Module Code & Module Title

CS6P05NI Final Year Project

Assessment Weightage & Type

5% FYP Proposal

Semester

2023 Autumn

PROJECT TITLE: APIDataFort

Student Name: Lasata Maharjan

London Met ID: 21049720

College ID: np01nt4a210079

Internal Supervisor: Mr. Prashant Pudasaini

External Supervisor: Mr. Suraj Neupane

Assignment Due Date: 29th November 2023

Assignment Submission Date: 28th November 2023

Word Count: 2000 words

I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded.

Acknowledgement

I express my sincere appreciation for the wonderful help and direction that Islington College's distinguished teachers and other faculty have given me. My academic development has benefited greatly from their constant commitment to creating a supportive learning environment. Their knowledgeable advice and experience have improved my comprehension of the study process and allowed me to carry out in-depth and significant studies. Their commitment to academic brilliance has motivated me to pursue information with constant passion and to aim for the greatest standards of knowledge.

To my supervisors, Mr. Suraj Neupane and Mr. Prashant Pudasaini, who have been the support structures of my research path with their essential guidance and mentorship, I extend my sincere gratitude. Through the difficulties of my study topic, Mr. Suraj Neupane's persistent support, prompt delivery of relevant details, and unfailing trust in my abilities have been invaluable. My study has always been motivated by Mr. Prashant Pudasaini's perceptive feedback encouragement to pursue new lines of research, and sincere enthusiasm. This project initiation would not have been completed without their continuous commitment to my accomplishment.

I would like to take this opportunity to express my sincere gratitude to the entire Islington College team for their constant support, encouragement, and direction during my academic career. Their support has been important in helping me grow both personally and professionally, making me what I am today. I sincerely appreciate the opportunity to be a part of this outstanding educational environment.

Abstract

API security has become an important concern for businesses as APIs are used more and more. By offering a comprehensive API security solution that protects sensitive data and guarantees safe API access, this initiative seeks to address this problem. The solution efficiently guards APIs against unauthorized access and data breaches by utilizing innovative security techniques including access control, authorization, and authentication. Furthermore, it offers organizations real-time access into API usage, allowing them to promptly detect and resolve possible security concerns.

To guarantee efficient planning, budgeting, and execution, the project follows the Scrum agile development process. Constant enhancement are encouraged throughout the development phase by establishing well-defined deliverables and milestones. Through the use of this powerful API security solution, companies can improve their entire security posture, safeguard confidential information from unwanted access, and obtain the visibility and control required to uphold a strong security posture. To sum up, these topics are briefly covered in this report.

Keywords: API, Data breach, Authentication, Authorization, Third-party API, ML, DL, Docker, Data

Table of Contents

List of Figures

Table of Tables

Table of Abbreviations

Abbreviations	**Definition**
API	Application Programming Interface
ML	Machine Learning
DL	Deep Learning
TLS	Transport Layer Security
SSL	Secure Sockets Layer

1. INTRODUCTION

1.1 Introduction to topic

We are all aware of how crucial APIs are in the present day. APIs bind the digital world together by facilitating seamless data interchange and communication between various software components. But because APIs are so widely used, cyber criminals can easily target them. Attackers can launch cyberattacks against other systems, steal confidential data, and interfere with essential business operations by taking advantage of API flaws.

Any business that employs or provides APIs should be aware that no technology is 100% secure. However, by taking precautionary actions to protect APIs, one can lessen the likelihood of attacks. This is where APIDataFort comes in. To protect API data from cyberattacks, unauthorized access, and data breaches, a system known as APIDataFort is strategically planned to be created.

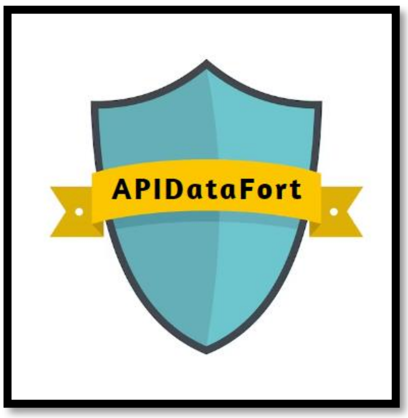
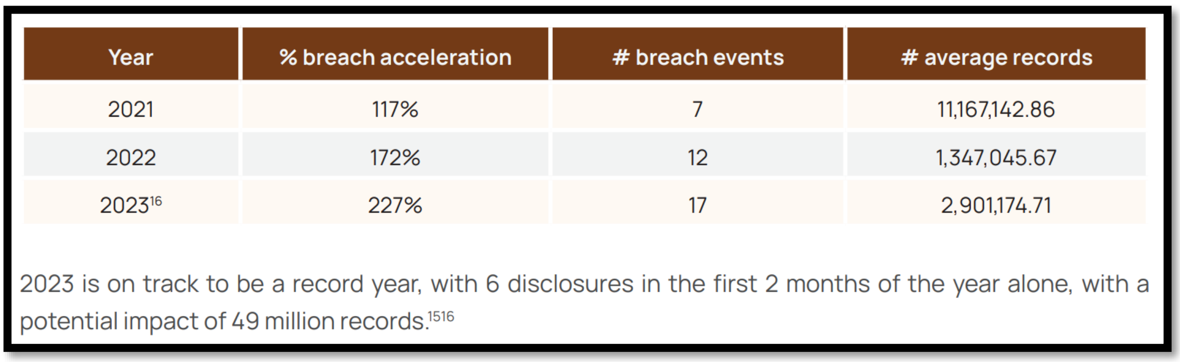


Figure 1: APIDataFort

1.2 Problem Scenario

APIs act as gateway points to the data and systems of an organization, increasing the area that attackers can target. Unauthorized access, data breaches, and even denial-of-service attacks can be easily initiated through unprotected or poorly built APIs and even giving criminals a blueprint to follow. Moreover, some of the issues that arise have been highlighted below.

- Forrester Research survey shows 78% of corporate decision-makers believe APIs are essential for competitiveness, customer interaction, and data ownership, with 60% having 25-250 internally released APIs and 49% having public APIs (Harrell & Stutzman, 2022).
- According to FireTail's API Security Report 2023, there has been a notable rise in the number of records exposed or compromised by APIs, with over 500 million records being affected.



Year	% breach acceleration	# breach events	# average records
2021	117%	7	11,167,142.86
2022	172%	12	1,347,045.67
2023[16]	227%	17	2,901,174.71

2023 is on track to be a record year, with 6 disclosures in the first 2 months of the year alone, with a potential impact of 49 million records.[15][16]

Figure 2: API data breach table by FireTail's API Security Report

- Authorization and authentication breaches are the most common, accounting for more than half of all data breaches (Snyder, et al., 2023). In addition, Appendix(13.1) addresses a few potential attack vectors and potential dangers.

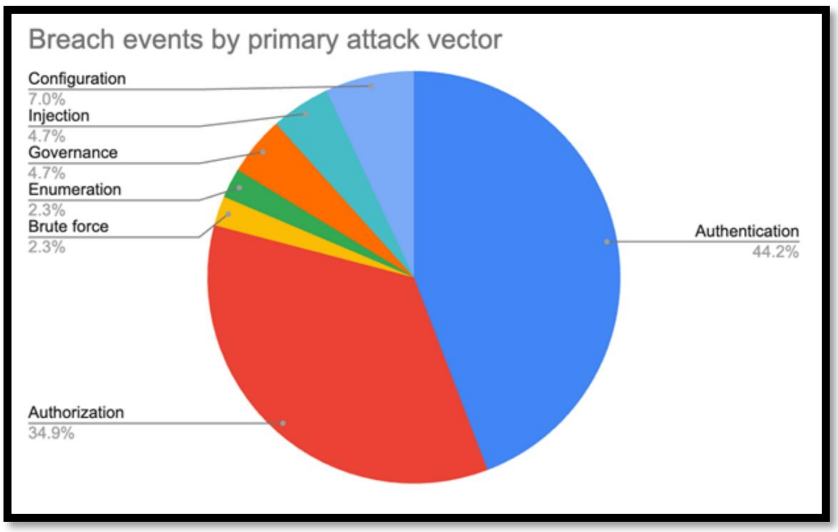
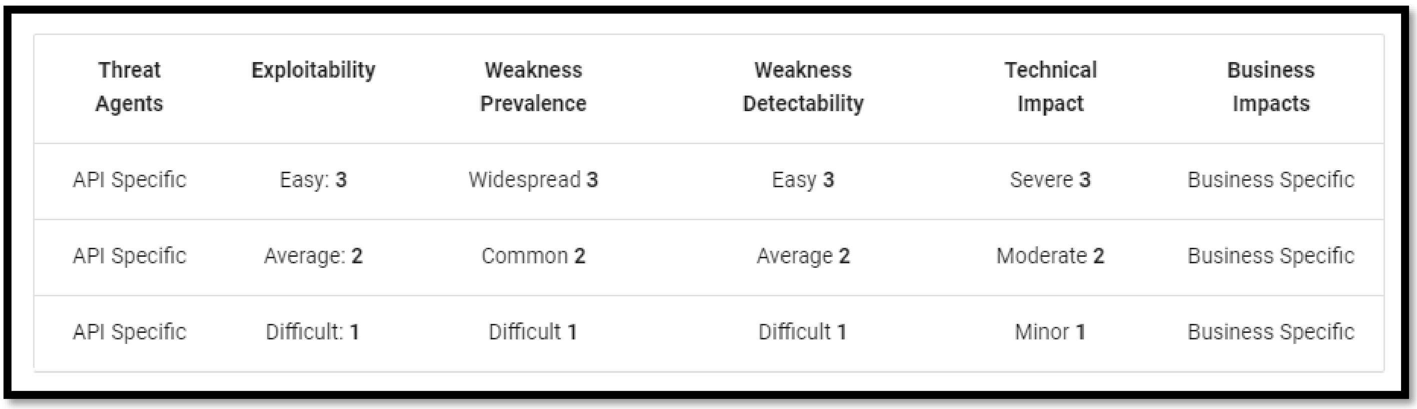


Figure 3: Prime Attack Prime Vector by FireTail's API Security Report

- In 2023, the average cost of a data breach with sensitive data hit an all-time high of USD 4.45 million, up 2% from 2022 and 15.3% from 2020. Organizations must put in security solutions and put in place strict rules to protect their data and lower the risk of breaches, given the troubling long-term trend of rising data breach costs (Hill, 2023).

- The risk score for an agent, comprising its threat, exploitability, vulnerability, prevalence, detectability, impact, business specificity, and API specificity, is presented in the OWASP API security Top 10 table. It describes the agent's susceptibility to a certain business or API, as well as its prevalence, detectability, potential impact, and specificity.



Threat Agents	Exploitability	Weakness Prevalence	Weakness Detectability	Technical Impact	Business Impacts
API Specific	Easy: **3**	Widespread **3**	Easy **3**	Severe **3**	Business Specific
API Specific	Average: **2**	Common **2**	Average **2**	Moderate **2**	Business Specific
API Specific	Difficult: **1**	Difficult **1**	Difficult **1**	Minor **1**	Business Specific

Figure 4: OWASP API security Top 10 risk score

1.3 The project as a solution

APIs are the new battleground for application security, and our digital lives depend on their security. APIDataFort, a solution designed to prevent unauthorized access, use, disclosure, and disruption of API data, addresses the two most prevalent attack vectors: authorization and authentication, in response to the growing demand for safeguarding API data among businesses.

APIDataFort supports the following features:

- Token management for authentication, authorization, and verification.
- ML/DL to detect and protect sensitive data.
- APIDataFort validates third-party APIs.
- Centralized dashboard for monitoring.

It could provide the following benefits with the use of the features:

- APIDataFort supports API security by addressing the two most major attack vectors in API data breaches: authorization and authentication.
- By encrypting sensitive data and certifying third-party APIs, APIDataFort helps to reduce the risk of data breaches.
- APIDataFort provides a centralized dashboard for administrators to administer and monitor the system, giving them greater visibility and control over their API security.
- By preventing data breaches and related financial penalties, harm to an organization's reputation, and loss of customer trust, it might help reduce expenses for businesses.

2. AIM AND OBJECTIVES

2.1 Aim

The primary aim of the project is to develop a solution that guards APIs from unauthorized access, protects sensitive data, and provides visibility over API security, thereby helping organizations improve their API security posture and reduce the risk of data breaches.

2.2 Objectives

The following are the objectives of the project to improve API security:

- Carry out in-depth research on APIs and security-related concerns.
- Utilize modern technologies such as ML/DL to reduce the number of cyberattacks due to APIs for the future by adapting to the changing technological landscape in which APIs operate.
- Carry out in-depth investigation in API and the issues with the API security.
- Create a system with strong authorization, authentication, and verification to reduce the API data breach attack vector.
- Provide a centralized administrator dashboard with necessary information and reporting.
- Deploy containerized systems using Docker, which improves scalability and manageability.
- Support the security of external APIs as well.
- Implement and evaluate an API data security system.
- Help businesses provide an affordable, reliable approach to protect their bottom lines.
- Create a user manual.
- Produce a technical document.

3. EXPECTED OUTCOMES AND DELIVERABLES

3.1 Expected Outcomes

All of the loosely sketched ideas are implemented in accordance with the project timeline, and following project completion, a system pertaining to API data security is expected to be anticipated, using various components. A system named APIDataFort is expected to be created which will have the following features:

- **Feature #1:** APIDataFort expects to offer robust authorization, verification, and authentication for APIs using OpenID Connect (OIDC) and JSON Web Tokens (JWTs) which guarantees that the data from the APIs may only be accessed by authorized users and that their identities are verified.
- **Feature #2:** APIDataFort expects to prevent sensitive data by identifying sensitive data in APIs using ML/DL, using a set of data that is trained to reliably recognize and encrypt sensitive data.
- **Feature #3:** APIDataFort expects to ensure the security of APIs by verifying the protocols (TLS/SSL) that lessen the chance of using vulnerabilities in third-party APIs to attack one's APIs.
- **Feature #4:** APIDataFort expects to offer administrators a centralized dashboard that is easy to use for managing and keeping an eye on the security of their APIs. This dashboard lets one create reports as needed and gives a high-level overview of the APIs' status.

3.2 Deliverables

After achieving the previously described outcomes, the project is expected to produce deliverables mentioned below designed for a targeted audience. Refer to Appendix(13.2) for further information on the targeted audience.

- **Deliverable 1: Functional Prototype**

  An APIDataFort functional prototype showcasing the following features:

  - Strong authentication, verification, and authorization for APIs utilizing JSON Web Tokens (JWTs) and OpenID Connect (OIDC)
  - Preventing sensitive data by employing ML/DL to detect sensitive data in APIs
  - Verification of the protocols (TLS/SSL) that reduce the possibility of an attack on one's APIs exploiting flaws in third-party APIs
  - A centralized, user-friendly dashboard for monitoring and controlling API security.

- **Deliverable 2: User Guides**

  An extensive APIDataFort user manual with steps for setting up, configuring, and utilizing the system.

- **Deliverable 3: Technical Documentation**

  A thorough technical specification comprising the following details for APIDataFort:

  - Architecture of the system
  - Descriptions of components
  - Models of data
  - APIDataFort reference guide
  - Instructions for deployment

4. PROJECT RISKS, THREATS, AND CONTINGENCY PLANS

Since all systems are inherently susceptible to risks and threats, there may be risks and threats during the project's development. The following risks and threats are addressed together with backup plans, to lessen systemic flaws.

4.1 Risks and Threats

- **ML/DL bias:** ML/DL models can be biased, leading to unfair and discriminatory outcomes.
- **Project management risks:** Limited time frame, budgeting, and execution can lead to delays, cost overruns, and project failure.
- **System testing risks:** If the system is not properly tested before deployment, it can cause problems after deployment.
- **Data loss risks:** Hardware failure, such as hard drive failure, or software failure can result in data loss.
- **Debugging risks:** Debugging can corrupt data if data structures are modified or accidentally deleted.

4.2 Contingency Plans

- **Mitigate ML/DL bias:** Train ML/DL models on high-quality, representative data that is representative of the population that the model will be used to make predictions about.
- **Reduce project management risks:** Use a project management methodology to help plan, budget, and execute the project effectively. Identify and track key milestones and deliverables.
- **Minimize system testing risks:** Conduct thorough testing of the system, including both functional and security testing. Deploy the system in a phased approach to minimize the risk of disruption.
- **Reduce data loss risks:** Use multiple backup copies and store backup media in a secure location. Consider using a cloud backup service for scalability and ease of use.
- **Reduce debugging risks:** Use a sandbox to isolate the code being debugged and prevent changes from affecting the rest of the code.

5. METHODOLOGY

5.1 Considered Methodology

Embarking on the project, a wide range of methodologies were explored and considered, weighing each one's distinct advantages and matching them to project objectives, which are outlined in the appendix below. To delve into this exploration, please refer to Appendix(13.3).

5.2 Selected Methodology

Among the approaches that were taken into consideration, the Agile Methodology was chosen for the project. The agile methodology is an iterative methodology for project management that divides work into many flexible stages known as sprints. Agile values team collaboration, working software, customer collaboration and adaptability (Laoyan, 2022). Please refer to Appendix(13.4) for more information on Agile methodology.

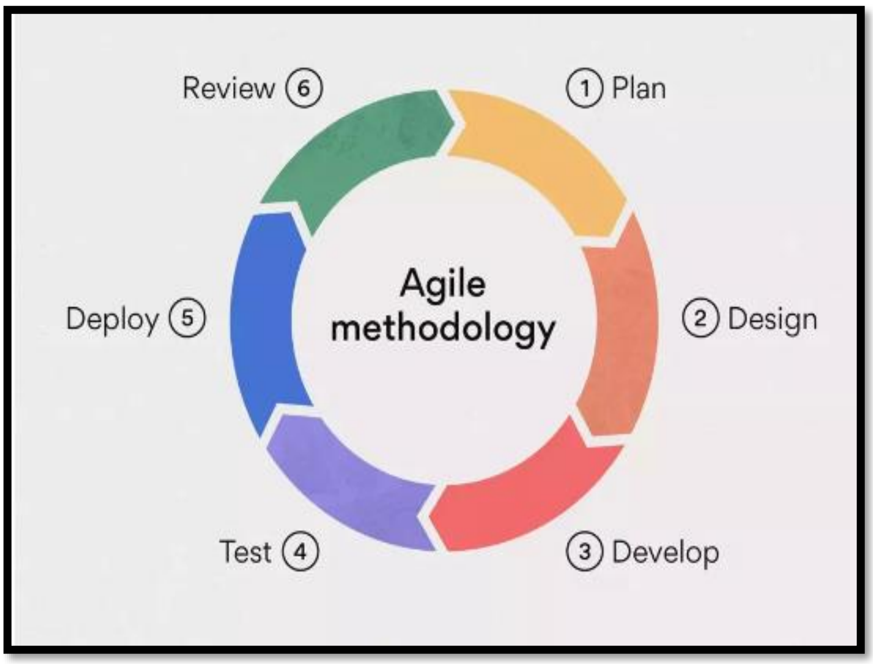


Figure 5: Agile Methodology (Laoyan, 2022)

Scrum:

Scrum is a management methodology used by software teams to self-organize, collaborate, and learn from experience, enabling efficient and sustainable problem-solving in complex projects. Scrum is a methodology that focuses on delivering client value within a Sprint by a self-organizing team, with each Sprint's associated roles, artefacts, and events defined.

Artifacts

- **Product Backlog:** Prioritized list of features, requirements, enhancements, and fixes.
- **Sprint Backlog:** List of items committed to completing in the current Sprint.
- **Increment:** Usable end product from a Sprint.

Roles

- **Product Owner:** Prioritizes needs of customers.
- **Scrum Master:** Facilitates Scrum events and removes impediments.
- **Development Team:** Designs, develops, and tests the product.

Events

- **Sprint Planning:** Select items from the Product Backlog for the Sprint.
- **Daily Scrum:** Reports progress, identifies blockers, and plans work.

- **Sprint Review:** Demonstrates completed work and gathers feedback.
- **Sprint Retrospective:** Reflects on the Sprint, identifies improvements, and makes plans (Amazon Web Services, 2023).

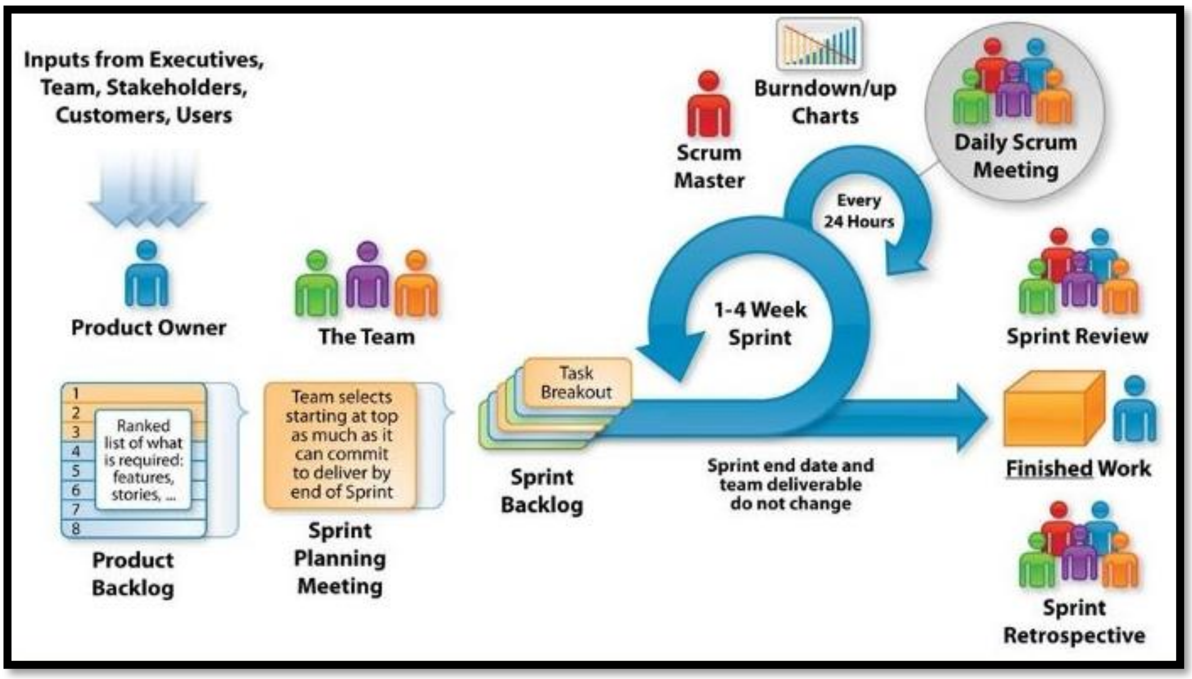

Inputs from Executives, Team, Stakeholders, Customers, Users
Product Owner
The Team
Scrum Master
Burndown/up Charts
Daily Scrum Meeting
Every 24 Hours
1-4 Week Sprint
Sprint Review
Task Breakout
1
2
3
4
5
6
7
8
Ranked list of what is required: features, stories, ...
Product Backlog
Team selects starting at top as much as it can commit to deliver by end of Sprint
Sprint Planning Meeting
Sprint Backlog
Sprint end date and team deliverable do not change
Finished Work
Sprint Retrospective

Figure 6: Scrum Methodology (Neon Rain Interactive, LLC, 2023)

Phases of the scrum methodology :

Scrum breaks down the development process into distinct phases, each of which plays a critical role in accomplishing project goals, which are outlined below.

- **Initiation Phase**: All the project's goals and objectives, as well as defining the project's vision, identifying organizations or groups with an interest in the project's outcome, creating a prioritised list of product backlog items, representing the desired features, and estimating the total effort required for the entire project, would be completed during this phase.
- **Planning & Estimates Phase:** The project will be divided into manageable sprints of 1-4 weeks, each reflecting a short-term development cycle. Tasks from the prioritized backlog will be selected for each sprint, with tasks assigned an estimated effort value for realistic completion time and resources.
- **Implementation Phase:** The Implementation Phase will be used to transform plans into reality by completing tasks in the sprint backlog, integrating code regularly, holding daily scrums for communication, and implementing plans as needed. Automated tests will be encouraged to ensure quality, detect flaws early, and prevent errors, ensuring high-quality results that meet stakeholder expectations.
- **Review & Retrospective Phase:** The phase will involve showcasing the product increment to stakeholders, analyzing their feedback on usability and quality, brainstorming for improvement, and recognizing successful tasks from the sprint, while also identifying areas for enhancement or feature requests.
- **Release Phase:** The final phase will involve deploying the product increment, testing and UAT to confirm functionality and compatibility, and monitoring the system to identify and address potential issues or flaws.

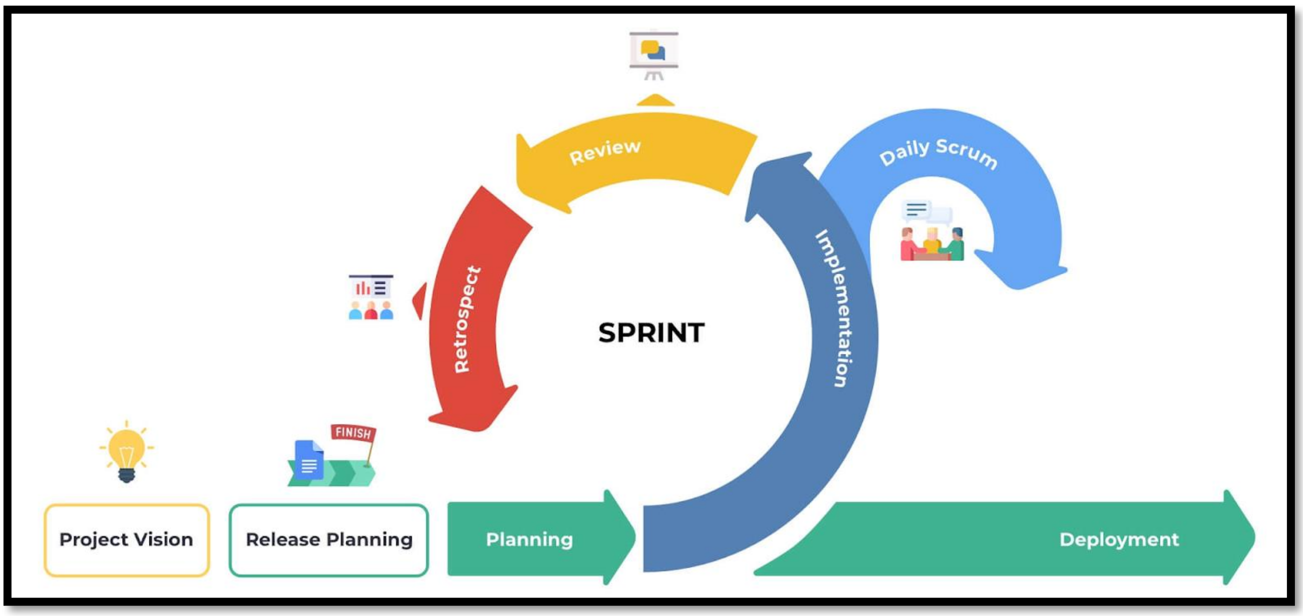


Figure 7: Phases of Scrum (Donato, 2023)

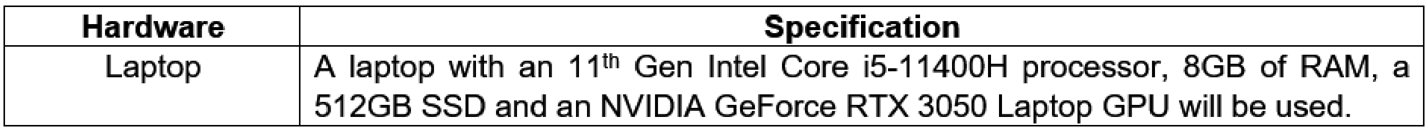
Please refer to Appendix (13.5) for the justification for using the scrum methodology.

6. RESOURCE REQUIREMENTS

To start up the project and create the system there are components that will be used which are as follows:

6.1 Hardware

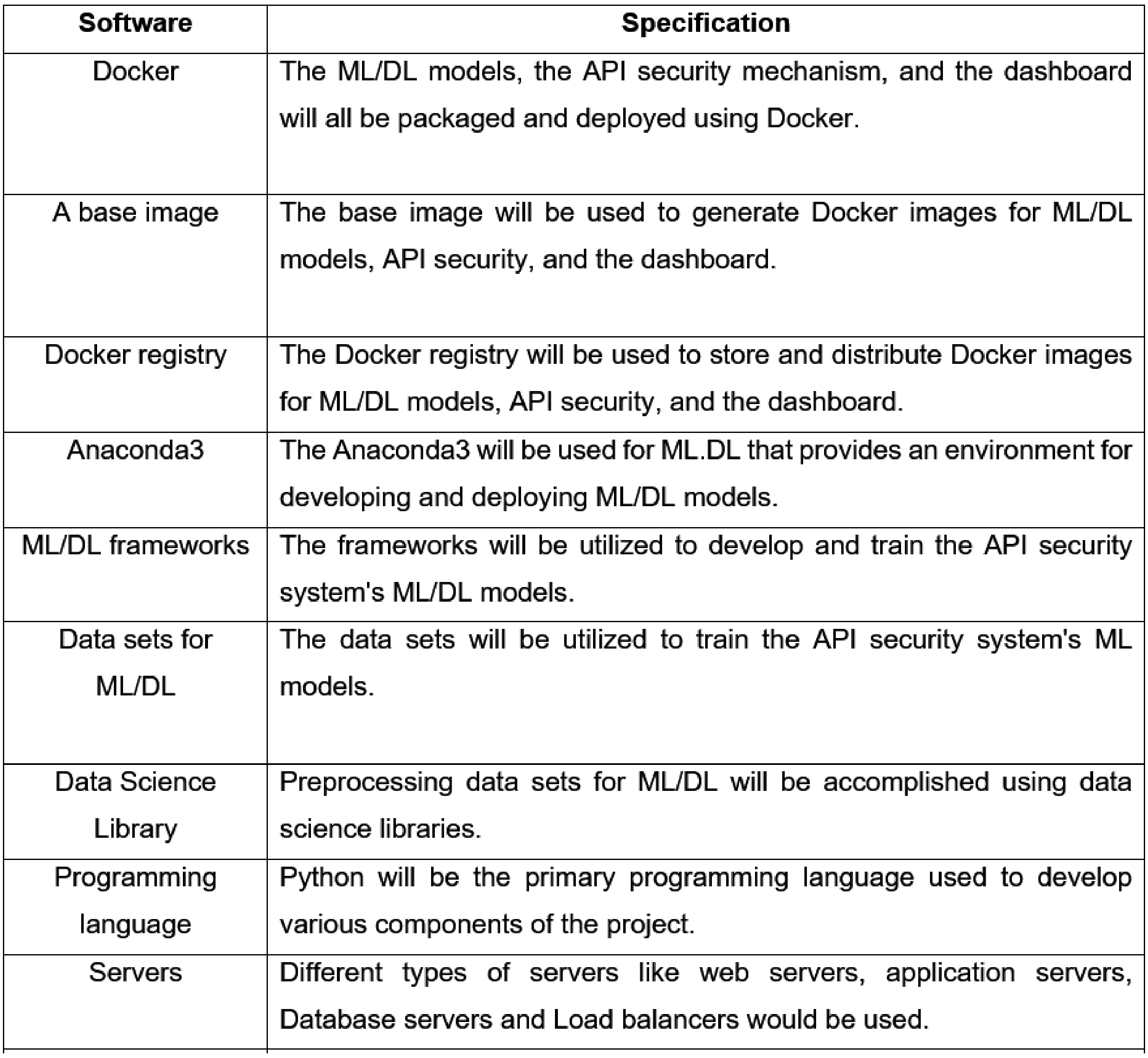
Table 1: List of Hardware Requirements



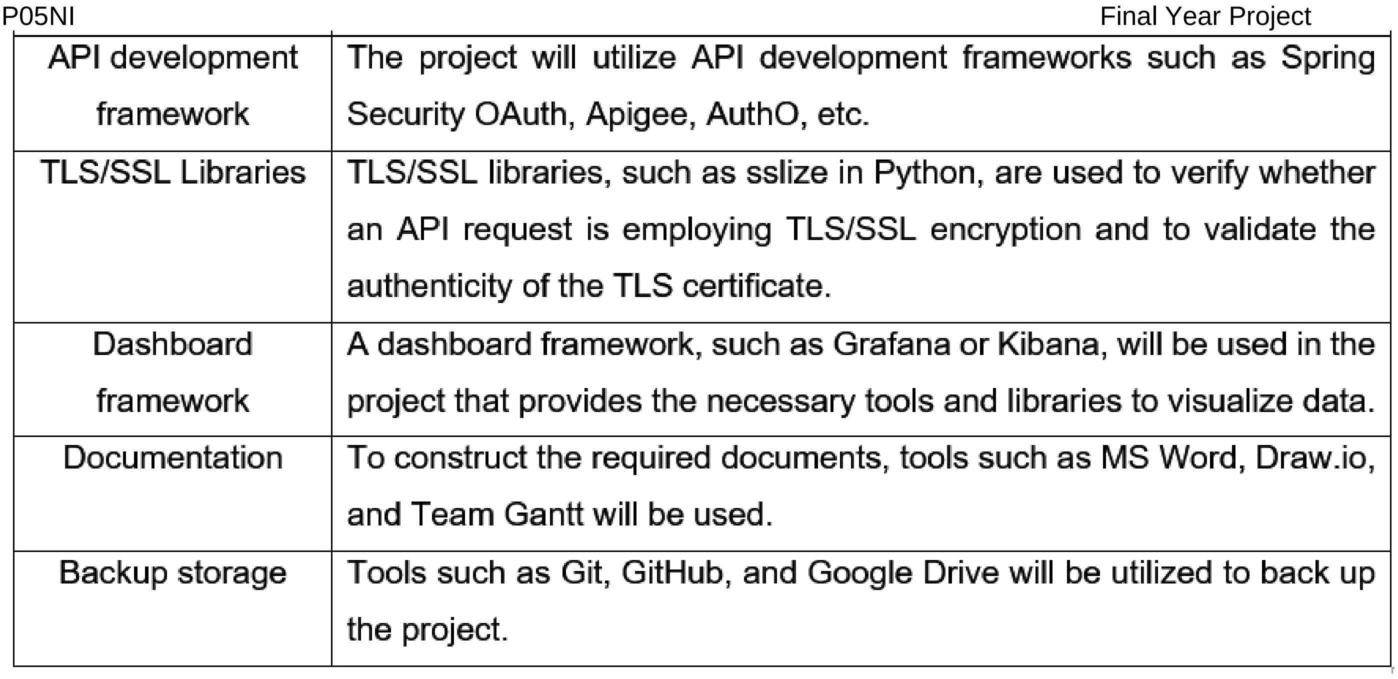
Hardware	Specification
Laptop	A laptop with an 11th Gen Intel Core i5-11400H processor, 8GB of RAM, a 512GB SSD and an NVIDIA GeForce RTX 3050 Laptop GPU will be used.

6.2 Software

Table 2: List of Software Requirements



Software	Specification
Docker	The ML/DL models, the API security mechanism, and the dashboard will all be packaged and deployed using Docker.
A base image	The base image will be used to generate Docker images for ML/DL models, API security, and the dashboard.
Docker registry	The Docker registry will be used to store and distribute Docker images for ML/DL models, API security, and the dashboard.
Anaconda3	The Anaconda3 will be used for ML.DL that provides an environment for developing and deploying ML/DL models.
ML/DL frameworks	The frameworks will be utilized to develop and train the API security system's ML/DL models.
Data sets for ML/DL	The data sets will be utilized to train the API security system's ML models.
Data Science Library	Preprocessing data sets for ML/DL will be accomplished using data science libraries.
Programming language	Python will be the primary programming language used to develop various components of the project.
Servers	Different types of servers like web servers, application servers, Database servers and Load balancers would be used.



API development framework	The project will utilize API development frameworks such as Spring Security OAuth, Apigee, AuthO, etc.
TLS/SSL Libraries	TLS/SSL libraries, such as sslize in Python, are used to verify whether an API request is employing TLS/SSL encryption and to validate the authenticity of the TLS certificate.
Dashboard framework	A dashboard framework, such as Grafana or Kibana, will be used in the project that provides the necessary tools and libraries to visualize data.
Documentation	To construct the required documents, tools such as MS Word, Draw.io, and Team Gantt will be used.
Backup storage	Tools such as Git, GitHub, and Google Drive will be utilized to back up the project.

7. WORK BREAKDOWN STRUCTURE

Work Breakdown Structure (WBS), which is illustrated below, is used to break down a project into smaller, more manageable components to ensure that all the work required to accomplish the project is identified and defined. Please refer to Appendix(13.6) for a clearer picture.

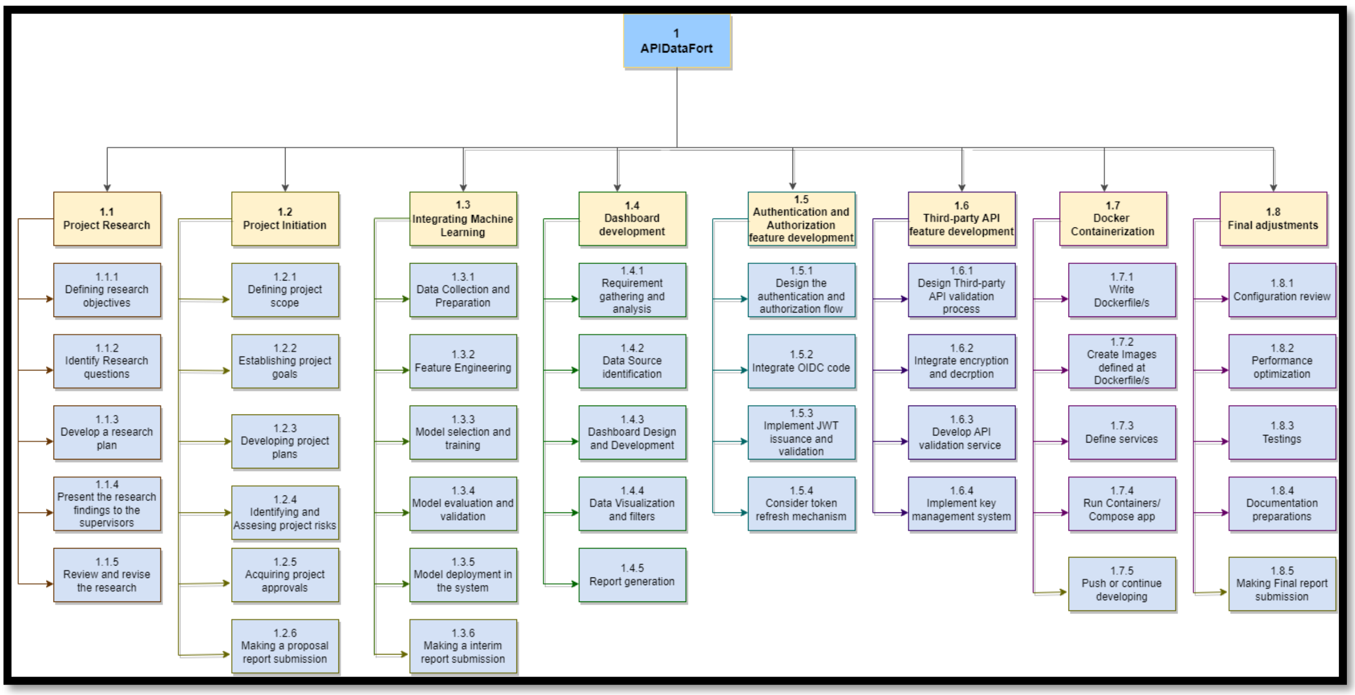

1 APIDataFort
1.1 Project Research
1.1.1 Defining research objectives
1.1.2 Identify Research questions
1.1.3 Develop a research plan
1.1.4 Present the research findings to the supervisors
1.1.5 Review and revise the research
1.2 Project Initiation
1.2.1 Defining project scope
1.2.2 Establishing project goals
1.2.3 Developing project plans
1.2.4 Identifying and Assesing project risks
1.2.5 Acquiring project approvals
1.2.6 Making a proposal report submission
1.3 Integrating Machine Learning
1.3.1 Data Collection and Preparation
1.3.2 Feature Engineering
1.3.3 Model selection and training
1.3.4 Model evaluation and validation
1.3.5 Model deployment in the system
1.3.6 Making a interim report submission
1.4 Dashboard development
1.4.1 Requirement gathering and analysis
1.4.2 Data Source identification
1.4.3 Dashboard Design and Development
1.4.4 Data Visualization and filters
1.4.5 Report generation
1.5 Authentication and Authorization feature development
1.5.1 Design the authentication and authorization flow
1.5.2 Integrate OIDC code
1.5.3 Implement JWT issuance and validation
1.5.4 Consider token refresh mechanism
1.6 Third-party API feature development
1.6.1 Design Third-party API validation process
1.6.2 Integrate encryption and decrption
1.6.3 Develop API validation service
1.6.4 Implement key management system
1.7 Docker Containerization
1.7.1 Write Dockerfile/s
1.7.2 Create Images defined at Dockerfile/s
1.7.3 Define services
1.7.4 Run Containers/ Compose app
1.7.5 Push or continue developing
1.8 Final adjustments
1.8.1 Configuration review
1.8.2 Performance optimization
1.8.3 Testings
1.8.4 Documentation preparations
1.8.5 Making Final report submission

Figure 8: Project Work Breakdown Structure (WBS)

8. MILESTONES

Some project milestones are brought out below to steer and sustain motivation. This is like putting signposts along a long road trip to give clear directions. Please refer to Appendix(13.6) for a clearer picture.

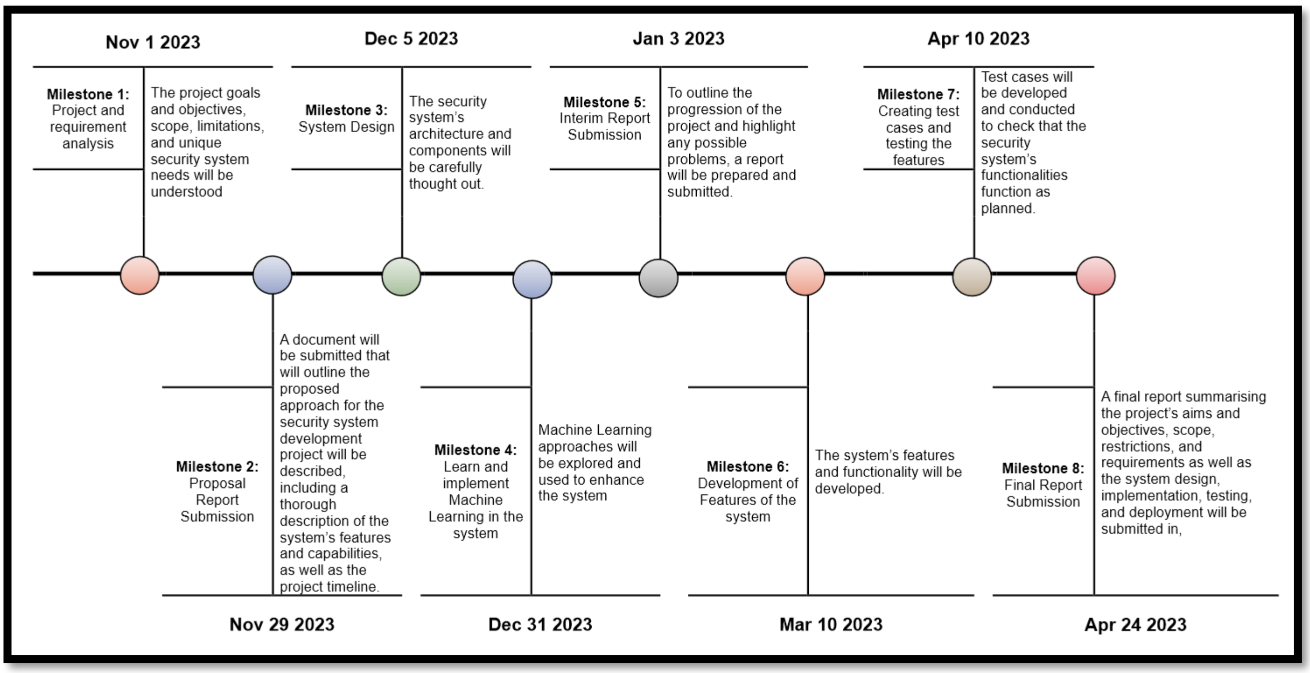

Nov 1 2023
Milestone 1: Project and requirement analysis
The project goals and objectives, scope, limitations, and unique security system needs will be understood
Dec 5 2023
Milestone 3: System Design
The security system's architecture and components will be carefully thought out.
Jan 3 2023
Milestone 5: Interim Report Submission
To outline the progression of the project and highlight any possible problems, a report will be prepared and submitted.
Apr 10 2023
Milestone 7: Creating test cases and testing the features
Test cases will be developed and conducted to check that the security system's functionalities function as planned.
Milestone 2: Proposal Report Submission
A document will be submitted that will outline the proposed approach for the security system development project will be described, including a thorough description of the system's features and capabilities, as well as the project timeline.
Nov 29 2023
Milestone 4: Learn and implement Machine Learning in the system
Machine Learning approaches will be explored and used to enhance the system
Dec 31 2023
Milestone 6: Development of Features of the system
The system's features and functionality will be developed.
Mar 10 2023
Milestone 8: Final Report Submission
A final report summarising the project's aims and objectives, scope, restrictions, and requirements as well as the system design, implementation, testing, and deployment will be submitted in,
Apr 24 2023

Figure 9: Project Milestones

9. PROJECT GANTT CHART

This project, as previously stated, will use the scrum methodology, an agile framework that breaks projects into sprints to allow for flexibility and adaptation because the project can be modified in response to feedback and new data. To properly allocate resources and ensure that all tasks are finished on time, a Gantt chart is created, as shown below, to illustrate the project's timetable. Please refer to Appendix(13.6) for a clearer picture.

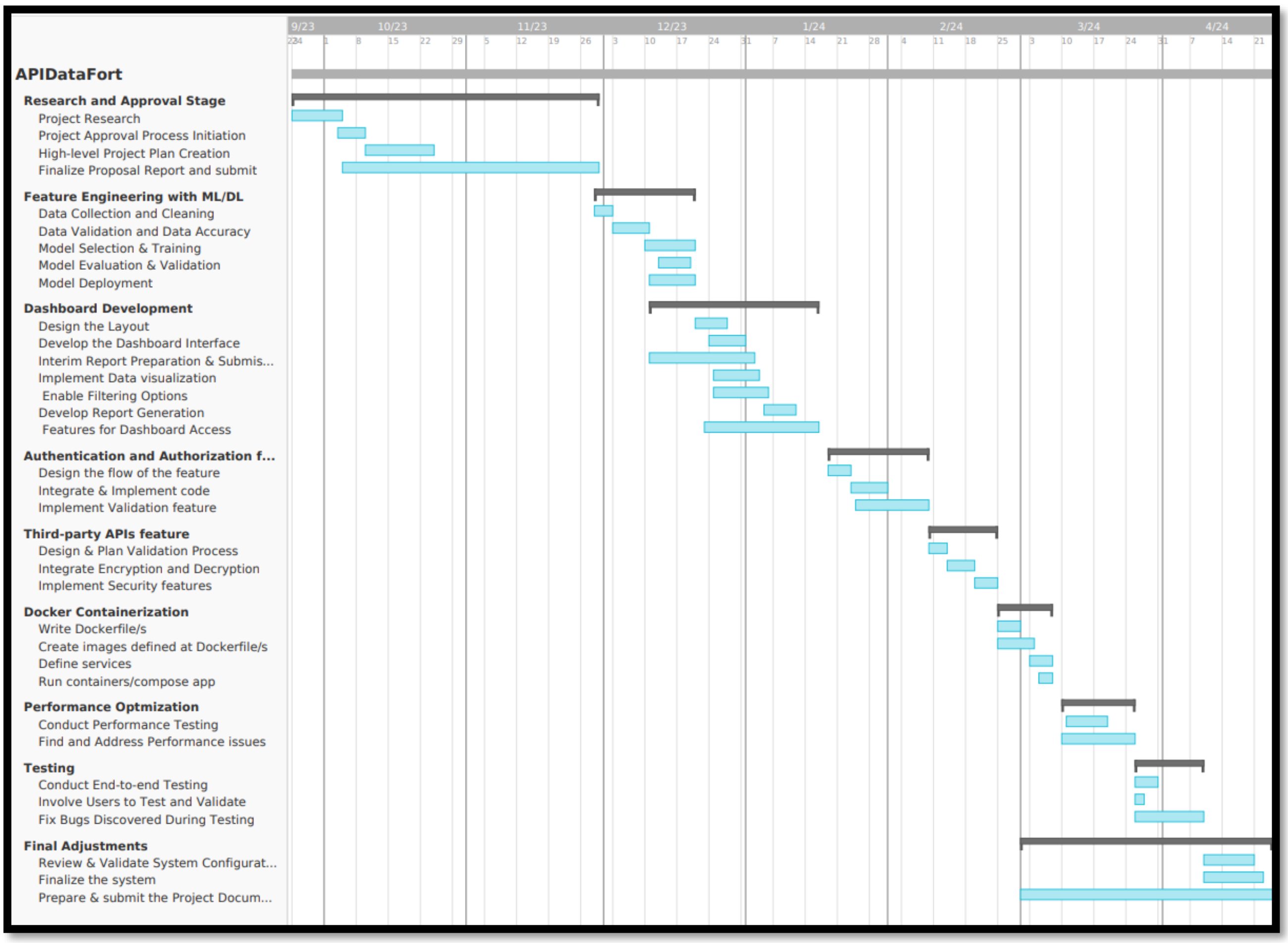
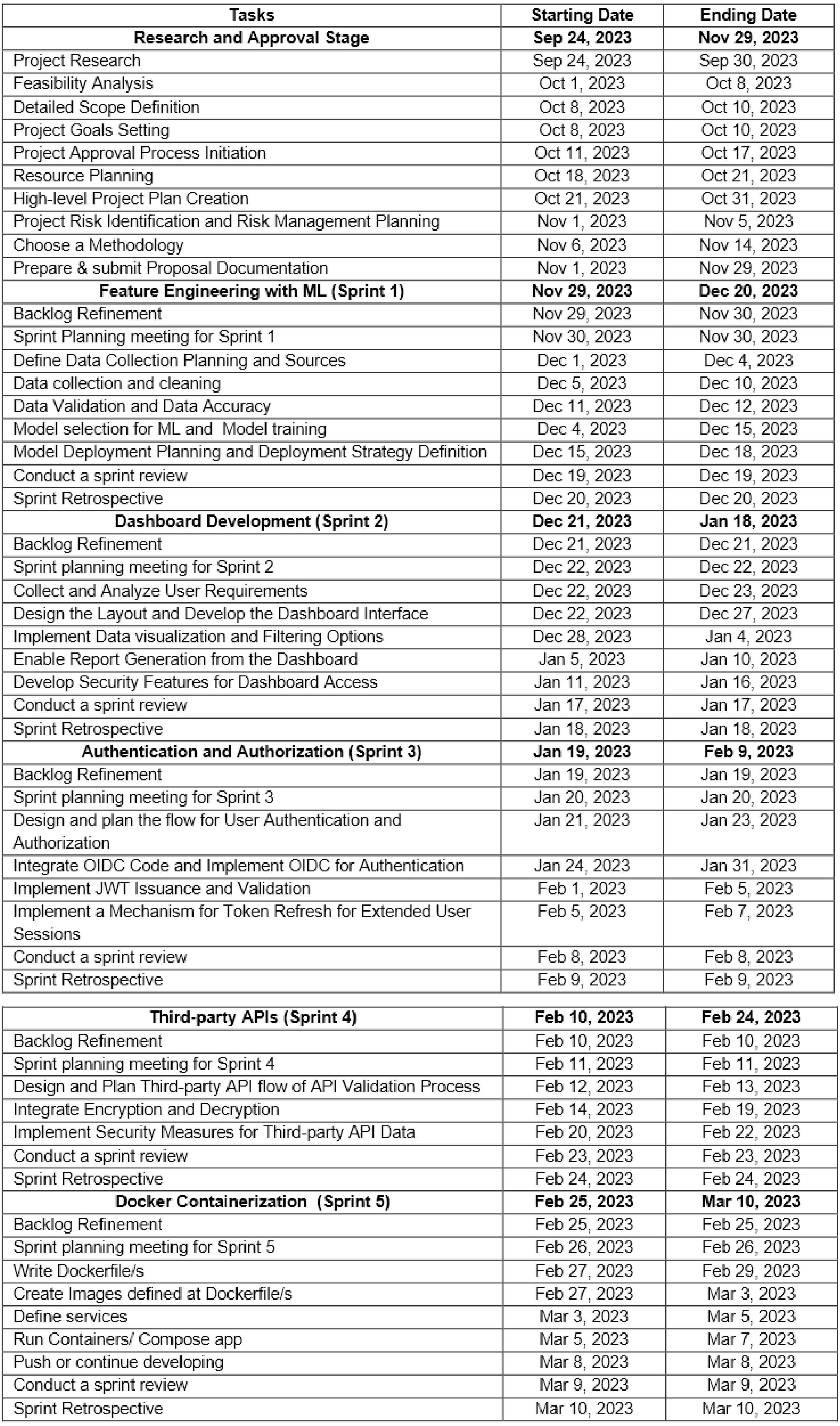


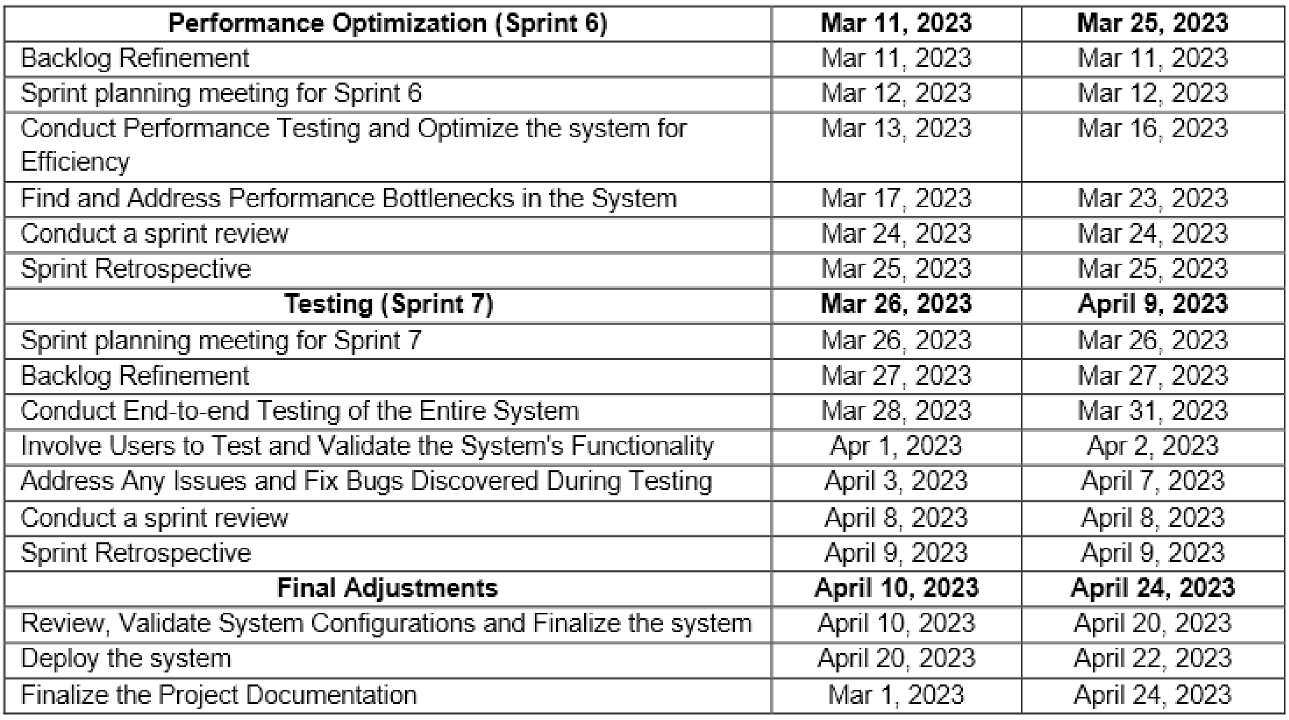
Figure 10: Project Gantt Chart

The following table contains an expanded anticipated schedule for the project.

Table 3: Gantt Chart in more detail



Tasks	Starting Date	Ending Date
Research and Approval Stage	**Sep 24, 2023**	**Nov 29, 2023**
Project Research	Sep 24, 2023	Sep 30, 2023
Feasibility Analysis	Oct 1, 2023	Oct 8, 2023
Detailed Scope Definition	Oct 8, 2023	Oct 10, 2023
Project Goals Setting	Oct 8, 2023	Oct 10, 2023
Project Approval Process Initiation	Oct 11, 2023	Oct 17, 2023
Resource Planning	Oct 18, 2023	Oct 21, 2023
High-level Project Plan Creation	Oct 21, 2023	Oct 31, 2023
Project Risk Identification and Risk Management Planning	Nov 1, 2023	Nov 5, 2023
Choose a Methodology	Nov 6, 2023	Nov 14, 2023
Prepare & submit Proposal Documentation	Nov 1, 2023	Nov 29, 2023
Feature Engineering with ML (Sprint 1)	**Nov 29, 2023**	**Dec 20, 2023**
Backlog Refinement	Nov 29, 2023	Nov 30, 2023
Sprint Planning meeting for Sprint 1	Nov 30, 2023	Nov 30, 2023
Define Data Collection Planning and Sources	Dec 1, 2023	Dec 4, 2023
Data collection and cleaning	Dec 5, 2023	Dec 10, 2023
Data Validation and Data Accuracy	Dec 11, 2023	Dec 12, 2023
Model selection for ML and Model training	Dec 4, 2023	Dec 15, 2023
Model Deployment Planning and Deployment Strategy Definition	Dec 15, 2023	Dec 18, 2023
Conduct a sprint review	Dec 19, 2023	Dec 19, 2023
Sprint Retrospective	Dec 20, 2023	Dec 20, 2023
Dashboard Development (Sprint 2)	**Dec 21, 2023**	**Jan 18, 2023**
Backlog Refinement	Dec 21, 2023	Dec 21, 2023
Sprint planning meeting for Sprint 2	Dec 22, 2023	Dec 22, 2023
Collect and Analyze User Requirements	Dec 22, 2023	Dec 23, 2023
Design the Layout and Develop the Dashboard Interface	Dec 22, 2023	Dec 27, 2023
Implement Data visualization and Filtering Options	Dec 28, 2023	Jan 4, 2023
Enable Report Generation from the Dashboard	Jan 5, 2023	Jan 10, 2023
Develop Security Features for Dashboard Access	Jan 11, 2023	Jan 16, 2023
Conduct a sprint review	Jan 17, 2023	Jan 17, 2023
Sprint Retrospective	Jan 18, 2023	Jan 18, 2023
Authentication and Authorization (Sprint 3)	**Jan 19, 2023**	**Feb 9, 2023**
Backlog Refinement	Jan 19, 2023	Jan 19, 2023
Sprint planning meeting for Sprint 3	Jan 20, 2023	Jan 20, 2023
Design and plan the flow for User Authentication and Authorization	Jan 21, 2023	Jan 23, 2023
Integrate OIDC Code and Implement OIDC for Authentication	Jan 24, 2023	Jan 31, 2023
Implement JWT Issuance and Validation	Feb 1, 2023	Feb 5, 2023
Implement a Mechanism for Token Refresh for Extended User Sessions	Feb 5, 2023	Feb 7, 2023
Conduct a sprint review	Feb 8, 2023	Feb 8, 2023
Sprint Retrospective	Feb 9, 2023	Feb 9, 2023
Third-party APIs (Sprint 4)	**Feb 10, 2023**	**Feb 24, 2023**
Backlog Refinement	Feb 10, 2023	Feb 10, 2023
Sprint planning meeting for Sprint 4	Feb 11, 2023	Feb 11, 2023
Design and Plan Third-party API flow of API Validation Process	Feb 12, 2023	Feb 13, 2023
Integrate Encryption and Decryption	Feb 14, 2023	Feb 19, 2023
Implement Security Measures for Third-party API Data	Feb 20, 2023	Feb 22, 2023
Conduct a sprint review	Feb 23, 2023	Feb 23, 2023
Sprint Retrospective	Feb 24, 2023	Feb 24, 2023
Docker Containerization (Sprint 5)	**Feb 25, 2023**	**Mar 10, 2023**
Backlog Refinement	Feb 25, 2023	Feb 25, 2023
Sprint planning meeting for Sprint 5	Feb 26, 2023	Feb 26, 2023
Write Dockerfile/s	Feb 27, 2023	Feb 29, 2023
Create Images defined at Dockerfile/s	Feb 27, 2023	Mar 3, 2023
Define services	Mar 3, 2023	Mar 5, 2023
Run Containers/ Compose app	Mar 5, 2023	Mar 7, 2023
Push or continue developing	Mar 8, 2023	Mar 8, 2023
Conduct a sprint review	Mar 9, 2023	Mar 9, 2023
Sprint Retrospective	Mar 10, 2023	Mar 10, 2023



Performance Optimization (Sprint 6)	**Mar 11, 2023**	**Mar 25, 2023**
Backlog Refinement	Mar 11, 2023	Mar 11, 2023
Sprint planning meeting for Sprint 6	Mar 12, 2023	Mar 12, 2023
Conduct Performance Testing and Optimize the system for Efficiency	Mar 13, 2023	Mar 16, 2023
Find and Address Performance Bottlenecks in the System	Mar 17, 2023	Mar 23, 2023
Conduct a sprint review	Mar 24, 2023	Mar 24, 2023
Sprint Retrospective	Mar 25, 2023	Mar 25, 2023
Testing (Sprint 7)	**Mar 26, 2023**	**April 9, 2023**
Sprint planning meeting for Sprint 7	Mar 26, 2023	Mar 26, 2023
Backlog Refinement	Mar 27, 2023	Mar 27, 2023
Conduct End-to-end Testing of the Entire System	Mar 28, 2023	Mar 31, 2023
Involve Users to Test and Validate the System's Functionality	Apr 1, 2023	Apr 2, 2023
Address Any Issues and Fix Bugs Discovered During Testing	April 3, 2023	April 7, 2023
Conduct a sprint review	April 8, 2023	April 8, 2023
Sprint Retrospective	April 9, 2023	April 9, 2023
Final Adjustments	**April 10, 2023**	**April 24, 2023**
Review, Validate System Configurations and Finalize the system	April 10, 2023	April 20, 2023
Deploy the system	April 20, 2023	April 22, 2023
Finalize the Project Documentation	Mar 1, 2023	April 24, 2023

10. CONCLUSION

Moreover, this project provides a comprehensive API security solution that employs revolutionary security approaches to protect against unauthorized access and data breaches. The system provides effective planning, budgeting, and implementation by employing the agile Scrum method of development. Furthermore, it enables real-time visibility into API usage, allowing for the early detection and resolution of possible vulnerabilities in security. This complete approach significantly boosts an organization's security posture and secures sensitive data.

11. References

Adam, J., 2021. *What is the Waterfall software development methodology and is it still relevant?.* [Online]
Available at:
https://www.google.com/url?sa=i&url=https%3A%2F%2Fkruschecompany.com%2Fwaterfall-software-development-methodology%2F&psig=AOvVaw2Op2teLrBIZQS53T7520I8&ust=1700804017823000&source=images&cd=vfe&opi=89978449&ved=0CBIQjRxqFwoTCNDDtoOz2YIDFQAAAAAdAAAAABAD
[Accessed 13 11 2023].

Adam, J., 2023. *The Kanban system for agile software development explained.* [Online]
Available at:
https://www.google.com/url?sa=i&url=https%3A%2F%2Fkruschecompany.com%2Fkanban-method-agile-software-development%2F&psig=AOvVaw2sFfc3Mgul3I7l32yC1m8j&ust=1700804488070000&source=images&cd=vfe&opi=89978449&ved=0CBIQjRxqFwoTCKi0yOO02YIDFQAAAAAdAAAAABAD
[Accessed 16 11 2023].

altexsoft, 2021. *Extreme Programming: Values, Principles, and Practices.* [Online]
Available at:
https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.altexsoft.com%2Fblog%2Fextreme-programming-values-principles-and-practices%2F&psig=AOvVaw1uQHtj7ARlsyrO4hRwmsCy&ust=1700804564185000&source=images&cd=vfe&opi=89978449&ved=0CBIQjRxqFwoTCPCq-4i12YIDFQAAA
[Accessed 16 11 2023].

Amazon Web Services, 2023. *What is Scrum?.* [Online]
Available at: https://aws.amazon.com/what-is/scrum/#:~:text=Scrum%20is%20a%20management%20framework,experience%2C%20and%20adapt%20to%20change.
[Accessed 3 11 2023].

APIsec, 2022. *Why APIs are Your Biggest Security Risk.* [Online]
Available at: https://www.apisec.ai/blog/why-apis-are-your-biggest-security-risk
[Accessed 20 10 2023].

Atlassian, 2023. *Kanban How the kanban methodology applies to software development.* [Online]
Available at: https://www.atlassian.com/agile/kanban
[Accessed 2 11 2023].

CodeStringers, 2021. *The four values of the Agile Manifesto.* [Online]
Available at:
https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.codestringers.com%2Fthe-four-values-of-the-agile-manifesto%2F&psig=AOvVaw2nQpoBdmJWvUxFBuM1Ss3y&ust=1700803688363000&source=images&cd=vfe&ved=0CBIQjRxqFwoTCLj-i-Wx2YIDFQAAAAAdAAAAABAE
[Accessed 14 11 2023].

Donato, H., 2023. *What Are The Phases Of Scrum?.* [Online]
Available at:
https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.workamajig.com%2Fblog%2Fscrum-methodology-guide%2Fscrum-phases&psig=AOvVaw0T-_D5aUbg3EYvpJ7zQfK5&ust=1701182798920000&source=images&cd=vfe&opi=89978449&ved=0CBIQjRxqFwoTCOCUyI625IIDFQAAAAAdAAAAABAI
[Accessed 15 11 2023].

Gatlan, S., 2022. *Dropbox discloses breach after hacker stole 130 GitHub repositories.* [Online]
Available at: https://www.bleepingcomputer.com/news/security/dropbox-discloses-breach-after-hacker-stole-130-github-repositories/
[Accessed 11 09 2023].

GeeksforGeeks, 2023. *Software Engineering* | *Spiral Model.* [Online]
Available at: https://www.geeksforgeeks.org/software-engineering-spiral-model/
[Accessed 05 11 2023].

GeeksforGeeks, 2023. *Software Engineering* | *Spiral Model.* [Online]
Available at: https://www.geeksforgeeks.org/software-engineering-spiral-model/
[Accessed 14 11 2023].

Harrell, M. & Stutzman, E., 2022. *Improve API Performance With A Sound API Security Strategy,* s.l.: Forrester.

Hill, M., 2023. *What is the cost of a data breach?.* [Online]
Available at: https://www.csoonline.com/article/567697/what-is-the-cost-of-a-data-breach-3.html
[Accessed 26 10 2023].

Intellipaat Software Solutions , 2023. *What is Agile: An Agile Methodology Guide.* [Online]
Available at:
https://www.google.com/url?sa=i&url=https%3A%2F%2Fintellipaat.com%2Fblog%2Fwhat-is-agile%2F&psig=AOvVaw2-3zQGYBeBshorYcUW5cuc&ust=1700803762603000&source=images&cd=vfe&opi=89978449&ved=0CBIQjRxqFwoTCKCelYmy2YIDFQAAAAAdAAAAABAD
[Accessed 14 11 2023].

Landi, H., 2019. *Quest Diagnostics breach may have exposed data of 11.9M patients.* [Online]
Available at: https://www.fiercehealthcare.com/tech/quest-diagnostics-breach-may-have-exposed-data-11-9m-patients
[Accessed 09 09 2023].

Laoyan, S., 2022. *What is Agile methodology? (A beginner’s guide).* [Online]
Available at: https://asana.com/resources/agile-methodology
[Accessed 2 11 2023].

Laoyan, S., 2022. *What is Agile methodology? (A beginner’s guide).* [Online]
Available at:
https://www.google.com/url?sa=i&url=https%3A%2F%2Fasana.com%2Fresources%2Fagile-methodology&psig=AOvVaw1n4zYSYELR9kbqhf5q2NDV&ust=1700803623499000&source=images&cd=vfe&ved=0CBIQjRxqFwoTCIiWk8ax2YIDFQAAAAAdAAAAABAE
[Accessed 14 11 2023].

Lewis, S., 2023. *prototyping model.* [Online]
Available at: https://www.techtarget.com/searchcio/definition/Prototyping-Model
[Accessed 05 11 2023].

Martin, M., 2023. *Prototype Model in Software Engineering.* [Online]
Available at: https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.guru99.com%2Fsoftware-engineering-prototyping-model.html&psig=AOvVaw1uge4U7CZm5musjQtnS93_&ust=1700804090144000&source=images&cd=vfe&opi=89978449&ved=0CBIQjRxqFwoTCKCR46Wz2YIDFQAAAAAdAAAAABAD
[Accessed 14 11 2023].

Naimisha, 2023. *The Biggest Data Breach in Australian History: An awakening for Organizations.* [Online]
Available at: https://securityboulevard.com/2023/07/the-biggest-data-breach-in-australian-history-an-awakening-for-organizations/#:~:text=The%20biggest%20data%20breach%20in%202023%20so%20far%20was%20at,driver's%20licenses%2C%20and%20passport%20numbers.
[Accessed 10 09 2023].

Neon Rain Interactive, LLC, 2023. *Agile Scrum for Web Development.* [Online]
Available at:
https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.neonrain.com%2Fblog%2Fagile-scrum-for-web-development%2F&psig=AOvVaw31pL2vlZ1PVSEtYMmdKaoI&ust=1700803839670000&source=images&cd=vfe&opi=89978449&ved=0CBIQjRxqFwoTCIiI9MCy2YIDFQAAAAAdAAAAABAJ
[Accessed 15 2023 2023].

OWASP API Security Project team , 2023. *API Security Risks.* [Online]
Available at: https://owasp.org/API-Security/editions/2023/en/0x10-api-security-risks/
[Accessed 25 10 2023].

ProjectManager.com, 2023. *What Is the Waterfall Methodology in Project Management?.* [Online]
Available at: https://www.projectmanager.com/guides/waterfall-methodology
[Accessed 04 11 2023].

Snyder, J., Priddle, R. & Foster, I., 2023. *The State of APIs and API security in 2023,* s.l.: FireTail.

Venema, M., 2023. *What Is Extreme Programming (XP)?.* [Online]
Available at: https://www.nimblework.com/agile/extreme-programming-xp/
[Accessed 2 11 2023].

12. Bibliography

Chandana, 2023. *Scrum Project Management: Advantages and Disadvantages.* [Online]
Available at: https://www.simplilearn.com/scrum-project-management-article
[Accessed 15 10 2023].

Magnus Mage (Pvt) Ltd., 2023. *The Essential Guide to Risk Management in Software Development.* [Online]
Available at: https://www.linkedin.com/pulse/essential-guide-risk-management-software-development-magnusmageltd/
[Accessed 12 11 2023].

Mascellino, A., 2023. *Supply Chain and APIs Top Security Concerns, CISO Survey Shows.* [Online]
Available at: https://www.infosecurity-magazine.com/news/supply-chain-top-security-concern/
[Accessed 12 10 2023].

Organ, C. & Bottorff, C., 2022. *Work Breakdown Structure (WBS) In Project Management.* [Online]
Available at: https://www.forbes.com/advisor/business/what-is-work-breakdown-structure/
[Accessed 15 11 2023].

Security Staff, 2022. *41% of organizations suffered API security incidents in the past year.* [Online]
Available at: https://www.securitymagazine.com/articles/97453-41-of-organizations-suffered-api-security-incidents-in-the-past-year
[Accessed 15 10 2023].

TeamGantt, 2023. *What Is a Gantt Chart? A How-to Guide with Examples for Project Management.* [Online]
Available at: https://www.teamgantt.com/what-is-a-gantt-chart
[Accessed 10 11 2023].

The Cyber Post, 2023. *APIs: Unveiling the Silent Killer of Cyber Security Risk Across Industries.* [Online]
Available at: https://thecyberpost.com/news/hackers/apis-unveiling-the-silent-killer-of-cyber-security-risk-across-industries/
[Accessed 10 10 2023].

13. Appendix

13.1 Common attack vectors

- Manipulating query parameters.
- Extracting excess data from responses.
- Disrupting service in the absence of rate limiting or response limiting constraints.
- Accessing unauthorized administrative functions by guessing endpoints.
- Stealing credentials.
- Exploiting security misconfigurations.

Risks of APIs:

- Excessive Data Exposure.
- Broken Authorization and Authentication.
- Compliance.
- API security Concerns (APIsec, 2022).

APIs are used by some large sectors, and the risks associated with not properly securing APIs are discussed below with specific case studies.

a. Healthcare:

One of the most severe data breaches that the industry-leading clinical laboratory service provider in the US, Quest Diagnostics, suffered resulted in a third-party API. Through the use of an exposed API, attackers were able to take advantage of a vulnerability in this third party's online payment page and obtain unauthorized access to the medical records of about 11.9 million patients.

According to some investigations, the average yearly API-related cyber loss in the US might be between $12 billion and $23 billion due to a lack of security APIs, while the global impact could be between $41 billion and $75 billion (Landi, 2019).

b. Financial Service Institution:

A stark reminder of the escalating threat posed by API attacks is provided by the Latitude Financial API leak. It is one of the biggest API breaches on record, with over 14 million records exposed, over 8 million driver's licenses taken, 53,000 passport numbers, and dozens of monthly financial statements (Naimisha, 2023).

c. Technology:

On November 1, 2022, Dropbox's internal GitHub code repositories were compromised by hackers. 130 internal code repositories, some of which contained API keys and user data, were made accessible to hackers as a result. For their phishing assault, hackers sent an email mimicking CircleCI, a well-known CI/CD pipeline. After that, users were directed to a fake CircleCI website and asked to enter their GitHub login information. After that, they received a One-Time Password, which they had to enter (Gatlan, 2022).

Return to the previous page.

13.2 Targeted Audience

Primary target:

Software companies are the most common target for this API security solutions since they are at the forefront of API development and utilization. They handle sensitive data processing and complicated API landscape management in addition to designing, developing, and maintaining APIs. Software companies need API security measures in order to protect their APIs from vulnerabilities and cyberattacks.

Secondary target:

Although the majority of API security system users are software companies, every company that exposes its APIs to the internet would have to think about adopting these solutions. This comprises companies in a range of sectors, including:

- **FinTech and e-commerce:** These companies are vulnerable to targeted attacks since they handle sensitive financial data and client information. Protecting sensitive data and preventing fraud in finance require API security.

- **Healthcare providers:** Under strict regulations like HIPAA, healthcare institutions are required to secure the sensitive patient data they retain and communicate. API security systems protect patient privacy and guarantee regulatory compliance.

- **Social media platforms:** Because they manage huge amounts of user data, social media platforms are frequently the focus of privacy infringement and data breaches. API security solutions protect user information and uphold customers.

- **Retailers:** Retailers connect online and in-store systems, handle payments, and keep track of inventory via APIs. API security systems guard consumer information and stop fraud in the retail industry.

- **Manufacturing companies:** Manufacturing companies connect supply chains, oversee production procedures, and gather data from Internet of Things devices via APIs. API security measures shield confidential information and keep production processes running smoothly.

Return to the previous page.

13.3 Considered Methodologies

Waterfall Model:

With the waterfall method, a sequential project plan is created to address the needs that are identified at the beginning of the project by gathering requirements from stakeholders and customers. Every project stage flows into the next and lowers progressively, resembling a waterfall, which is how the waterfall model acquired its name.

It is a linear software development process with a predetermined phase-by-phase order. The phases are: requirements gathering, design, implementation, verification, and maintenance and each phase must be completed before the next phase can begin.

- At the start of the project, requirements are obtained, and until the product is finished, no more client correspondence is permitted.
- The logical design and physical design subphases comprise the design phase.
- The actual code that programmers write is known as the implementation phase.
- During the verification phase, the client inspects the product to ensure that all requirements are met.
- The production team addresses any faults or errors found by the client during the maintenance period (ProjectManager.com, 2023).

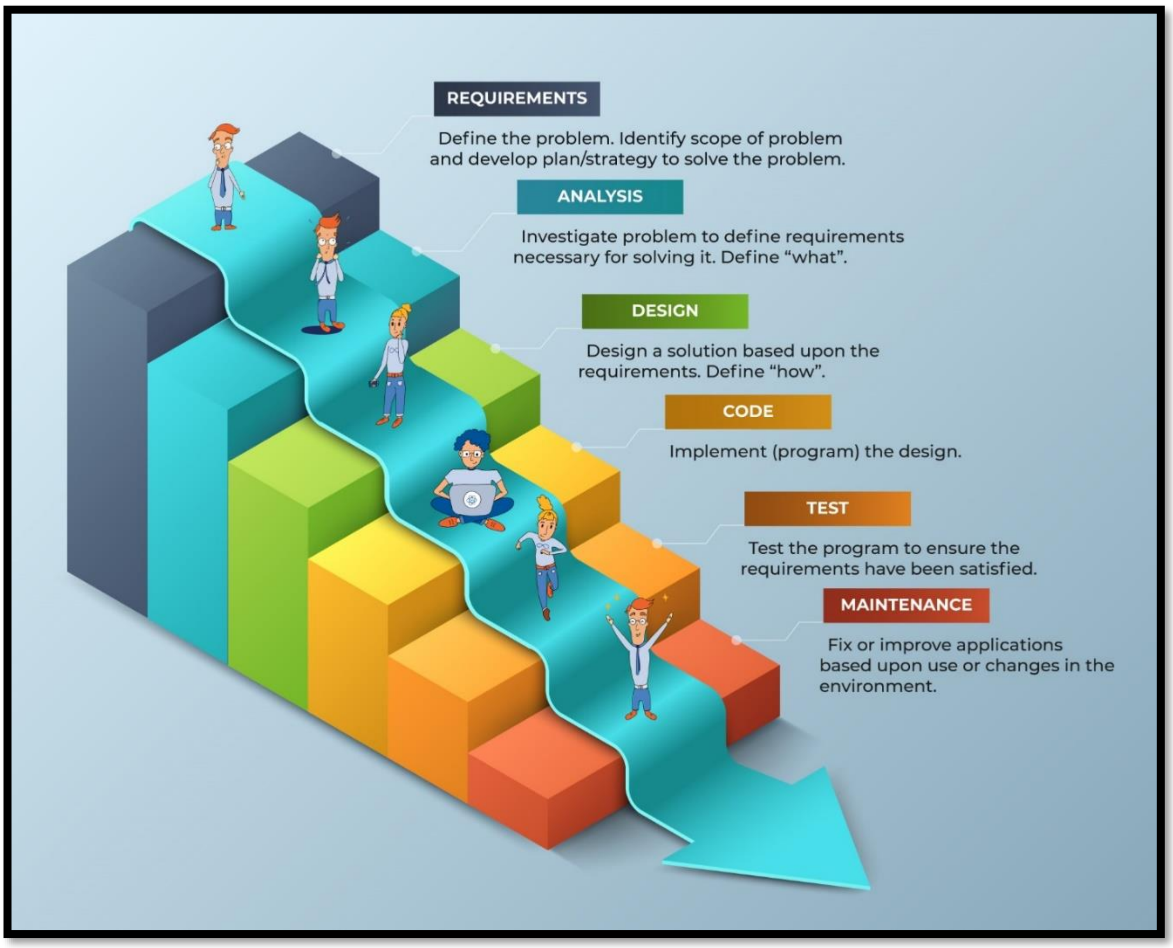

REQUIREMENTS
Define the problem. Identify scope of problem and develop plan/strategy to solve the problem.
ANALYSIS
Investigate problem to define requirements necessary for solving it. Define “what”.
DESIGN
Design a solution based upon the requirements. Define “how”.
CODE
Implement (program) the design.
TEST
Test the program to ensure the requirements have been satisfied.
MAINTENANCE
Fix or improve applications based upon use or changes in the environment.

Figure 11: Waterfall Model (Adam, 2021)

Prototyping Model:

Within the dynamic field of software development, the prototype model is a versatile and adaptive process that places an emphasis on ongoing refinement and early user feedback. In contrast to the waterfall methodology, which follows a strict order of steps, prototyping is an iterative procedure that facilitates continuous modifications and improvements over the course of the development cycle.

The following steps are commonly included in the prototyping model:

- **Requirements Collecting:** To get a rough idea of the intended system or product, the first step entails collecting and evaluating user needs.
- **Development of the First Prototype:** During the requirements gathering stage, a basic prototype is made that includes the main features and functionalities found.
- **User Assessment and Input:** Users and other relevant parties are shown the prototype for their assessment and input. Their observations are duly noted and taken into account.
- **Prototype Refinement:** Any usability problems or new features are added, and the prototype is improved and refined in light of the feedback received.
- **Iteration:** Until the prototype reaches the required degree of functionality and user satisfaction, the process of prototype evaluation, feedback assimilation, and refining is repeated.
- **Finalization and Implementation:** The prototype forms the foundation for the creation of the finished good or system after it has attained an acceptable degree of maturity (Lewis, 2023).

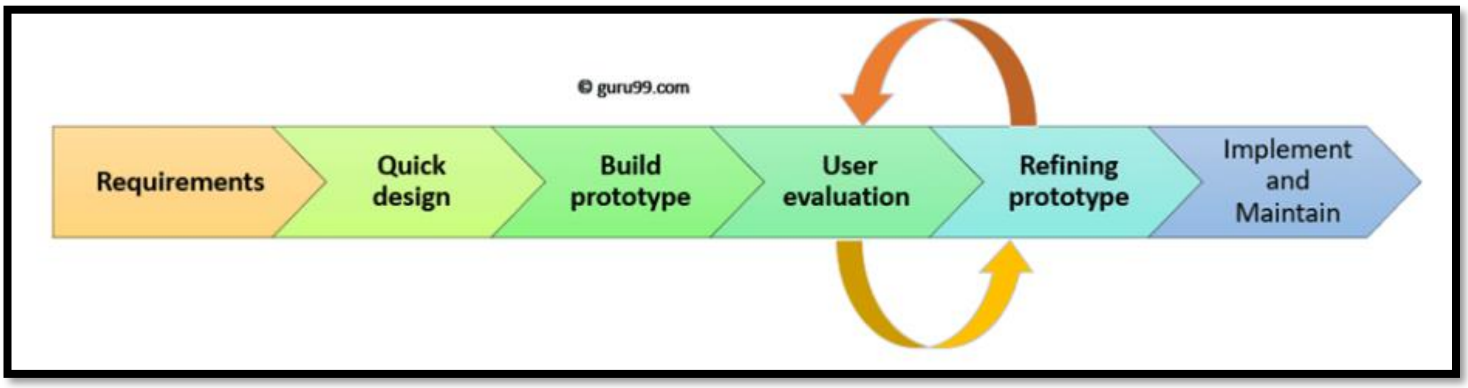


Figure 12: Prototype Model (Martin, 2023)

Spiral Model:

One paradigm that supports risk handling in the software development life cycle is the spiral model. When shown diagrammatically, it resembles a spiral with several loops. It is uncertain how many loops there are in the spiral; the number can differ from project to project. A phase of the software development process is denoted by each loop in the spiral.

Phases of the Spiral Framework:

- **Planning:** A plan for the upcoming iteration is developed, together with the project's goals and scope.
- **Risk analysis** involves identifying and evaluating potential hazards related to the iteration and developing mitigation solutions.
- **Engineering:** The requirements and risk analyses from earlier iterations are used to build the programme.
- **Evaluation:** Customer needs, quality standards, and risk concerns are taken into consideration when evaluating the generated programme.
- **Iteration:** The procedure is repeated, building on the knowledge and results from the previous iteration with each iteration (GeeksforGeeks, 2023).

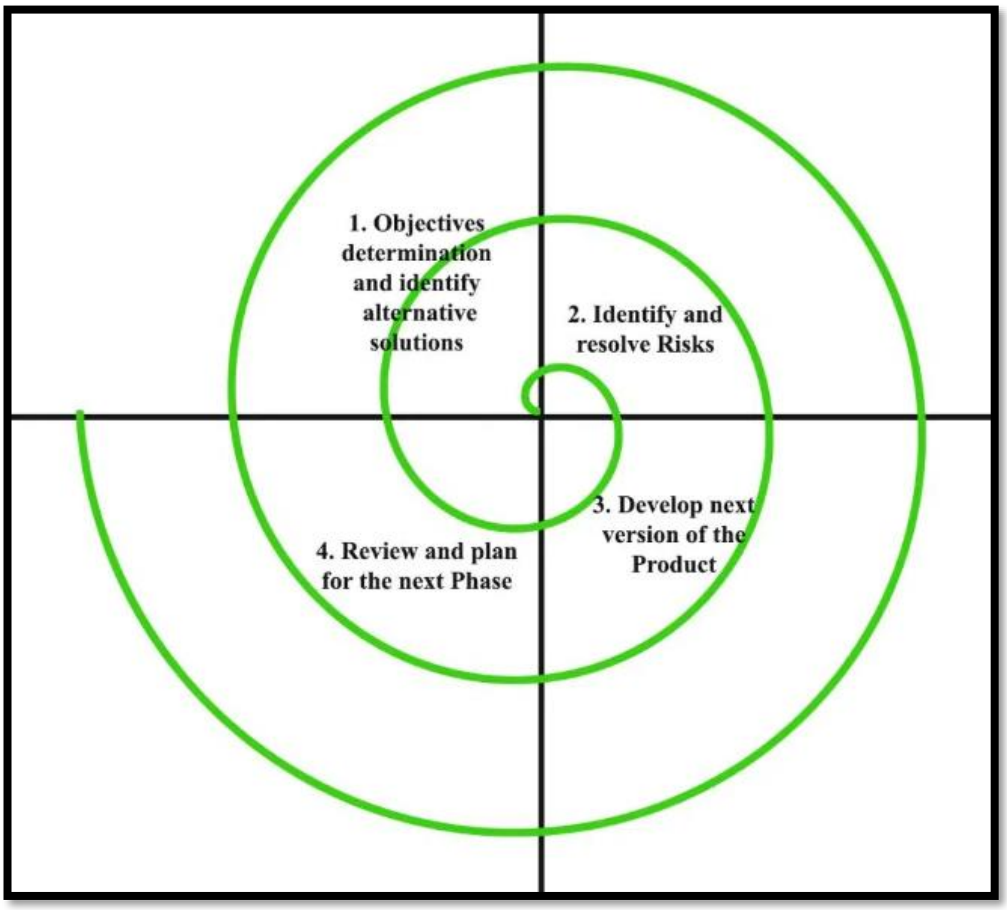

1. Objectives determination and identify alternative solutions
2. Identify and resolve Risks
3. Develop next version of the Product
4. Review and plan for the next Phase

Figure 13: Spiral Model (GeeksforGeeks, 2023)

Return to the previous page.

13.4 Agile Methodology

There are four values that are the pillars of Agile project management:

- Individuals and Interactions: Focus on teamwork and collaboration.
- Working Software: Give early and frequent delivery of functional software a top priority.
- Customer Collaboration: Involve clients at every stage of the creation process.
- Responding to change: Acknowledge change and make the necessary adjustments.

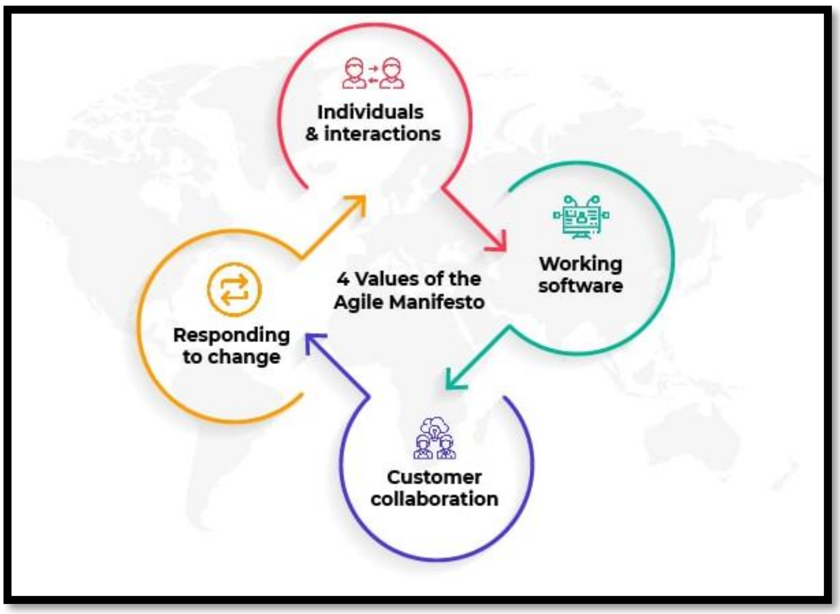


Figure 14: Values of Agile (CodeStringers, 2021)

The four values of Agile are the pillars of Agile methodology. From those values, the team developed 12 principles.

- **Frequent delivery:** Deliver software to clients on a regular basis and with speed.
- **Close collaboration:** Coordinate closely with stakeholders to match requirements with software.
- **Individual empowerment:** Empower individuals to succeed and contribute.
- **Self-organization:** Allow teams the freedom to decide and assume responsibility.
- **Continuous improvement:** Regular process evaluation and enhancement.
- **Customer feedback:** Gather feedback to improve the product.
- **Value focus:** Give providing value to customers a priority.
- **Sustainable practices:** To prevent burnout, work at a sustainable pace.
- **Technical excellence:** Make an investment in quality and maintainable code.
- **Simplicity:** Avoid complexity and concentrate on the important tasks.
- **Frequent releases:** Make modest software releases on a regular basis.
- **Face-to-face:** Prioritize in-person interactions to improve communication (Laoyan, 2022).

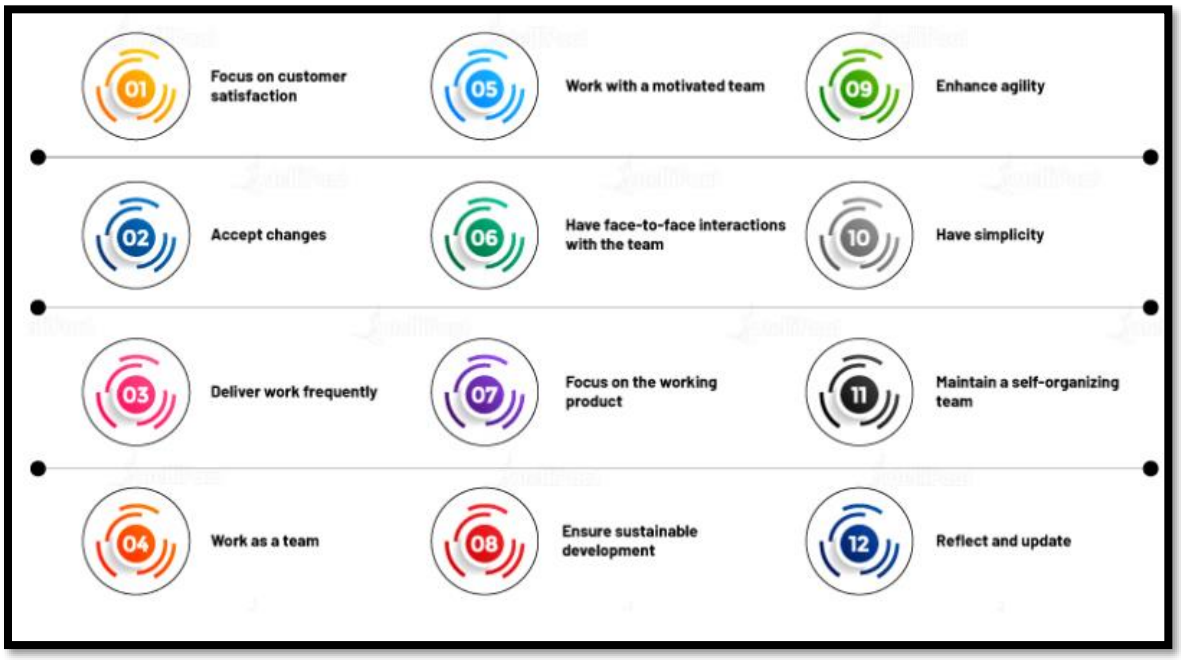

01 Focus on customer satisfaction
05 Work with a motivated team
09 Enhance agility
02 Accept changes
06 Have face-to-face interactions with the team
10 Have simplicity
03 Deliver work frequently
07 Focus on the working product
11 Maintain a self-organizing team
04 Work as a team
08 Ensure sustainable development
12 Reflect and update

Figure 15: Principles of Agile (Intellipaat Software Solutions , 2023)

The agile methodology is an umbrella for several different variations. The most common agile methodologies are:

- Kanban
- Extreme Programming
- Scrum

Kanban:

The Kanban methodology is widely used in the implementation of agile development. It necessitates real-time capacity communication and total transparency in the work. Because work items are clearly depicted on a kanban board, team members can always see the status of every job.

Individual tasks or work items are listed in several columns on a kanban board, which also represents different stages of a workflow (such as "To Do," "In Progress," and "Done"). Teams may see tasks, their progress, and any possible delays or bottlenecks in real time by using kanban boards. It increases productivity by assisting teams in setting priorities, limiting work in progress, and upholding an orderly and controllable flow of tasks (Atlassian, 2023).

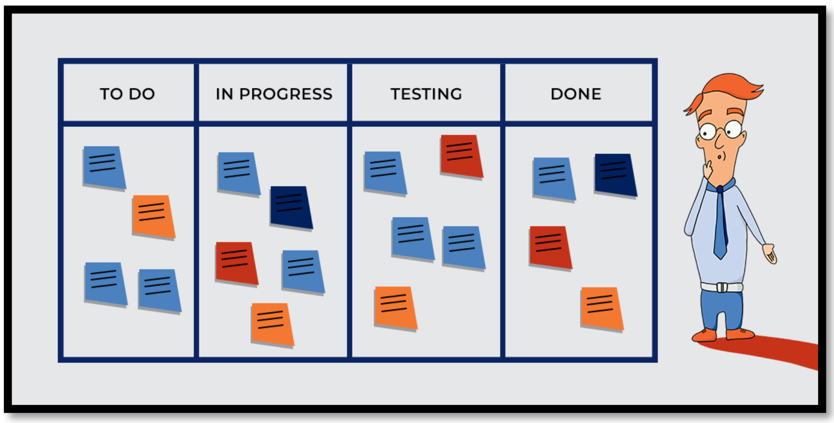


Figure 16: Kanban (Adam, 2023)

Extreme Programming (XP):

Extreme Programming is a software development process that aims to enable small to mid-sized teams to build high-quality software and adapt to changing needs. It is based on values, principles, and practices.

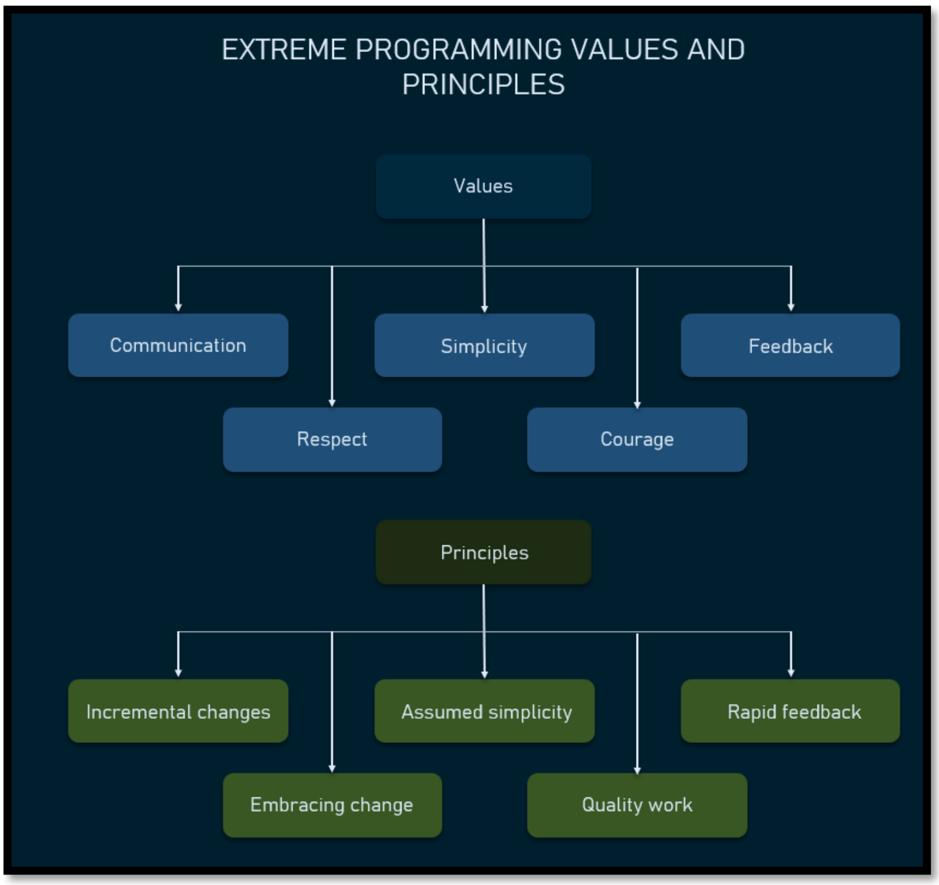


Figure 17: Extreme Programming (XP) (altexsoft, 2021)

Values:

- **Communication:** Open and regular communication promotes teamwork and problem-solving.
- **Simplicity:** Aim for simple designs and code that are easy to read, update, and work with.
- **Feedback:** Accept ongoing input from users and other team members to make the programme better.
- **Courage:** Be brave and take calculated chances in order to adapt and innovate.
- **Respect:** Value each other’s contributions and expertise to foster a positive team.

Principles:

- **Rapid Feedback:** Reduce feedback cycles to find and fix problems more rapidly.
- **Embrace Change:** Acknowledge and adapt to change as an opportunity.
- **Incremental change:** Implement small, manageable changes to minimize risk and facilitate adaptation.
- **Quality Work:** Produce high-quality software to create high-quality software (Venema, 2023).

The Scrum methodology was chosen for the project from among these agile methodologies.

Return to the previous page.

13.5 Justification for selected methodology

Due to its applicability to one-person development projects, Scrum was chosen for the project. The development of a security system by a single person can be easily managed with its built-in simplicity, versatility, and emphasis on feedback and improvement. The primary reasons for choosing Scrum are outlined below:

- Scrum's simple structure and small number of roles make it simple for one person to understand and use the approach.

- The flexibility of Scrum is especially important for security system development projects, as the requirements are frequently complicated and dynamic. Scrum can be adjusted to changing needs as they arise.

- Scrum's brief sprints and time-boxing methodology enable a single developer to effectively manage the project scope by avoiding the addition of redundant functions that may hinder development.

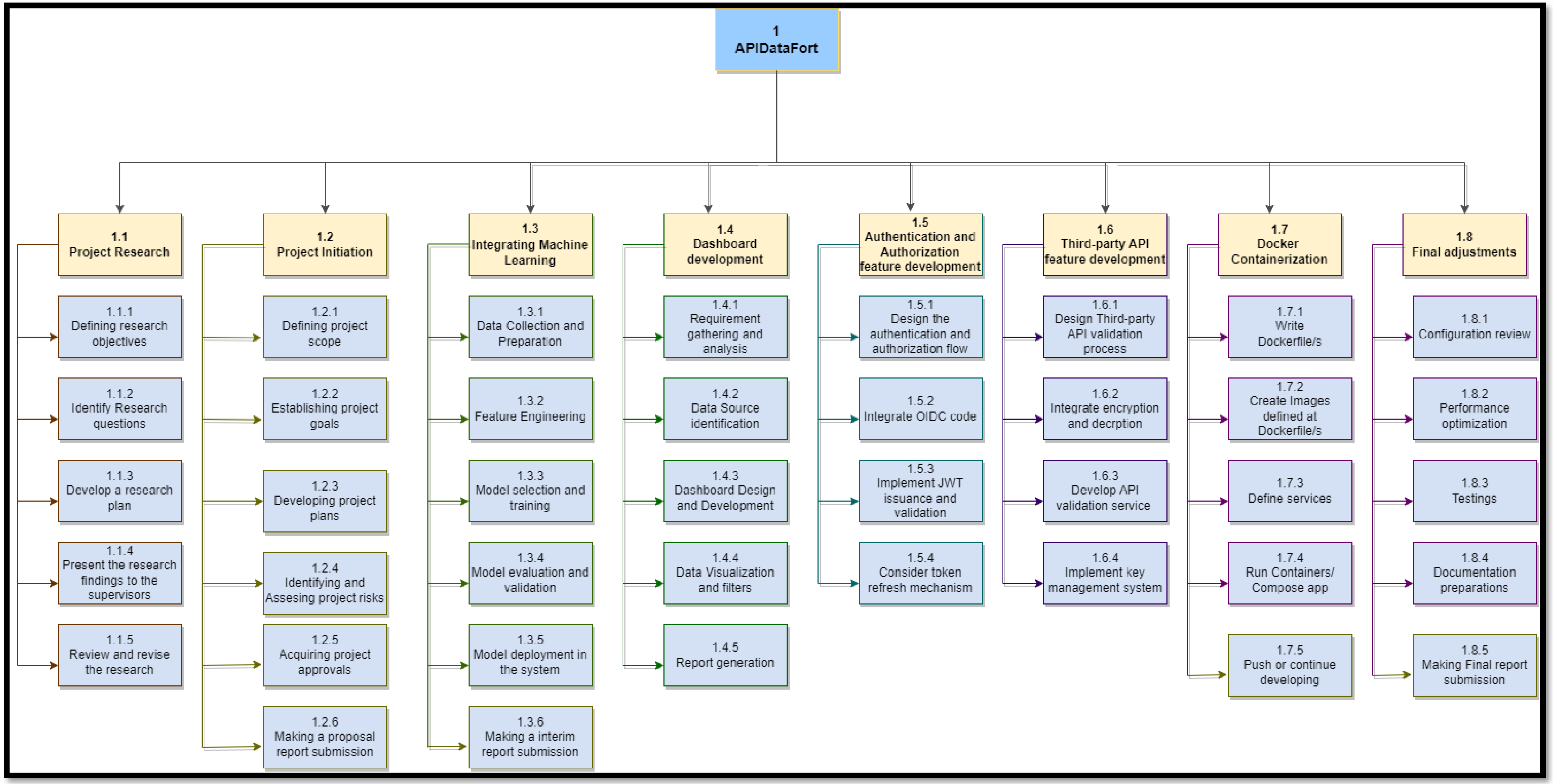
- Scrum places a strong emphasis on ongoing feedback and improvement, which helps developers spot problems early on and fix them to produce finest security systems that satisfy users.

Return to the previous page.

13.6 Charts

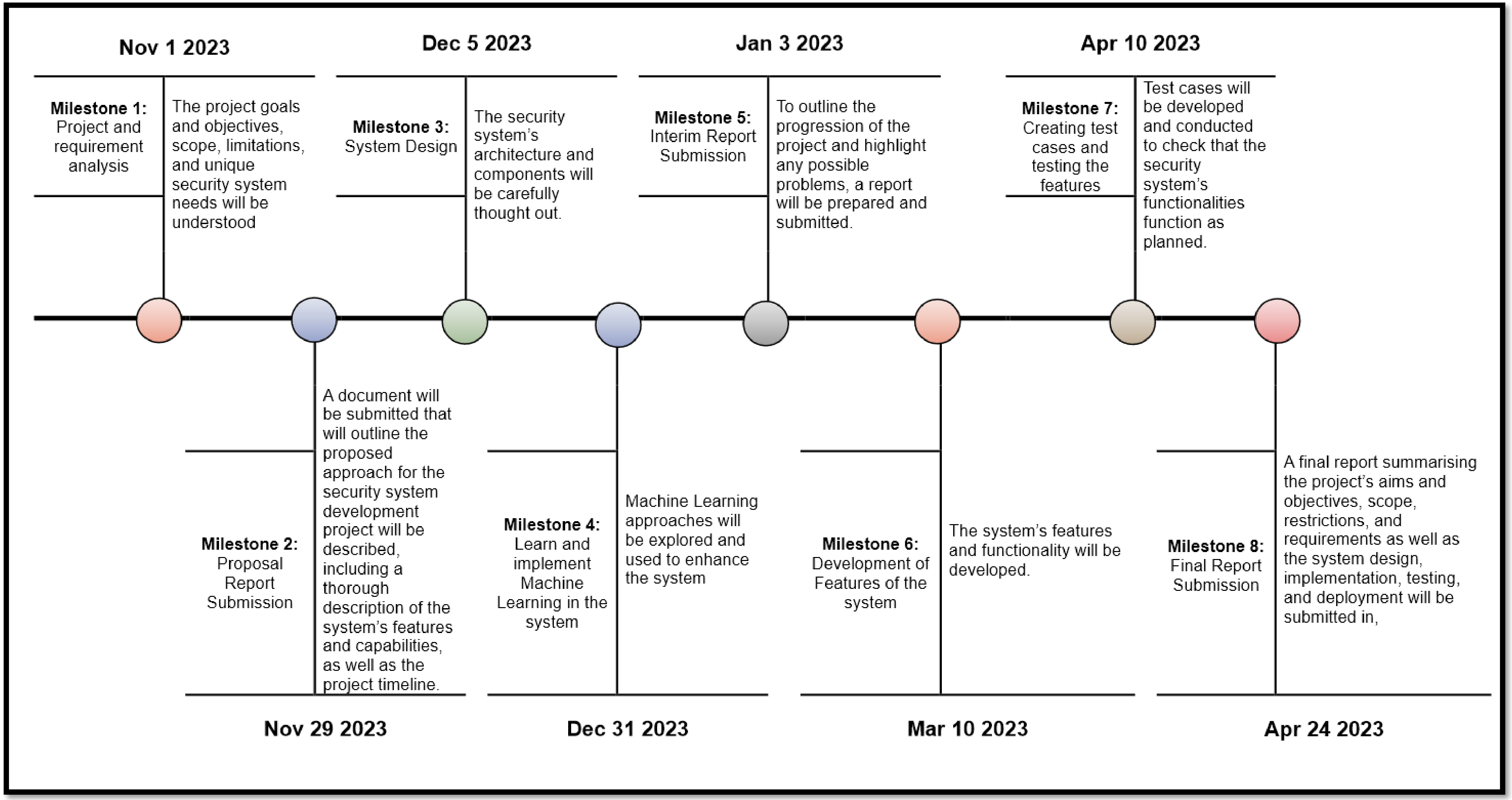
Work Breakdown Structure

Return to the previous page.


1
APIDataFort
1.1
Project Research
1.1.1
Defining research objectives
1.1.2
Identify Research questions
1.1.3
Develop a research plan
1.1.4
Present the research findings to the supervisors
1.1.5
Review and revise the research
1.2
Project Initiation
1.2.1
Defining project scope
1.2.2
Establishing project goals
1.2.3
Developing project plans
1.2.4
Identifying and Assesing project risks
1.2.5
Acquiring project approvals
1.2.6
Making a proposal report submission
1.3
Integrating Machine Learning
1.3.1
Data Collection and Preparation
1.3.2
Feature Engineering
1.3.3
Model selection and training
1.3.4
Model evaluation and validation
1.3.5
Model deployment in the system
1.3.6
Making a interim report submission
1.4
Dashboard development
1.4.1
Requirement gathering and analysis
1.4.2
Data Source identification
1.4.3
Dashboard Design and Development
1.4.4
Data Visualization and filters
1.4.5
Report generation
1.5
Authentication and Authorization feature development
1.5.1
Design the authentication and authorization flow
1.5.2
Integrate OIDC code
1.5.3
Implement JWT issuance and validation
1.5.4
Consider token refresh mechanism
1.6
Third-party API feature development
1.6.1
Design Third-party API validation process
1.6.2
Integrate encryption and decrption
1.6.3
Develop API validation service
1.6.4
Implement key management system
1.7
Docker Containerization
1.7.1
Write Dockerfile/s
1.7.2
Create Images defined at Dockerfile/s
1.7.3
Define services
1.7.4
Run Containers/ Compose app
1.7.5
Push or continue developing
1.8
Final adjustments
1.8.1
Configuration review
1.8.2
Performance optimization
1.8.3
Testings
1.8.4
Documentation preparations
1.8.5
Making Final report submission

Milestones

Return to the previous page.



Nov 1 2023

Milestone 1: Project and requirement analysis

The project goals and objectives, scope, limitations, and unique security system needs will be understood

Nov 29 2023

Milestone 2: Proposal Report Submission

A document will be submitted that will outline the proposed approach for the security system development project will be described, including a thorough description of the system's features and capabilities, as well as the project timeline.

Dec 5 2023

Milestone 3: System Design

The security system's architecture and components will be carefully thought out.

Dec 31 2023

Milestone 4: Learn and implement Machine Learning in the system

Machine Learning approaches will be explored and used to enhance the system

Jan 3 2023

Milestone 5: Interim Report Submission

To outline the progression of the project and highlight any possible problems, a report will be prepared and submitted.

Mar 10 2023

Milestone 6: Development of Features of the system

The system's features and functionality will be developed.

Apr 10 2023

Milestone 7: Creating test cases and testing the features

Test cases will be developed and conducted to check that the security system's functionalities function as planned.

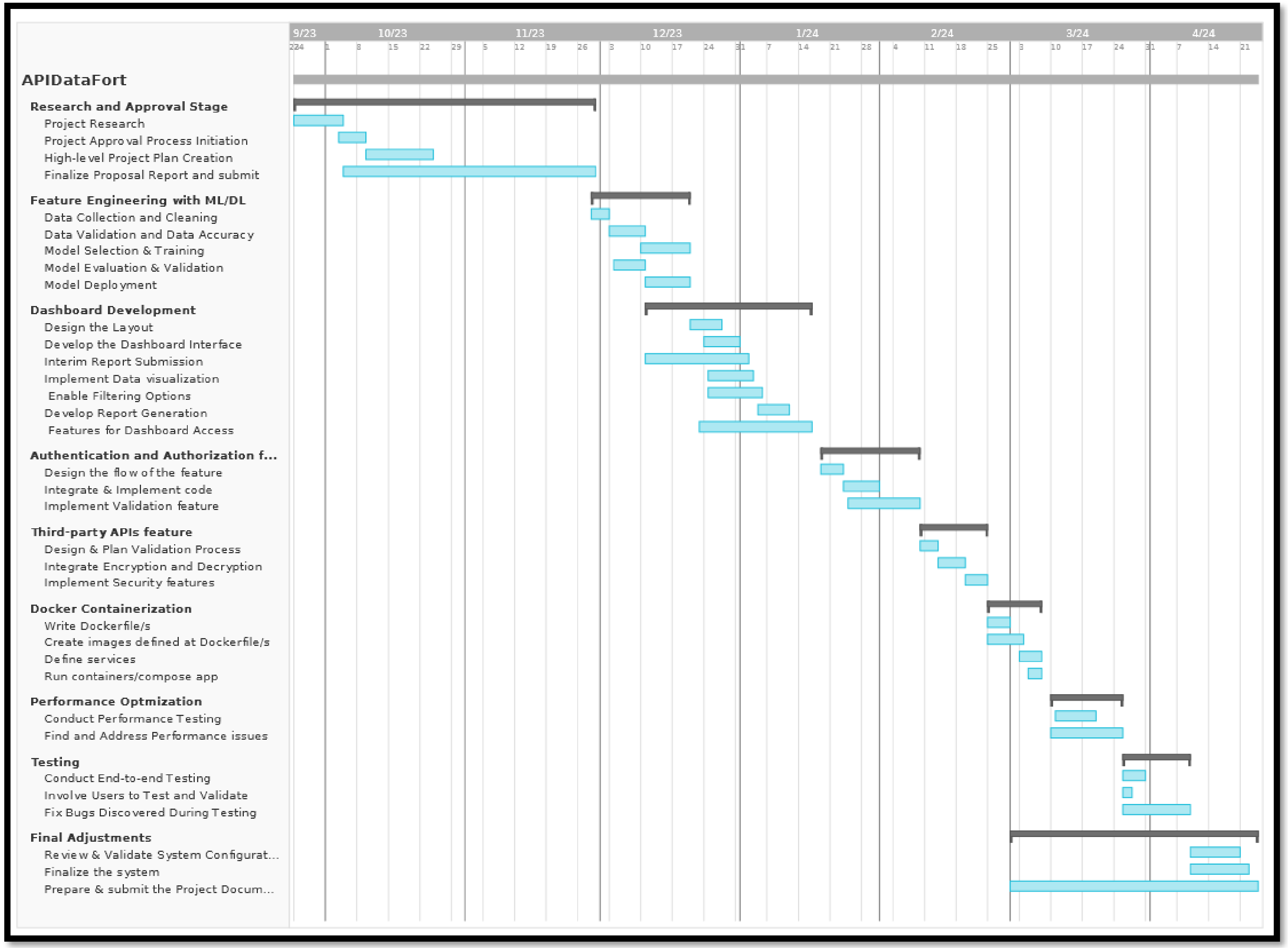
Apr 24 2023

Milestone 8: Final Report Submission

A final report summarising the project's aims and objectives, scope, restrictions, and requirements as well as the system design, implementation, testing, and deployment will be submitted in,

Gantt Chart

Return to the previous page.


9/23
10/23
11/23
12/23
1/24
2/24
3/24
4/24
APIDataFort
Research and Approval Stage
Project Research
Project Approval Process Initiation
High-level Project Plan Creation
Finalize Proposal Report and submit
Feature Engineering with ML/DL
Data Collection and Cleaning
Data Validation and Data Accuracy
Model Selection & Training
Model Evaluation & Validation
Model Deployment
Dashboard Development
Design the Layout
Develop the Dashboard Interface
Interim Report Submission
Implement Data visualization
Enable Filtering Options
Develop Report Generation
Features for Dashboard Access
Authentication and Authorization f...
Design the flow of the feature
Integrate & Implement code
Implement Validation feature
Third-party APIs feature
Design & Plan Validation Process
Integrate Encryption and Decryption
Implement Security features
Docker Containerization
Write Dockerfile/s
Create images defined at Dockerfile/s
Define services
Run containers/compose app
Performance Optmization
Conduct Performance Testing
Find and Address Performance issues
Testing
Conduct End-to-end Testing
Involve Users to Test and Validate
Fix Bugs Discovered During Testing
Final Adjustments
Review & Validate System Configurat...
Finalize the system
Prepare & submit the Project Docum...