBuild and CodeDeploy ensure that every stage of your development lifecycle proceeds smoothly without any hitches or downtime.[27]

2.6 Summary and comparison of existing approaches

Table 1: Comparison with existing applications

Storage Platform	Integration with Other Tools	Collaboration Features	Security and Compliance	Scalability	Pricing Model
GitHub	Extensive integration with a wide range of development tools and services.	Pull requests, code reviews, and project boards for effective collaboration	Strong emphasis on security with features like two-factor authentication and vulnerability scanning.	Highly scalable infrastructure to accommodate large code repositories and user base	Offers free plans for public repositories, paid plans for private repositories, and enterprise options.
GitLab	Built-in CI/CD capabilities and integration with popular DevOps tools.	Comprehensive collaboration features, including merge requests, inline feedback, and issue tracking.	Provides robust security features, including built-in CI/CD security scanning and access controls.	Scalable architecture supporting large-scale code repositories and distributed teams.	Offers various pricing options, including free plans for individuals, self-managed options, and enterprise editions.
BitBucket	Offers various pricing	Collaboration features such as pull	Offers advanced security	Scalable infrastructure to support	Provides free plans for small

	options, including free plans for individuals, self-managed options, and enterprise editions.	requests, inline commenting, and task tracking.	features like IP whitelisting, branch permissions, and encryption at rest.	teams of all sizes and repository growth.	teams, paid plans based on the number of users, and enterprise options.
Azure DevOps	Seamless integration with Microsoft ecosystem, including Azure cloud services.	Agile planning, work tracking, and real-time collaboration with features like boards and backlogs.	Strong security and compliance measures, including identity and access management controls.	Scalable and highly available infrastructure powered by Microsoft Azure.	Pricing based on the number of users and services used, with free options and enterprise plans available

3 Requirements analysis

3.1 Functional requirements

3.1.1 Actors and agents

The driving force behind ProjectVault is its proficient group of software developers responsible for generating concepts alongside executing file uploads & management procedures. Utilizing a diverse set of functions & modules incorporated within ProjectVault further aids them in optimizing work processes through effective communication practices among team members. Additionally guiding them in ensuring version control remains streamlined across all developmental phases.[28]

To oversee ProjectVault's complex system of operations, trained administrators take charge. They handle vital responsibilities such as monitoring user registration processes, ensuring seamless authentication protocols are in place while enforcing relevant security rules effectively. Moreover, through their efforts at streamlined access control mechanisms and efficient permission management tools; these professionals ensure the protection of sensitive information within the platform. Any difficulties faced by users are promptly addressed by these competent personnel.[29]

In ProjectVault, it is important to have skilled individuals who can manage software development projects with skillful coordination- these are referred to as Project Managers. They are responsible for utilizing various capabilities of ProjectVault like assigning tasks, setting goals & tracking deadlines while overseeing all activities related to such schemes. Project Managers leverage excellent collaboration & communication features of this platform whilst working together with developers & stakeholders pursuing coherent & efficient execution of each respective scheme. Collaboration among technical experts via particular projects is imperative too; Tools like Code review & Issue Tracking make contributions towards improved quality, process efficiency. Projectvault facilitates teamwork by streamlining communications & assisting innovative efforts.

3.1.2 Software use case diagram

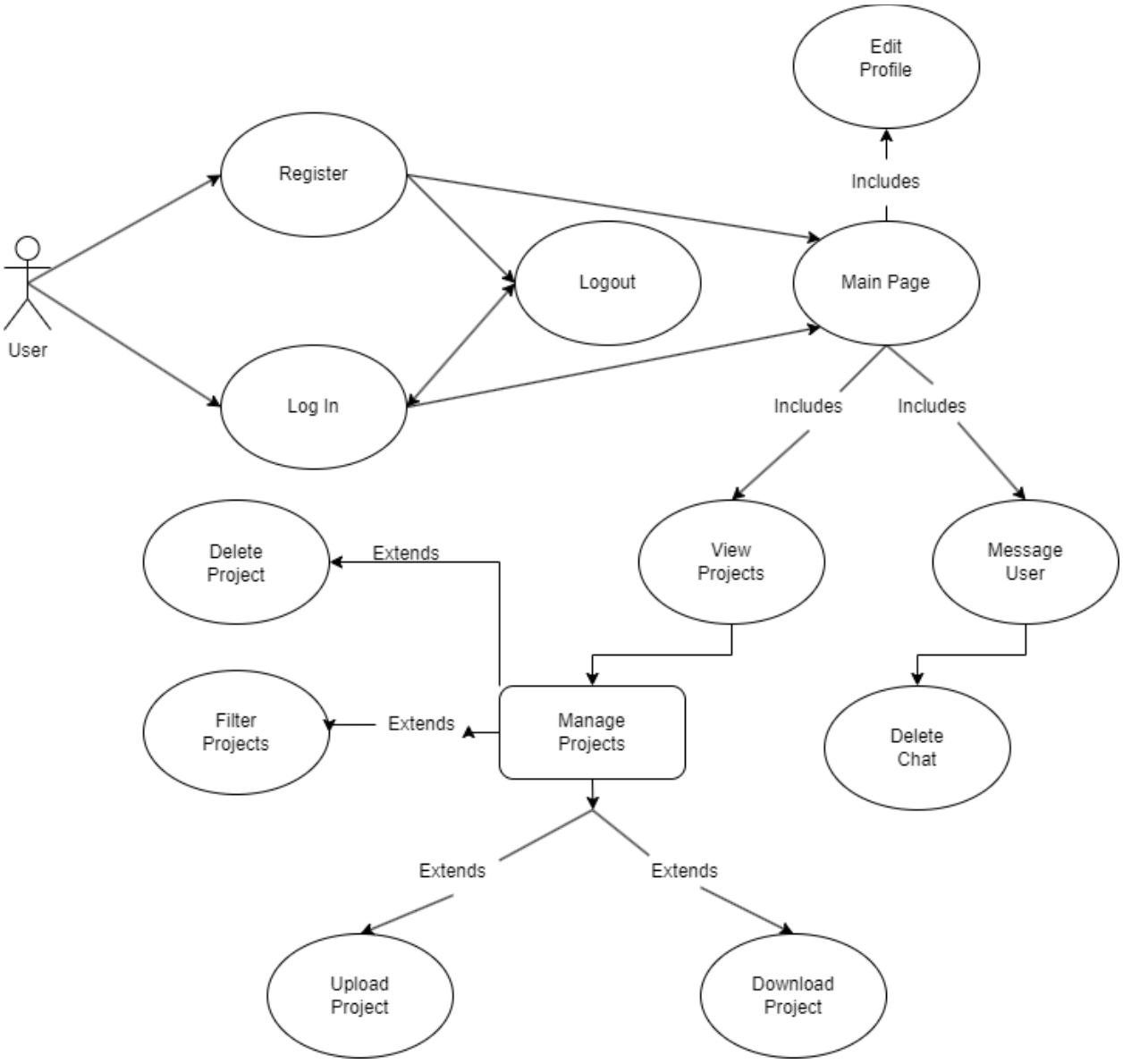

Figure 3: Software Use Case Diagram

3.1.3 Use case functional requirements

3.1.3.1 Log in – UC1

a) System shows login form

b) System validates form

c) System generates authentication token

d) System displays error message

3.1.3.2 Register – UC2

a) System shows register form

b) System validates form

c) System generates authentication token

d) System displays error message

3.1.3.3 Logout – UC3

a) System pop up menu

b) System validates form

3.1.3.4 View projects – UC4

a) System gets project model data

b) System displays project list

c) System provides filter inputs

d) System refreshes projects list

3.1.3.5 Manage projects – UC5

a) System displays document models management view

b) System adds new project

c) System filter project

d) System upload project

e) System download project

3.1.3.6 Filter projects – UC6

a) System provides filter inputs

b) System validates filter values

c) System shows error messages

d) System filter projects

3.1.3.7 Upload project – UC7

a) Systems shows upload area

b) System checks files formats

c) System stores the documents

d) System displays error message

3.1.3.8 Delete project – UC8

a) System displays project

b) System deletes project

c) System notifies user

3.1.3.9 Download project – UC9

a) System displays project

b) System download project

c) System saves project into user computer

3.1.3.10 Message user – UC10

a) System displays user chat

b) System enable messaging function

3.1.3.11 Delete chat – UC11

a) System displays chats

b) System deletes chat

c) System notifies user

3.1.3.12 Edit profile – UC12

a) System displays profile information

b) System validates new information

c) System update user profile information

d) System display error messages

3.1.4 Use case descriptions and system sequence diagrams

3.1.4.1 UC1

User Case Name: Log in

Scope: System

Preconditions: The system must run

Post-conditions: The home page is displayed

Main success scenario

User	System
1. Access website	
	2. show log in form
3. enter credential	
	4. validate credentials – A1
	5. authentication token
	6. display home page

Extension: A1. Incorrect username or password. Show error message

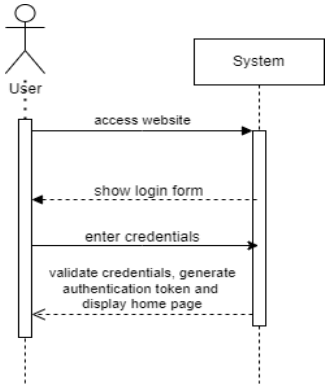

Figure 4: Log In Sequence Diagram

3.1.4.2 UC2

User Case Name: Register

Scope: System

Preconditions: The system must run

Post-conditions: The home page is displayed

Main success scenario

User	System
1. Access website	
	2. show log in form
3. enter credential	
	4. validate credentials – A1
	5. authentication token
	6. display home page

Extension:

A1. Incorrect username or password. Show error message

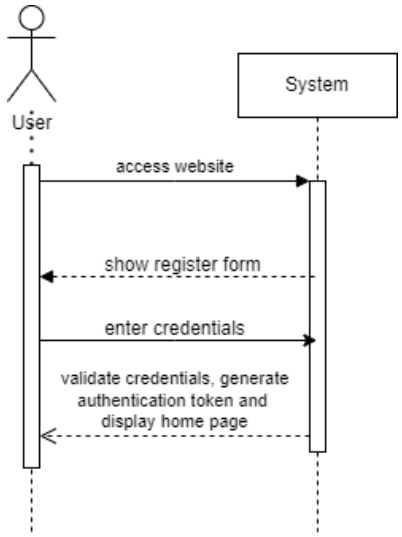

Figure 5: Register Sequence Diagram

3.1.4.3 UC3

User Case Name:	Register
Scope:	System
Preconditions:	Must be logged in
Post-conditions:	The log in/register form will be displayed

Main success scenario

User	System
1. clock on logout button	
	2. show logout popup
3. confirm	
	4. display login/register form

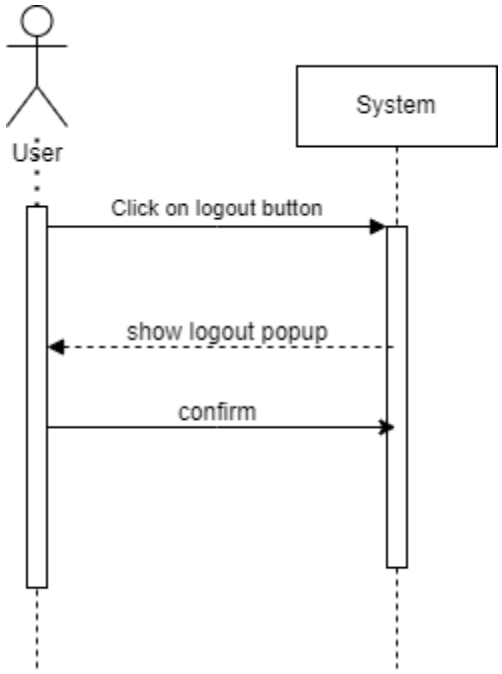

Figure 6: Logout Sequence Diagram

3.1.4.4 UC4

User Case Name: View Project

Scope: System

Preconditions: The user must be logged in

Post-conditions: The project page is displayed

Main success scenario

User	System
1. click on projects button	
	2. display projects page

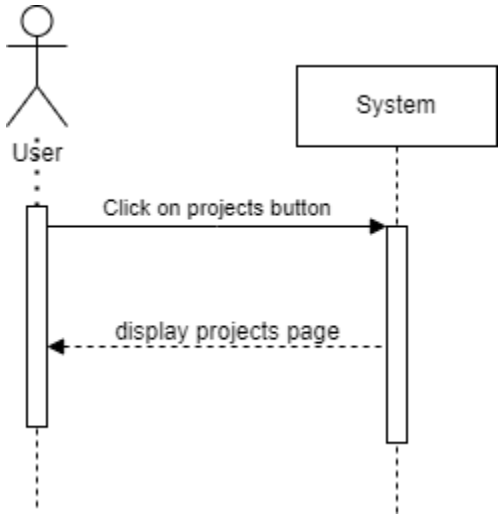

Figure 7: View projects Sequence Diagram

3.1.4.5 UC5

User Case Name:	Manage Projects
Scope:	System
Preconditions:	Must be logged in
Post-conditions:	Project is added, deleted, filtered and downloaded

Main success scenario

User	System
1. navigate to projects page	
	2. show projects page
3. select action(upload, download, filter, delete)	
	4. validate action – A1
	5. save changes

Extension: A1. Validation failure

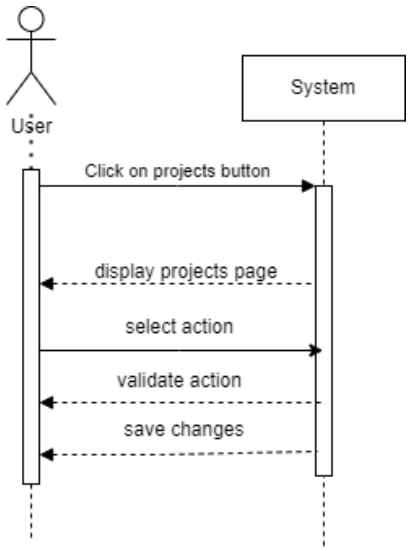

Figure 8: Manage projects Sequence Diagram

3.1.4.6 UC6

User Case Name:	Filter Projects
Scope:	System
Preconditions:	Current page is projects
Post-conditions:	Projects are filtered

Main success scenario

User	System
1. select filter	
	2. filter
	3. display changes

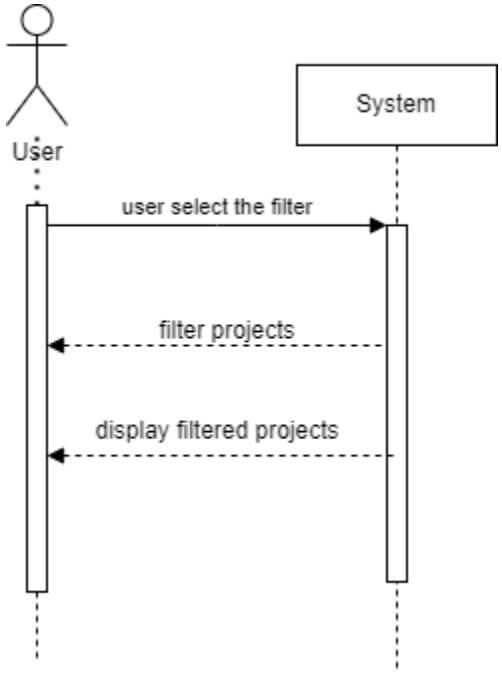

Figure 9: Filter projects Sequence Diagram

3.1.4.7 UC7

User Case Name: Upload Projects

Scope: System

Preconditions: User must be logged in

Post-conditions: New project is added

Main success scenario

User	System
1. click on upload button	
	2. show upload popup
3. select project	
	4. save project
	5. update projects list

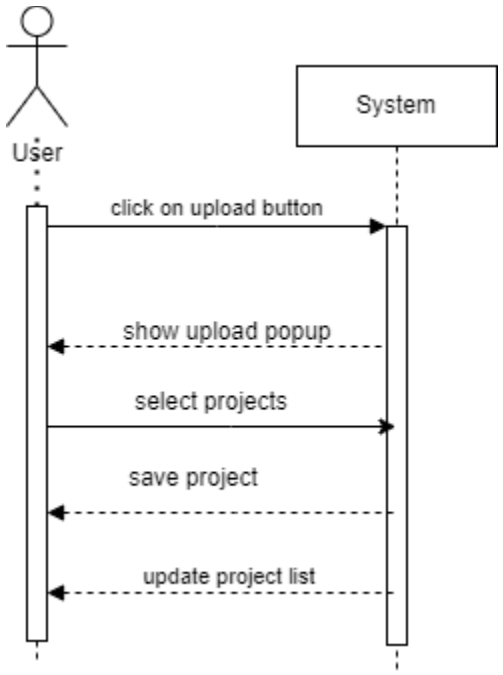

Figure 10: Upload projects Sequence Diagram

3.1.4.8 UC8

User Case Name: Delete Projects

Scope: System

Preconditions: User must be logged in

Post-conditions: Project is deleted

Main success scenario

User	System
1. click on delete button	
	2. display uploaded projects
3. select project to delete	
	4. delete project
	5. refresh projects list

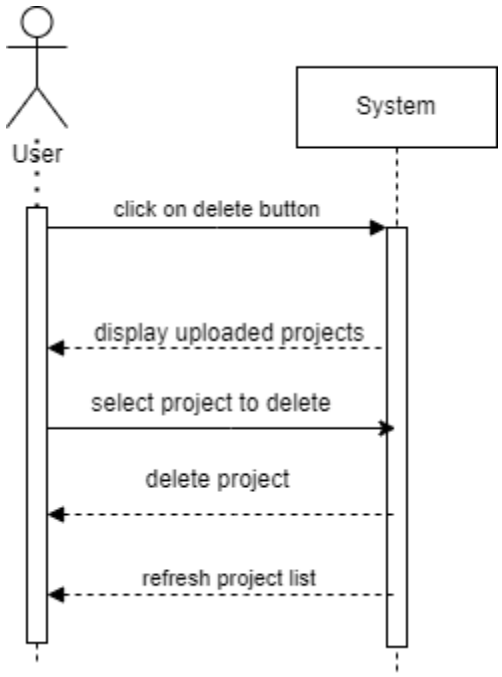

Figure 11: Delete projects Sequence Diagram

3.1.4.9 UC9

User Case Name: Download Projects

Scope: System

Preconditions: User must be logged in

Post-conditions: Project is downloaded

Main success scenario

User	System
1. click on download button	
	2. download project
3. select location	
	4. save project in computer

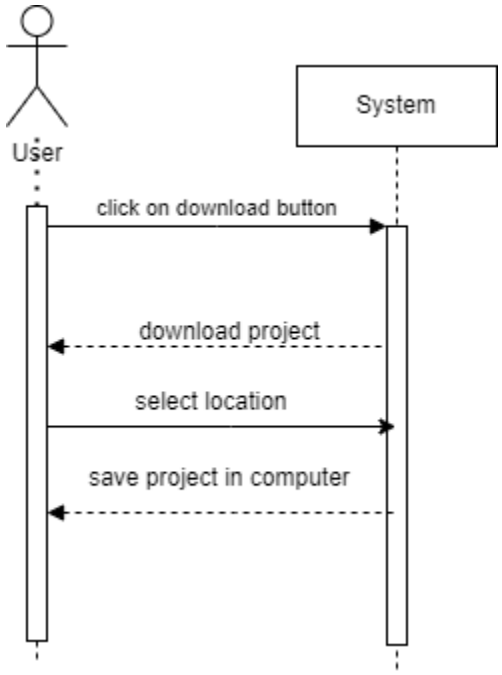

Figure 12: Download projects Sequence Diagram

3.1.4.10 UC10

User Case Name:	Message User
Scope:	System
Preconditions:	Click on messaging button
Post-conditions:	Enter message page

Main success scenario

User	System
1. click on chat button	
	2. display chat page
	3. enable messaging function

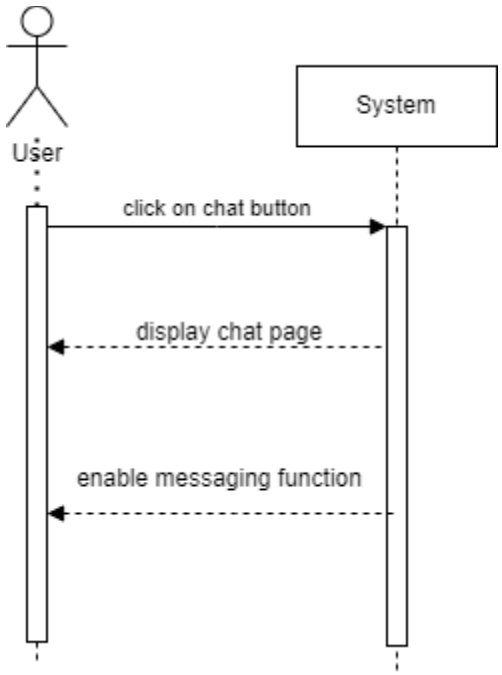

Figure 13: Message User Sequence Diagram

3.1.4.11 UC11

User Case Name: Delete Chat

Scope: System

Preconditions: Click on delete chat button

Post-conditions: Delete a chat

Main success scenario

User	System
1. click on delete button	
	2. confirmation popup
3. confirm	
	4. delete chat
	5. refresh messaging page

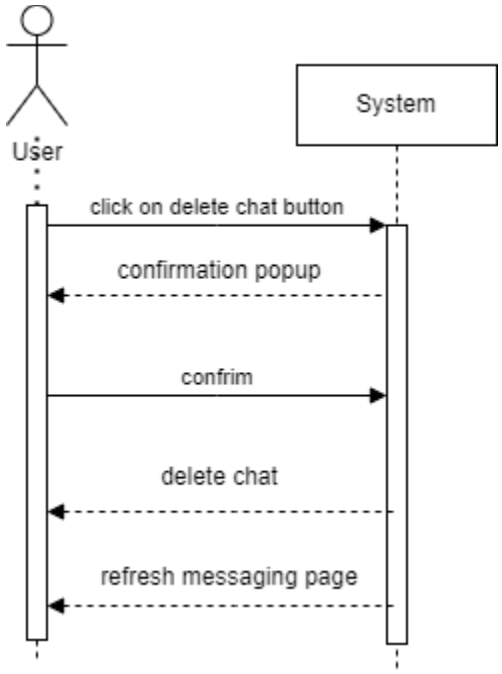

Figure 14: Delete Chat Sequence Diagram

3.1.4.12 UC12

User Case Name:	Edit Profile
Scope:	System
Preconditions:	Click on delete chat button
Post-conditions:	Delete a chat

Main success scenario

User	System
1. click on edit profile button	
	2. display profile page
3. edit information	
	4. validate information
	5. update user profile information

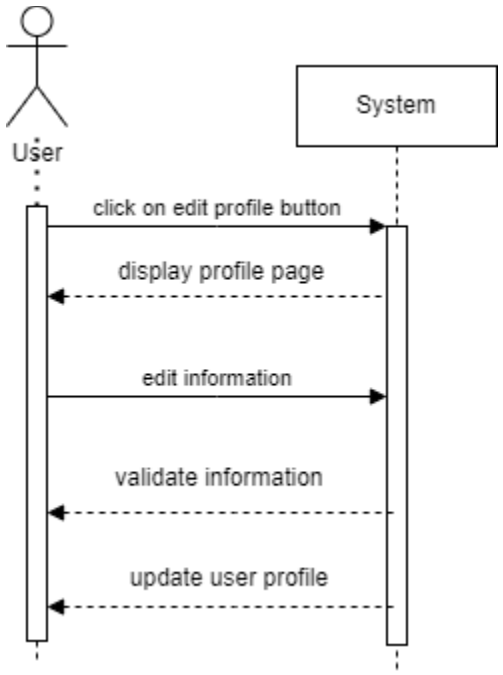

Figure 15: Edit Profile Sequence Diagram

3.2 Non-functional requirements[30]

Security:

Platforms have a key role in enacting stringent security measures aimed at safeguarding valuable user data including intellectual property and projects from potential breaches or losses due unauthorized access by third parties. It is crucial for system administrator teams implement secure authentication procedures particularly for systems incorporating multi-factor encryption ensuring only authorized use/entry into their accounts on the platform occurs. Without exception, encryption techniques should always be used when transmitting sensitive information over networks and requiring storage solutions whereby such approaches create fortified barriers protecting said sensitive information against thefts afforded by sophisticated hacking mechanisms.

Performance:

To offer users a seamless experience, the platform must possess top-notch responsiveness and ability to handle numerous users simultaneously without compromising on performance quality. Moreover, ensuring minimal downtimes while providing efficient file upload and download rates is indispensable.

Scalability:

It is necessary for ProjectVault to be designed in a scalable manner, as it may need to accommodate more users, projects and data storage in the future. Scalability can be achieved by adding servers or nodes to handle increased workloads and user bases.

Reliability:

Ensuring the platform's dependability and constant availability for users is highly significant. Putting in place efficient mechanisms to handle system failures helps reduce disruptions in project availability and user access.

Usability:

To ensure maximum user satisfaction on ProjectVault, it is crucial that the platform provide a highly accessible, intuitive interface for effortless navigation throughout the system. Simple project uploads and improved collaborative features will further elevate user productivity.

Compatibility:

It is crucial for platforms to prioritize user accessibility and compatibility by supporting a diverse range of operating systems, web browsers, and devices. Furthermore, integrating seamlessly with well-known development tools and frameworks can facilitate smoother workflow integration.

- **Collaboration and Communication**

To optimize team productivity on projects through coordinated efforts it is critical that effective means of communication between project members are established via a software solution like ProjectVault. To this end, features such as comments sections, notifications and real-time messaging should be integrated into its offerings for smooth flow of information in real time. Seamless compatibility with commonly used collaborative tools including project management systems alongside issue tracking systems would provide an invaluable asset to users keen on leveraging their productivity by utilizing ProjectVault services efficiently.

Support and Maintenance

Ensuring that platforms offer timely and responsive assistance is crucial in addressing user requests, issues, and feedback. Platform developers must also undertake regular maintenance and updates for ensuring platform stability, security along with being compatible with developing technologies.

4 Design

4.1 Architecture

4.1.1 Monolithic architecture

The monolithic architecture is an age-old software architectural technique where the entire application is built as a single unit in a tightly coupled system of modules, components and services running in one runtime process. With this architecture style comes uniformity across its code base; all features are developed together followed by comprehensive testing before being deployed together on the same scaling plane. For ProjectVault's purposes specifically, simpler development with less complex communication between its parts can be achieved due to its single code base.[31]

At ProjectVault, the monolithic architecture facilitates seamless integration of all features, enabling smooth data exchange, collaboration and efficient workflow management within the platform.

Additionally, this architecture provides excellent performance by co-locating all components and communicating through in-memory function calls. This efficient communication eliminates network overhead, yielding faster response times and lower latency.

The easy scalability of a monolithic application allows for quick adaptation to increasing user demands and workloads. With the ability to scale up or out, small teams or startups with limited resources can efficiently manage their growth.

The simplified deployment process is made possible by bundling all components into a single unit. This makes it easy to deploy updates and bug fixes without worrying about compatibility issues.

Lastly, the cost effectiveness of our monolithic architecture eliminates the need for complex distributed systems and infrastructure. This reduces operating costs for small development teams and startups with limited resources.

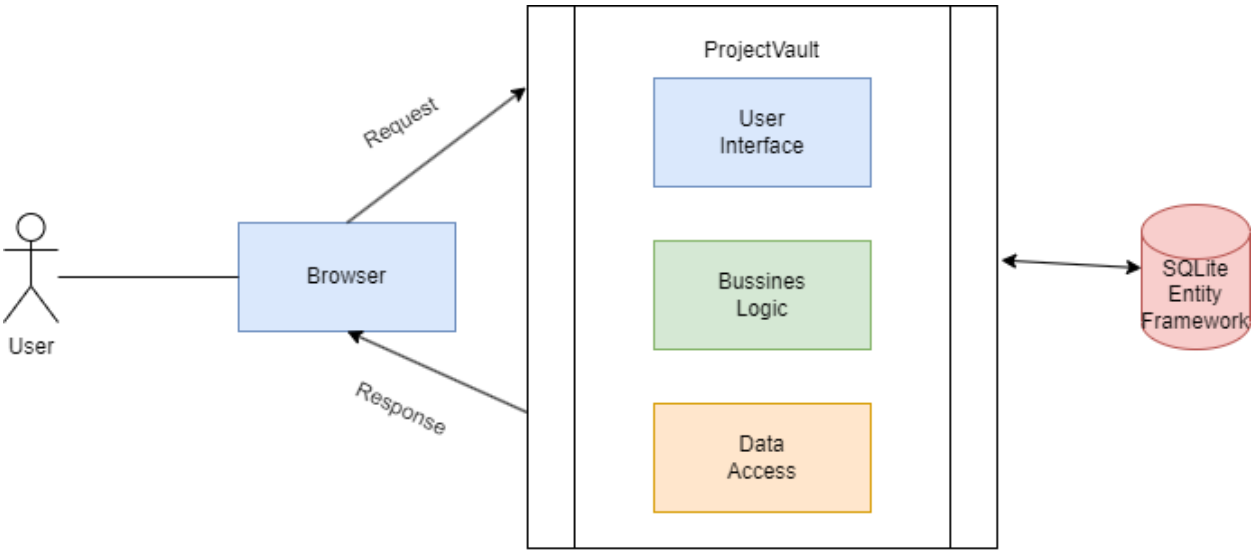

Figure 16: Architecture for ProjectVault

4.1.2 Deployment

The software development life cycle's deployment phase is crucial for ensuring that users have access to your application. When it comes to ProjectVault, deploying it using the monolithic architecture process demands careful attention and preparation.[32]

- **Environment selection:** For optimal results, careful consideration must go into choosing a compatible deployment framework. There are various valid choices such as using on-premises hardware or leveraging suitable cloud solutions offering required infrastructure and resource support. In making this selection decision factors like scalability options available for expansion outgrowths over time alongside high system uptime reliability levels accessibility requirements for data protection against threats must be taken seriously coupled with reasonable cost analysis enabling you to pick an ideal setup finely tailored per need purposes only after close examination.
- **Infrastructure setup:** It is important to configure and optimize infrastructure adequately to support anticipated user load and data storage obligations.
- **Compile and package your code**: Compile the ProjectVault application code and package it into a deployable artifact. Build your entire monolithic application, including all modules, libraries, and dependencies, as a single entity.

- **Configuration management:** To ensure seamless deployment, is recommended to compile the application code and integrating it into a deployable artifact.
- **Deployment automation:** Efficient delivery can be achieved by implementing automation tools and processes. By utilizing technologies like continuous integration or continuous deployment (CI/CD) pipelines to automatically carry out key tasks including monolithic application building, testing, and deploying; risks associated with manual labor are mitigated whilst accelerating project completion times significantly.
- **Scalability and performance:** A successful monolithic architecture requires careful foresight into scalability and performance optimization planning. By incorporating load balancing techniques, caching mechanisms, and resource allocation strategies into the framework of your operations, you can meet consumer demand growth. Moreover, application performance must be continuously monitored and analyzed to proactively detect potential bottlenecks that could compromise your system's optimal functioning.
- **Safety measure:** A crucial step towards securing the sensitive information associated with ProjectVault and its users would be to implement sound security measures. Such measures should encompass many aspects. Enforcing reliable coding practices is essential, stringent authentication and authorization mechanisms should be put into place and implementing cryptography protocols for data protection while underway or stored is also integral.

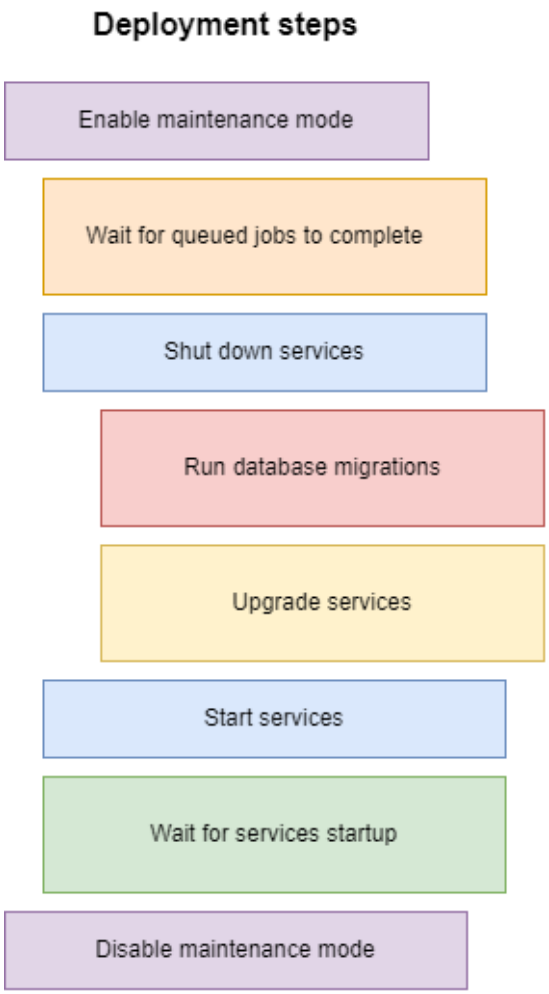

Figure 17: Deployment Steps

5 Implementation

5.1 Technologies

5.1.1 Development

For the web application's frontend architecture, I adopted Angular 16 together with TypeScript Bootstrap open-source CSS framework features such as Bootstrap5 while also utilizing HTML5 and CSS solutions resulting in a well-rounded user experience.

TypeScript is a superset of JavaScript based on better structuring capabilities due to its strict coding standards.

This leads to a more consistent coding practice, which many developers worldwide prefer when using Angular, a framework specifically designed for building single-page client applications.

Bootstraps' extensive range of predefined classes remains useful in improving both visual aestheticism and range usability on web pages, thanks to the UI-customized templates featuring interface components targeted precisely towards fulfilling User Interface (UI) requirements.

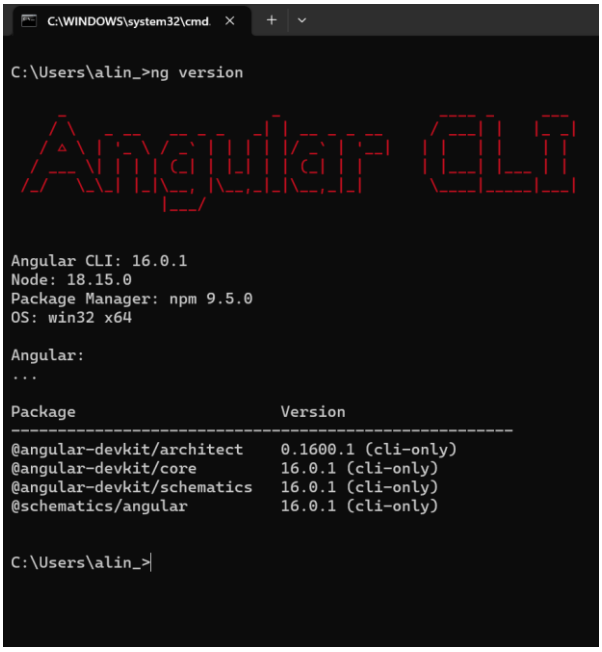

Figure 18: AngularCLI Version

The decision to choose a programming framework holds immense value when undertaking a software development endeavor. For ProjectVault, picking the right one means ensuring efficacy in development processes and scalability in performance outcomes. Following research-based analysis, I concluded that .NET 7.0 was best suited owing to its multiple advantages such as:

- Streamlined performance: The incorporation of .NET 7.0 introduces several noteworthy safety nets to ensure top-notch execution that is sure to please users. These enhancements include optimized runtime processes, quicker startups and reduced memory consumption - all aimed at bolstering ProjectVault's overall productivity while ensuring a seamless and responsive user experience.
- By introducing fresh features and enhancements to C# and the .NET Framework, .NET 7.0 empowers developers to produce more elegant, expressive ProjectVault code. The result is a productive development process that promotes efficient maintenance.
- .NET 7.0's latest improvements have brought vital benefits to organizations seeking to deploy and manage ProjectVault on modern containerized platforms like Docker and Kubernetes. By elevating their level of support for containerization technologies, users can enjoy increased scalability alongside optimized resource utilization within these environments at an unprecedented level.
- With .NET 7.0's cross-platform compatibility feature integrated into ProjectVault, it becomes feasible for developers to fashion an application that is operable across different operating systems such as Windows, Linux and macOS. Consequently, the platform becomes accessible to an extensive range of individuals who can leverage its advantages for their work or personal activities.
- The well-developed ecosystem and strong community support surrounding .NET make it an optimal choice for ProjectVault. This support enables access to extensive documentation, libraries, and ample community-driven resources that can expedite development processes while also reducing time to market.

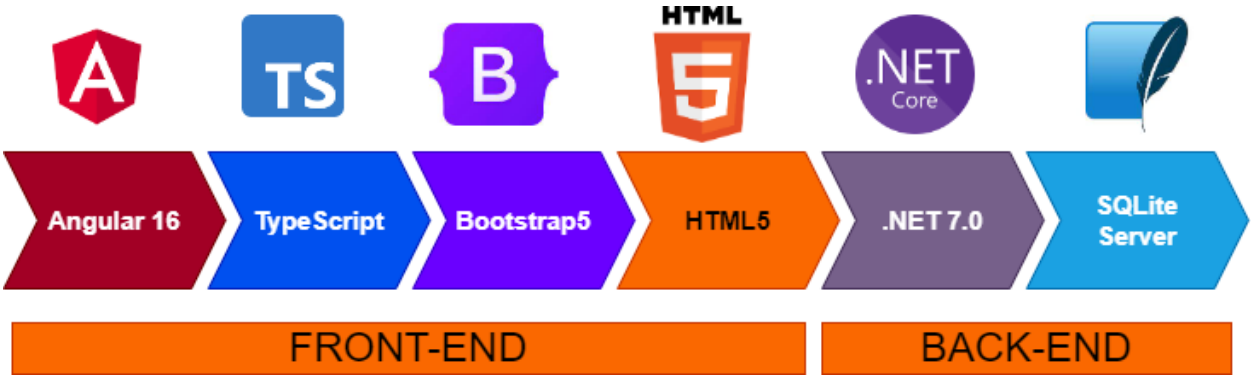

Figure 19: Technologies Used

5.1.2 Converting TypeScript in JavaScript

Developers choose the highly acclaimed Angular framework while constructing web applications with JavaScript often. The platform is built on the foundation of TypeScript. However, once it's time to deploy these applications, TypeScript must be transformed into JavaScript for seamless functionality across several browsers and tech-environments.

The convenience of development and build processes are significantly increased by utilizing the powerful features offered by the Angular CLI when working on an Angular project. Its capabilities include built-in support for converting TypeScript to JavaScript, providing more efficient workflows that save users' valuable time.

To start with this process, it's essential first to confirm that Node.js is fully set up on the device. Next, install the Angular CLI globally using the given instruction inside the command-line tool:

```
npm install -g @angular/cli
```

Figure 20: Angular Install

After that, the project must be build using the following command:

```
ng build
```

Figure 21: Build Application

JavaScript files are generated in the "dist" folder by default during the build process. Converted to a deployable format, these JavaScript files can be accessed from inside this same folder. Any web server or hosting platform must be used for running or hosting the converted code files. To prevent any hiccups along the way, it's essential to indicate which exact contents should be transferred to your server in the "dist" folder and determine ideal server configurations.

Optimizing an angular project for smooth conversion from TypeScript into JavaScript is achievable with great precision by leveraging tools such as Angular CLI and Node.js. The streamlined integration provided by these facilities automates most processes related to building and transitioning between languages, leaving behind resultant JavaScript files ready for deployment on any platform without issues. Additionally, thanks to being able to occupy oneself entirely with coding in TypeScript, developers can now streamline their focus maximally while letting go of any concerns related to managing extensive overhaul towards implementing an effortlessly sustainable developmental environment.

Figure 22: TypeScript to JavaScript

5.1.3 Entity Framework

The Entity Framework stands as one of Microsoft's popular ORM (Object Relational Mapping) frameworks, designed to offer an abstraction layer between the application and the underlying database in .NET applications. This paper aims to examine the core components and benefits of using Entity Framework while also exploring how it streamlines database-dependent application development.

Working with relational databases using Entity Framework can allow developers to leverage object-oriented concepts. Using simple mapping functions present in the framework, developers can map database tables, columns directly into a class properties allowing them to

interact with familiar object-oriented programming principles - thus simplifying data access layers reducing time spent on complex SQL queries.

Entity Framework's code-first methodology is particularly useful in defining corresponding database schemas for defined classes such as tables, columns, relationships, or constraints using C# or VB.Net. The framework automatically generates a schema based off this definition relieving developers from worrying about SQL commands basic syntax or any further management relating to these schema definitions.

The Language Integrated Query (LINQ) feature provided allows developers to write query strings that are type-safe and independent of any self-contained query language that you may choose for your application. Furthermore, LINQ provides several operators and functions that filter, sort or project data only making life easier for retrieving information from databases.

Finally, thanks to built-in features such as Migrations offered by Entity Framework adding features or updating existing ones becomes easier than ever before; there will be no need for data-loss since migration between entities is done efficiently through bi-directional communication leaving you miss nothing during upgrades ensuring access even after changes without any loss. Additionally, being able to interact with ASPNet/Core builds lets seamless seamless interaction between entities no matter their origin or tech stack. For those seeking to optimize their application's performance, the Entity Framework provides a formidable solution through its strong data access layer. This framework boasts support for key features like lazy loading, immediate loading, and caching capabilities which help to improve data retrieval efficiency.

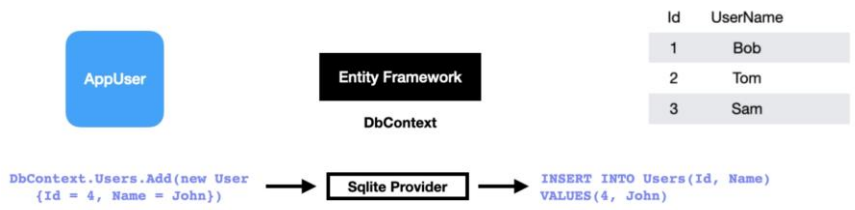

Figure 23: Entity Framework

5.1.3.1 Adding Entity Framework

A prerequisite for utilizing Entity Framework with VS Code is to have already added the NuGet Gallery extension and it needs to be installed two components: "Microsoft.EntityFrameworkCore.Design" and "Microsoft.EntityFrameworkCore.Sqlite"

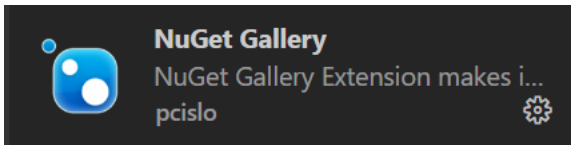

Figure 24: NuGet Gallery

When working on Entity Framework Core systems development, Mircrosoft.EntityFrameworkCore.Design package is crucial. With this key component, developers are offered an excellent design-time functionality that allow them seamlessly tackle tasks related to scaffolding and even more, database migrations. Mircrosoft.EntityFrameworkCore.Design gives devs access critical tools/APIs required create & manage these databases' redesign. Incorporating Migrations becomes even smoother using Mircrosoft.EntityFrameworkCore.

Visual Studio's Command Line Interface (CLI) or Package Manager Console generates significantly favorable outcomes, waiting until later may cause inconveniences. Explore options allowed by Mircosoft.EntityFrameowrkCore.SQLIte especially if the SQLite database engine deserves the integration for a flawless relational capabilities & enhanced performance. With Microsoft.EntityFrameworkCore.Sqlite, connecting .NET applications to SQLite databases is quite easy. This package contains all necessary APIs and components for developers to execute queries, retrieve data as well as data manipulations altogether with managing your databases' transactions.

5.1.3.2 Creating the Connection String

A connection string is a string composition consisting of different elements. It provides access to various databases or other critical sources of information. The crucial points embedded within these strings include details such as server identity/destination, identifying texts used for authentication, and several relevant options belonging solely to each data type being connected.

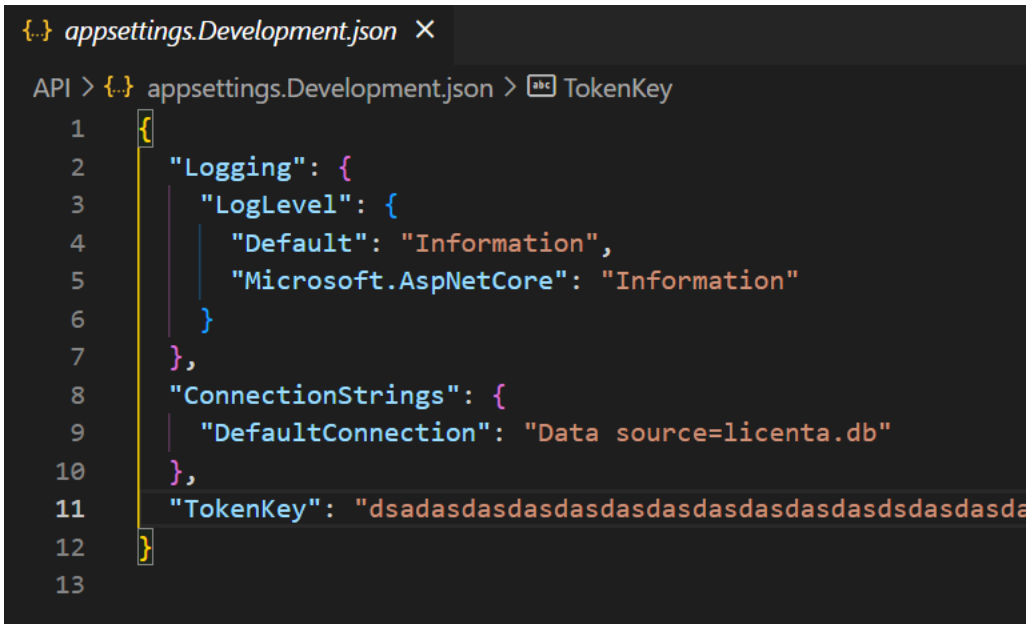

Figure 25: Connection String

For effective creation of the database, it is important to utilize "dotnet ef database update" once the connection string is established. This key command enables the application of necessary migration files onto the target schema, thereby ensuring uniformity with regards to the data model in use. All outstanding migrations are applied before any upgrades within your schema take effect.

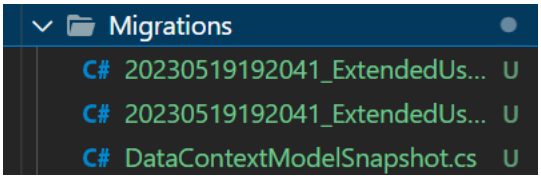

Figure 26: Migration Files

Testing all API as well as database requests calls for utilizing Postman significantly. This tool provides a user-friendly route for crafting HTTP queries while also receiving inline responses promptly from the APIs. Queries spanning across various areas like GETs through to DELETEs are all supported by this versatile platform which enables better accountability over interactions between development activities involving APIs.

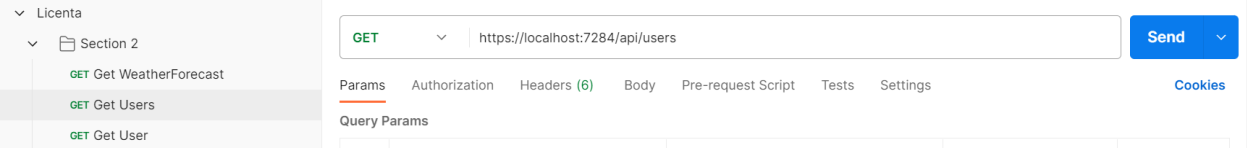

Figure 27: Postman Request

5.1.4 Asynchronous Code

Asynchronous code development methods are crucial for developers looking to maximize efficiency and improve software performance by running multiple operations concurrently without stalling their application's execution. These modern programming principles provide an innovative alternative to traditional synchronous protocols used in programming methodologies worldwide. When it comes to implementing ProjectVault, leveraging asynchronous coding patterns can unlock several key benefits significant to the app's end-users, such as:

- **Improved Responsiveness:** Empowering applications using asynchronous code improves responsiveness - even when performing long-running processes such as uploading or downloading larger files. With concurrency, applications enhance their capacity to deliver responsive resolutions while users can still interact with the software.
- **Improved Scalability:** Asynchronous coding techniques offer ProjectVault users' better scalability with this platform being given opportunities that highly favor scalability where adequate processing power is available. Applications can dedicate system resources to efficiently handle a considerable number of concurrent requests successfully.
- **Efficient Resource Utilization:** Application resource optimization results in resource utilization efficiency becoming an unmanageable challenge for synchronous application models. Asynchronously coded platforms like ProjectVault take resource management

seriously - achieving optimal states that maximize performance whilst avoiding blocking processes which often limit optima (often quite rapid turnaround times).

- **Improved Performance:** The use of non-blocking techniques in asynchronously designed applications measurably increases performance & stamina over time through minimizing idle periods and running operations concurrently proceeding without delay whenever possible keeps systems online many more hours per day than lacking these optimizations altogether! An additional advantage is an increase in CPU usage during mission-critical data transfers leading directly towards better consistency and sustainability within high-usage environments!
- **Concurrency & Parallelism:** For ProjectVault, executing essential concurrency is made possible through asynchronous code; ideal for running multiple tasks simultaneously. With the strategic employment of this architecture type can achieve significant speed improvements by allowing maximum throughput rates within all environments that scale well with size-making it great for large task-oriented projects or small startups.
-

```
1 reference
private async Task<bool> UserExists(string username)
{
    return await _context.Users.AnyAsync(x => x.UserName == username.ToLower());
}
```

Figure 28: Example of asynchronous code in ProjectVault

5.2 Authentication

5.2.1 JSON Web Token

Web applications commonly rely on JSON Web Tokens (JWT) as a reputable standard for crafting highly secure authentication tokens. Thus, allowing server-generated tokens containing individualized data that is digitally signed and sent to clients thus enabling controlled system access. The implementation of JWT technology within ProjectVault promises an enhanced user authentication experience boasting high-grade security while facilitating robust resource accessibility safeguards.

ProjectVault with JSON Web Tokens (JWT) offers secure authentications accomplishing such by producing digital signatures with customized data catering specifically towards each end-user thus ensuring data validity.

Moreover, thanks to the advanced design of stateless JWT technology, servers no longer require storing sensitive or confidential client information allowing increased safety measures while reducing internal operational costs thereby streamlining company workflow offering more dynamic flexibility efficiency within its resource management structure ultimately delivering the most precise and tailored approach in ProjectVault.

Security must always be a top priority, which includes managing authorization policies. Thanks to JWT functionality, ProjectVault is equipped to support a comprehensive access control structure utilizing tokenized data from custom claims augmented for each individual user, improving overall work environment safety protocols while simplifying security checks.

5.2.2 JWT Class Implementation

```
1 reference
public class TokenService : ITokenService
{
    2 references
    private readonly SymmetricSecurityKey _key;
    0 references
    public TokenService(IConfiguration config)
    {
        _key = new SymmetricSecurityKey(Encoding.UTF8.GetBytes(config["TokenKey"]));
    }
    2 references
    public string CreateToken(AppUser user)
    {
        var claims = new List<Claim>
        {
            new Claim(JwtRegisteredClaimNames.NameId, user.UserName)
        };

        var creds = new SigningCredentials(_key, SecurityAlgorithms.HmacSha512Signature);

        var tokenDescriptor = new SecurityTokenDescriptor
        {
            Subject = new ClaimsIdentity(claims),
            Expires = DateTime.Now.AddDays(7),
            SigningCredentials = creds
        };
```

Figure 29: TokenService Class

The System.IdentityModel.Tokens.Jwt namespace has a designated class called JwtRegisteredClaimNames that facilitates JSON Web Token (JWT) programming by providing set claim names frequently featured in these standards. Such names remain standardized and therefore guarantee specific data gets transmitted within respective JWTs consistently.

The SigningCredential sets up secure JWT signatures by gathering crucial elements like keys and signing algorithms into one package. Once provided with valid components, it prepares an impenetrable fingerprint guaranteeing reinforced safety measures within signature generation.

The token key will be read from the configuration file:

Figure 30: Configuration File

5.2.3 Authorization

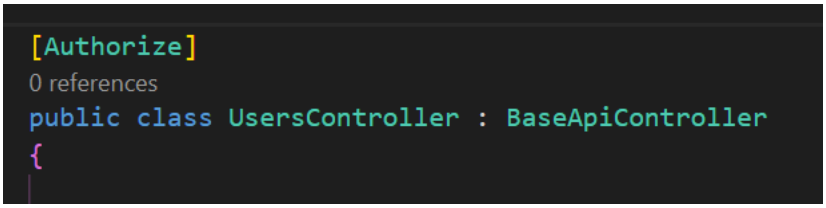

Figure 31: Authorization

A safeguard measure for ProjectVault is set in place for applications using its [Authorize] attribute, one that provides authentication and authorization functionalities. These are crucial protocols in maintaining the platform's overall security and integrity by ensuring individuals profiles or credentials before entry to restricted parts of the system is allowed. That way, sensitive materials remain out of reach from unauthorized entities.

```
public static IServiceCollection AddIdentityServices(this IServiceCollection services,
IConfiguration config)
{
    services.AddAuthentication(JwtBearerDefaults.AuthenticationScheme)
        .AddJwtBearer(options =>
    {
        options.TokenValidationParameters = new TokenValidationParameters
        {
            ValidateIssuerSigningKey = true,
            IssuerSigningKey = new SymmetricSecurityKey(Encoding
            .UTF8.GetBytes(config["TokenKey"])),
            ValidateIssuer = false,
            ValidateAudience = false
        };

    });
    return services;
}
```

Figure 32: Adding authentication

Developers frequently utilize the AddJwtBearer feature within ASP.NET apps (for instance, ProjectVault ones). By doing so, they enable JSON Web Token (JWT) authentication with diverse validation settings for tokens that fit specific security needs or other unique requirements.

Moreover, when building these applications programmatically, SymmetricSecurityKey objects become instrumental since they encompass both encrypting and validating mechanisms to be used solely on JWTs. During retrieval of APIs calling for said encryption/validation technology via Encoding.UTF8.GetBytes(config["TokenKey"]), developers take out token key values from source-based configurations to turn it into an array of bytes format before proceeding further.

Upon extraction from configs and byte-array transformation stages aforementioned above, developers ultimately have at their disposal an configured SymmetricSecurityKey object needed to perform arbitrary signature validation actions regarding JWTs exclusively. Validation actions regarding JWTs exclusively.

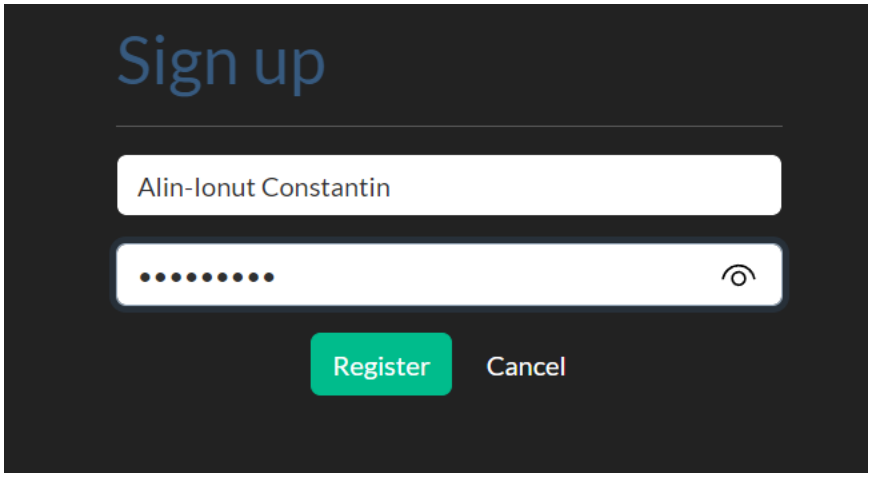

Figure 33: Design for Register page

5.3 Functionality

5.3.1 Upload and Download

```
[HttpPost("projects/upload")]
public IActionResult UploadProject([FromForm] IFormFile file)
{
    if (file == null || file.Length <= 0)
    {
        return BadRequest("No file uploaded.");
    }

    // Process the uploaded file (e.g., save it to storage)

    // Return a success response
    return Ok("Project uploaded successfully.");
}
```

Figure 34: Upload Function

The code offers an HTTP POST endpoint /projects/upload that permits receiving a form file (IFormFile) which stands for the uploaded project. I utilized the FromForm attribute to bind the uploaded file to the required parameter. Within this code, I included specific logic concerning

the file handling, like storing or processing it accordingly based on custom needs and desired actions related to the uploaded project.

```
export class UploadComponent {
  selectedFile: File;

  constructor(private http: HttpClient) { }

  onFileSelected(event: any): void {
    this.selectedFile = event.target.files[0];
  }

  onUpload(): void {
    if (this.selectedFile) {
      const formData = new FormData();
      formData.append('file', this.selectedFile);

      this.http.post('/api/projects/upload', formData).subscribe(
        () => {
          console.log('Project uploaded successfully.');
        },
        (error) => {
          console.error('Error uploading project:', error);
```

Figure 35: TypeScriptUpload Function

The system calls up the 'onUpload' function which carries out three essential tasks: First of all- creating a FormData object; next - adding onto file; thirdly - sending off an HTTP POST request towards what's dubbed as '/api/projects/upload' through usage of HttpClient module; finally- once the files reach there and something gets returned by server end-point (positive or negative feedback), I accomplish proper handling of this feedback through use of subscribe method which affords us with full freedom over performing additional essential actions based upon whether delivery was successful or not.

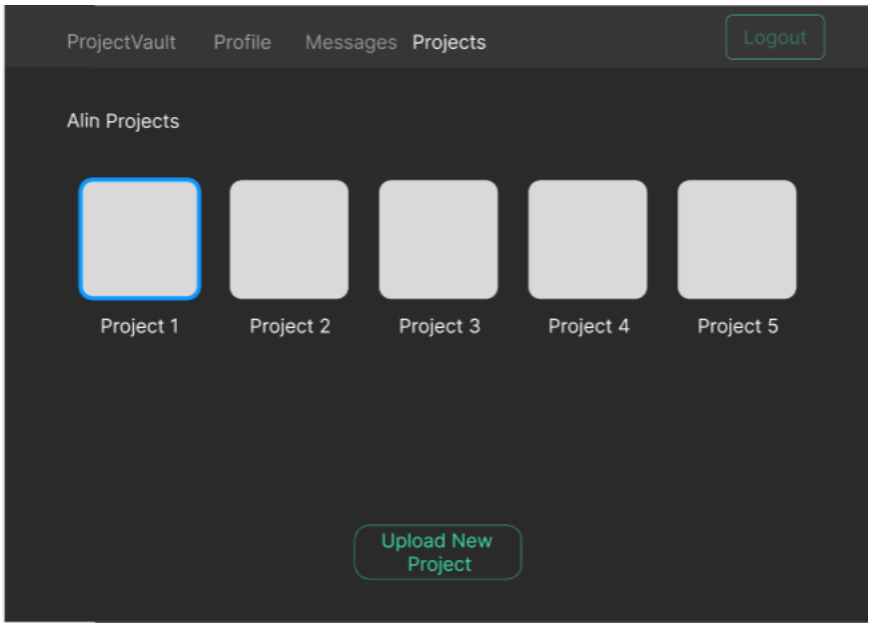

Figure 36: Project Page

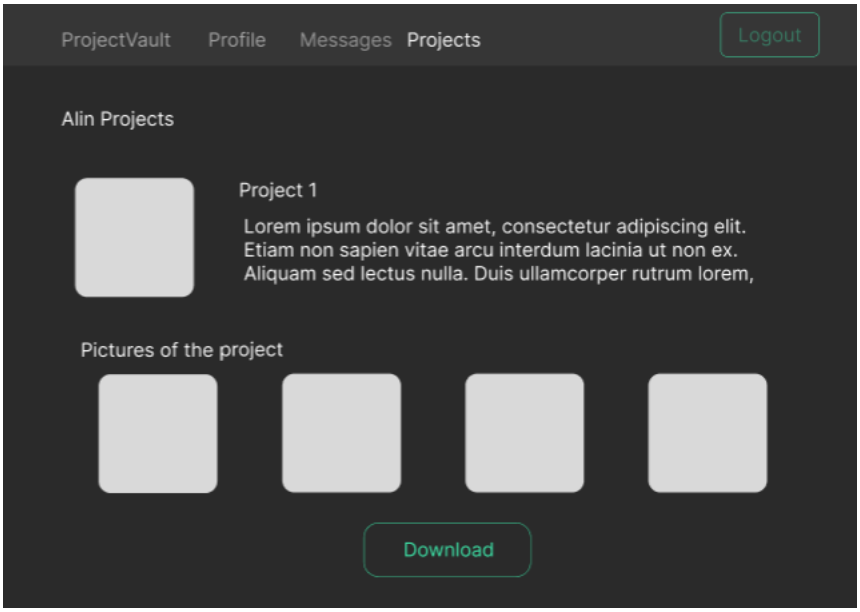

Figure 37: Download Project Page

5.3.2 Messaging

```
public class ChatHub : Hub
{
    public async Task SendMessage(string sender, string message)
    {
        await Clients.All.SendAsync("ReceiveMessage", sender, message);
    }
}
```

Figure 38: Message Function

Through proper utilization of the URL '/chatHub', establishing connectivity with ‘SignalR’ hub can lead to an excellent chatting experience. The vital feature responsible for message exchange among clients in this context is facilitated using 'SendMessage' method present within the powerful ‘ChatHub’ class. Additionally, it skillfully covers all necessary aspects and helps in broadcasting every single message through reliable means such as exciting 'ReceiveMessage' event.

```
constructor() {
  this.hubConnection = new signalR.HubConnectionBuilder()
    .withUrl('/chatHub')
    .build();

  this.hubConnection.start()
    .then(() => console.log('Connected to chat hub'))
    .catch(err => console.error('Error while starting connection: ', err))

  this.hubConnection.on('ReceiveMessage', (sender: string, message: string
    // Handle received message here
    console.log(`Received message from ${sender}: ${message}`);
  });
}

public sendMessage(sender: string, message: string): void {
  this.hubConnection.invoke('SendMessage', sender, message)
    .catch(err => console.error('Error while sending message: ', err));
}
```

Figure 39: TypeScript Message Function

The use of @microsoft/signalr library enables the ‘ChatService’ class to establish contact with a chat hub. To initialize this connection, an instance of ‘hubConnection’ is created. The constructor contains settings that dictate how this connection will operate, such as providing a URL for accessing resources on the chat hub (‘/chatHub’) and configuring an event handler for receiving messages ('ReceiveMessage' event). For transmitting messages from clients to servers, apply 'sendMessage', calling upon 'SendMessage' method via hub connection.

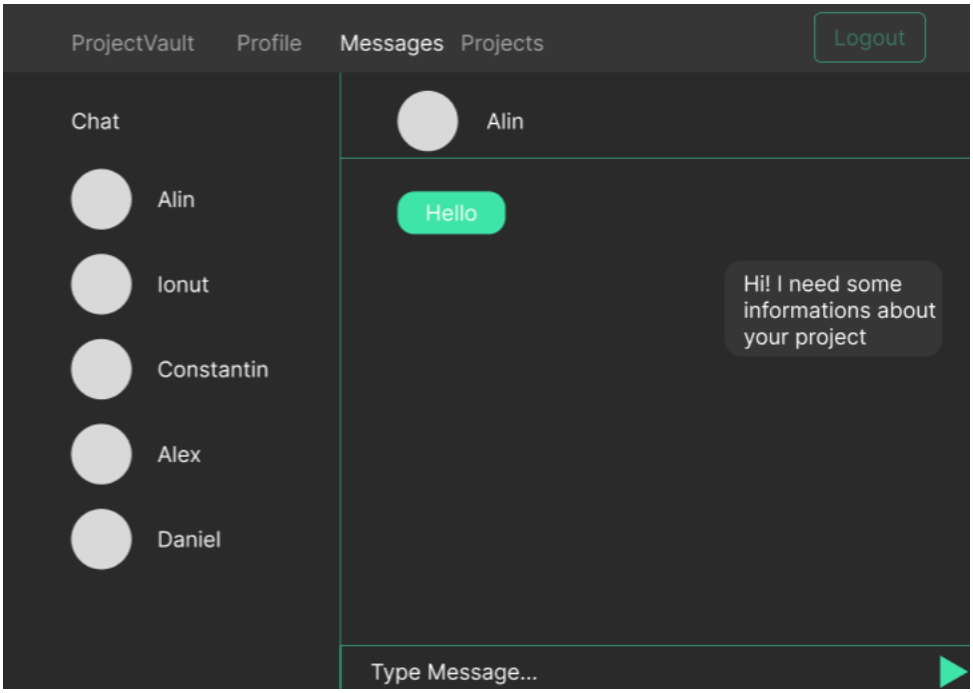

Figure 40: Message Page

Conclusions

ProjectVault is an impressive storage platform that caters comprehensively to software developers' unique requirements from storing to managing as well as securing their projects effectively. Developed with utmost care from its ideological phase down to its practical application phase, ensuring the integration of necessary functionalities that provide flexibility to meet diverse needs.

With an easily navigable user interface, developers can quickly upload, store information properly while maintaining organization while searching through previous works is made easy by advanced search features incorporated on the platform; hence improving productivity with ease via existing codes.

Project Vault has an outstanding collaborative capacity as compared to others in this niche market; this feature allows the developer's unfinished projects shared with their counterparts fostering growth bolstering creativity through teamwork ultimately leading efficient project execution processes.

Glossary

ABAC	Attribute Based Access Control
DevOps	Development Operations
CI	Continuous Integration
CD	Continuous Deployment
SMEs	Small and medium-sized enterprises
VSTS	Visual Studio Team Services
IDE	Integrated Development Environment
TFVC	Team Foundation Version Control
AWS	Amazon Web Services
IAM	Identity and Access Management
HIPAA	Health Insurance Portability and Accountability Act
PCI DSS	Payment Card Industry Data Security Standard
IP	Internet Protocol
UC	Use Case
CLI	Command Line Interface
REST	Representational State transfer
API	Application Programming Interface
JSON	JavaScript Object Notation
JWT	JSON Web Token
HTML	Hyper Text Markup Language
CSS	Cascading Style Sheets
UI	User Interface
EF	Entity Framework
SQL	Structured Query Language
VS Code	Visual Studio Code
LINK	Language Integrated Query

References

[1] Ghemawat, S., Gobioff, H., & Leung, S. (2003). The Google file system. ACM SIGOPS Operating Systems Review, 37(5), 29-43.

[2] Zhu, Y., Li, H., Hu, Y., & Hu, X. (2014). A survey of data deduplication in cloud storage. Frontiers of Computer Science, 8(1), 1-16.

[3] Sun, Y., Zhang, Y., & Yu, S. (2016). Cloud data security: A survey. Journal of Software, 27(1), 31-56.

[4] Dean, J., & Ghemawat, S. (2008). MapReduce: Simplified data processing on large clusters. Communications of the ACM, 51(1), 107-113.

[5] GitHub - https://github.com/, GitLab - https://about.gitlab.com/, Bitbucket - https://bitbucket.org/

[6] AWS Security - https://aws.amazon.com/security/, Microsoft Azure Security - https://azure.microsoft.com/en-us/trust-center/security/, Google Cloud Security - https://cloud.google.com/security/

[7] "GitHub: The world's largest code hosters" by Tim Anderson, The Register.

[8] "GitHub Unveils GitHub Discussions for Community Collaboration" by Mike Vizard, Container Journal.

[9] "GitHub Enterprise: The definitive guide" by Brandon Butler, Network World.

[10] https://about.gitlab.com/company/

[11] https://about.gitlab.com/blog/2016/06/03/what-is-open-core/

[12] https://about.gitlab.com/features/

[13] https://about.gitlab.com/security/

[14] https://www.atlassian.com/software/bitbucket

[15] https://www.atlassian.com/software/bitbucket/pricing

[16] https://www.atlassian.com/software/bitbucket/jira

[17] https://www.atlassian.com/software/bitbucket/security

[18] https://azure.microsoft.com/en-us/services/devops/

[19] https://docs.microsoft.com/en-us/azure/devops/?view=azure-devops-rest-7.1

[20] https://azure.microsoft.com/en-us/pricing/details/devops/azure-devops-services/

[21] https://visualstudio.microsoft.com/team-services/

[22] https://azure.microsoft.com/en-us/services/devops/security/

[23] https://aws.amazon.com/codecommit/

[24] https://docs.aws.amazon.com/codecommit/

[25] https://aws.amazon.com/codecommit/pricing/

[26] https://aws.amazon.com/codepipeline/

[27] https://aws.amazon.com/codebuild/

[28] Dabbish, L., Stuart, C., Tsay, J., & Herbsleb, J. (2012). Social coding in GitHub: Transparency and collaboration in an open software repository

[29] Goodall, D., Leventhal, A., & Dray, S. (2019). GitLab for DevOps: Master the configuration, deployment, and management of large-scale and distributed applications using GitLab

[30] "Software Requirements" by Karl E. Wiegers and Joy Beatty

[31] Bass, L., Clements, P., & Kazman, R. (2012). Software Architecture in Practice. Addison-Wesley Professional

[32] Newman, S. (2015). Building Microservices: Designing Fine-Grained Systems. O'Reilly Media.