**Baku Higher Oil School**

**Information Technology Department**

**Lecturer: Khayyam H. Masiyev**

**Course Name: System Analysis and Design**

**Student Name: Ziya Malikov**

**Student ID: 220104023**

**Final Report: Security Evaluation of Cloud Services**

**Table of Contents**

[1. Abstract 3](#_Toc184927189)

[2. Introduction 3](#_Toc184927190)

[3. Preliminary Investigation Report 6](#_Toc184927191)

[3.1 Fishbone diagram 6](#_Toc184927192)

[3.2 Project Scope and Constraints 7](#_Toc184927193)

[3.3 Fact Findings 9](#_Toc184927194)

[3.3.1 Survey 9](#_Toc184927195)

[3.3.2 Interview 12](#_Toc184927196)

[3.4 Analyze project usability, cost, benefit, schedule data 13](#_Toc184927197)

[3.5 Evaluation of Feasibility 14](#_Toc184927198)

[3.5.1 Operational Feasibility 14](#_Toc184927199)

[3.5.2 Technical Feasibility 14](#_Toc184927200)

[3.5.3 Economic Feasibility 15](#_Toc184927201)

[3.5.4 Schedule Feasibility: 15](#_Toc184927202)

[3.6 Project Request Form 16](#_Toc184927203)

[3.6.1 Description of the project 16](#_Toc184927204)

[3.6.2 Requested Features 16](#_Toc184927205)

[3.6.3 Main Objectives of the Project 16](#_Toc184927206)

[3.7 Memo 19](#_Toc184927207)

[3.7.1 Overview of Meeting 19](#_Toc184927208)

[3.7.2 Problems and Result Solutions 19](#_Toc184927209)

[3.7.3 Result of Meeting 20](#_Toc184927210)

[3.8 Recommendations 20](#_Toc184927211)

[4. Requirement Analysis 21](#_Toc184927212)

[4.1 Work Breakdown Structure (WBS) 21](#_Toc184927213)

[4.2 Gantt Chart 24](#_Toc184927214)

[4.3 CPM 29](#_Toc184927215)

[4.4 PERT Chart 30](#_Toc184927216)

[4.5 List of requirements 31](#_Toc184927217)

[4.5.1 User Requirements 31](#_Toc184927218)

[4.5.2 System Requirements 31](#_Toc184927219)

[5. Agile parts 33](#_Toc184927220)

[5.1 Product Backlogs 33](#_Toc184927221)

[5.2 User stories 34](#_Toc184927222)

[6. UML Diagram 37](#_Toc184927223)

[6.1 Context Model Diagram 37](#_Toc184927224)

[6.2 Activity Diagram 38](#_Toc184927225)

[6.3 User Case Diagram 39](#_Toc184927226)

[6.4 Sequence Diagrams 40](#_Toc184927227)

[6.5 Class Diagram 43](#_Toc184927228)

[6.6 ER Diagram 43](#_Toc184927229)

[7. Test Strategy Documentation 44](#_Toc184927230)

[7.1. Test Life Cycle 44](#_Toc184927231)

[7.2. Integration Test 45](#_Toc184927232)

[7.3. Penetration Test 56](#_Toc184927233)

[8. Defect Management 62](#_Toc184927234)

[Defect Management Life Cycle 62](#_Toc184927235)

[9. Conclusion 63](#_Toc184927236)

[10. Remediation Steps 63](#_Toc184927237)

[11. References 64](#_Toc184927238)

# Abstract

As organizations increasingly adopt cloud services for data storage and processing, ensuring robust security becomes essential to protect sensitive information and maintain user trust. This project provides a comprehensive evaluation of cloud service security, focusing on identifying potential vulnerabilities, assessing the effectiveness of existing security measures, and proposing enhancements. Key areas of analysis include data protection, access control, encryption, and compliance with security standards. Through methods such as threat modeling, risk assessment, and analysis of industry best practices, this project aims to highlight the strengths and weaknesses of current cloud security practices. The findings aim to guide organizations in making informed decisions about cloud adoption while ensuring compliance and minimizing risks. This evaluation contributes to the field by offering insights and recommendations for enhancing security within cloud environments.

# Introduction

With the rise of cloud computing as a foundation for modern data storage and application deployment, secure cloud environments have become essential. Cloud security protects sensitive data from breaches, helping organizations avoid financial losses, reputational damage, and legal issues. Secure cloud systems offer key benefits, including data encryption to maintain privacy, access controls to prevent unauthorized access, and scalable security measures that adapt as an organization grows. Additionally, compliance with regulations such as GDPR is more achievable within secure cloud frameworks. As cyber threats evolve, robust cloud security is vital to ensure trust, legal compliance, and business continuity. Understanding real-case scenarios provides valuable insights into common security pitfalls and the evolving landscape of cloud threats. The following section examines notable cloud security breaches, illustrating the need for organizations to prioritize security practices in their cloud environments.

* **The Capital One Data Breach (2019)**

The Capital One Data Breach (2019) exposed the personal data of over 100 million U.S. customers and 6 million Canadians. The breach was caused by a misconfigured Web Application Firewall (WAF) in Capital One’s AWS cloud infrastructure. Hacker Paige Thompson, a former AWS employee, exploited a Server-Side Request Forgery (SSRF) vulnerability to gain access to sensitive data. Compromised information included 140,000 Social Security numbers, 80,000 bank account numbers, and credit scores. Thompson shared details of the breach online, which led to her arrest in July 2019. Capital One was fined $80 million by U.S. regulators for failing to adequately secure its cloud environment. A $190 million class-action settlement was reached with affected customers.

* **Uber Data Breach (2016)**

The Uber Data Breach (2016) exposed the personal data of 57 million customers and drivers. Hackers gained access to Uber’s GitHub account, where sensitive credentials for Uber's AWS servers were stored. Using these credentials, the attackers accessed a trove of customer and driver information, including names, email addresses, and phone numbers.

Instead of reporting the breach, Uber paid the hackers $100,000 to delete the stolen data and keep the breach quiet. This cover-up lasted for over a year before it was publicly revealed in 2017. As a result, Uber faced significant legal and regulatory consequences, including a $148 million settlement with U.S. authorities.

* **Tesla Kubernetes Cluster Hijack (2018)**

Tesla Kubernetes Cluster Hijack (2018) was a cloud security breach where attackers infiltrated Tesla’s Kubernetes cluster and used it for cryptocurrency mining. The breach was discovered by cybersecurity firm RedLock, which found that Tesla's Kubernetes console was left unsecured and exposed to the internet without a password. This allowed the attackers to gain access to Tesla's Amazon Web Services (AWS) environment.

Once inside, the hackers set up a cryptojacking operation to mine cryptocurrency using Tesla's computing power, while remaining undetected by masking their traffic. Sensitive data such as telemetry information was potentially exposed, though Tesla confirmed that customer data was not compromised. Tesla quickly secured the environment after being alerted by RedLock.

* **Code Spaces DDoS Attack (2014)**

The Code Spaces DDoS Attack (2014) was a devastating cyberattack that resulted in the company's permanent shutdown. Code Spaces, a cloud-based code-hosting and project management service, suffered a Distributed Denial of Service (DDoS) attack followed by an intrusion into its Amazon Web Services (AWS) control panel. The attacker demanded ransom, and when Code Spaces refused to comply, the attacker gained access to their AWS console and deleted most of their data, including backups.

The loss of critical customer information and infrastructure made recovery impossible. Code Spaces attempted to restore services, but the damage was too extensive, leading to the company announcing its closure.

* **Accenture Ransomware Attack (2021)**

The Accenture Ransomware Attack (2021) underscored critical vulnerabilities in cloud security, as the consulting firm faced a breach orchestrated by the LockBit ransomware group. Attackers claimed to have stolen 6 terabytes of sensitive data, including client information, and demanded a ransom of $50 million. Although Accenture quickly contained the incident and asserted that their operations and client services were unaffected, the attack raised significant concerns about the security of cloud environments.

* **Dropbox Data Breach (2012)**

The Dropbox Data Breach (2012) was a significant incident that highlighted vulnerabilities in cloud security, affecting the accounts of 68 million users. The breach occurred when attackers accessed Dropbox's internal systems using credentials stolen from a compromised employee account. This incident raised alarms about the importance of securing employee accounts and implementing strong access controls in cloud environments.

Dropbox quickly responded by notifying affected users and enhancing their security measures, including the introduction of two-factor authentication.

* **Google Cloud Project Breach (2019)**

The Google Cloud Project Breach (2019) was a significant incident where sensitive data from various organizations was exposed due to misconfigured Google Cloud Storage buckets. Security researchers identified that improperly configured settings allowed public access to these buckets, which contained a wealth of confidential information, including personal data, financial records, and proprietary business information.

The breach was not limited to a single organization; multiple companies were affected, demonstrating the widespread risk associated with cloud storage misconfigurations. The exposed data included customer records, internal documents, and other sensitive materials that could be exploited by malicious actors. Researchers reported that some of the data was accessible without any authentication, making it easy for anyone with the link to access the information.

# Preliminary Investigation Report

## 3.1 Fishbone diagram

**A diagram with blue rectangles and text

Description automatically generated**

When evaluating the security of cloud services, several important issues must be taken into account. Security breaches in cloud services often stem from several critical areas that organizations need to address. A major factor is the **lack of sufficient encryption**, which leaves sensitive data vulnerable to unauthorized access. This is compounded by **vulnerabilities within the cloud infrastructure** itself, which can be exploited by attackers if not properly secured. Additionally, the absence of **Multi-Factor Authentication (MFA)** significantly weakens user account security, making it easier for malicious actors to gain access through compromised credentials.

Another key area of concern is the role of people and processes. Many breaches result from **human errors in configuration** due to a lack of comprehensive security training. This is often exacerbated by **weak security policies** and inadequate access management, allowing unauthorized users to gain entry to critical systems. In many organizations, there is also a noticeable gap in **incident response plans**—without effective measures in place, response times are delayed, increasing the impact of security incidents.

From an environmental perspective, there's often confusion around **shared responsibility** between cloud providers and clients. Many organizations mistakenly assume that security is fully handled by the cloud provider, leading to unaddressed vulnerabilities. Furthermore, failure to comply with industry **regulations and standards** not only increases legal risks but also opens up further opportunities for breaches.

## 3.2 Project Scope and Constraints

**Project Scope:**

The main objective of this project is to conduct an in-depth evaluation of the security posture of cloud services through the development of a **Cloud Security Evaluation App**. The app will automatically assess security vulnerabilities and compliance in cloud environments, while integrating with SIEM solutions for enhanced threat detection and response. This project will focus on ensuring that the app supports different cloud platforms, emphasizes security best practices, and incorporates the necessary tools for continuous monitoring and incident management. The scope of this project includes several key aspects:

* **Cloud Service Security Assessment**: The app will assess the security configurations, vulnerabilities, and compliance status of cloud services used by organizations. This includes evaluating access controls, encryption standards, and identifying misconfigurations that could lead to security breaches.
* **Integration with SIEM Solutions**: We will ensure that the app integrates effectively with major SIEM platforms, facilitating the monitoring and analysis of security events. This involves assessing the app’s ability to incorporate security alerts, incident response mechanisms, and log management.
* **Disaster Recovery and Business Continuity**: The project will include the development of disaster recovery and business continuity plans to ensure the app can handle disruptions or failures in cloud services.
* **Incident Detection and Response**: The app will leverage SIEM capabilities to provide real-time alerts for suspicious activity, potential security breaches, and vulnerabilities. The app will also provide guidance on how to respond to such incidents.
* **Regulatory Compliance Evaluation**: The app will check compliance with various industry standards and regulations (e.g., GDPR, HIPAA, NIST) for the cloud environments in use. It will provide recommendations for improving compliance status.
* **Automated Vulnerability Scanning**: The app will automatically scan for known vulnerabilities, outdated software, and misconfigurations in cloud environments to ensure that organizations are protected from potential exploits.
* **Scalability and Multi-Organization Support**: The app will be designed to scale for use by multiple organizations, allowing each to evaluate their own cloud environments independently while maintaining secure operations.
* **Data Protection**: Ensuring the app uses strong encryption for both data in transit and data at rest to protect sensitive information from unauthorized access.

A diagram of a diagram

Description automatically generated

**Deliverables:**

The final deliverable for this project will be the **Cloud Security Evaluation App**, complete with SIEM integration, automated security evaluation tools, compliance assessment capabilities, and real-time threat detection. The app will provide users with comprehensive reports and actionable recommendations to enhance cloud security.

**Stakeholders:**

* **Organizations' IT Departments**: Responsible for managing the cloud infrastructure and security posture.
* **Cloud Architects**: Involved in designing and maintaining cloud environments.
* **Cloud Security Engineers**: Tasked with ensuring the security of cloud services and integrations.
* **Security Analysts**: Using the app to assess and monitor security threats in real-time.
* **Compliance Officers**: Overseeing the regulatory compliance of cloud environments.
* **End Users**: Organizations of various sizes and industries utilizing cloud services.

**Constraints:**

* **Time Constraint**: The project must be completed within three months, considering the complexity of cloud security evaluations and the integration of SIEM solutions.
* **Budget**: The project must stay within the allocated budget for development tools, cloud service usage, and any necessary external expertise.
* **Technological Limitations**: Compatibility with different cloud environments and SIEM tools can introduce technical challenges. The app must function across a variety of cloud providers (e.g., AWS, Azure, Google Cloud) and support integration with different SIEM platforms.
* **Security Risks**: Risks such as unauthorized access, data loss, and security breaches need to be considered when developing the app.
* **Regulatory and Compliance Challenges**: Ensuring that the app adheres to evolving compliance requirements and that sensitive data remains secure during evaluations is critical.
* **Scalability**: The app must be designed to handle growth in data volume and the number of users without performance degradation.
* **Third-Party Integration**: The app must account for integration with external cloud services, which could introduce unexpected challenges related to infrastructure and hosting choices.

## 3.3 Fact Findings

### 3.3.1 Survey

A diagram with colorful rectangular shapes

Description automatically generated with medium confidenceThe surveys were conducted with IT students and related individuals to understand their views on cloud security. Participants were asked about the cloud providers used by their organizations, their confidence in the security measures provided by these providers, and the key security features they prioritize when evaluating a cloud service. They were also asked whether their organizations use third-party tools like SIEM systems for monitoring security events in the cloud, and how important they believe SIEM systems are for managing security threats and incidents. The results provided valuable insights into the use of cloud services and security practices.

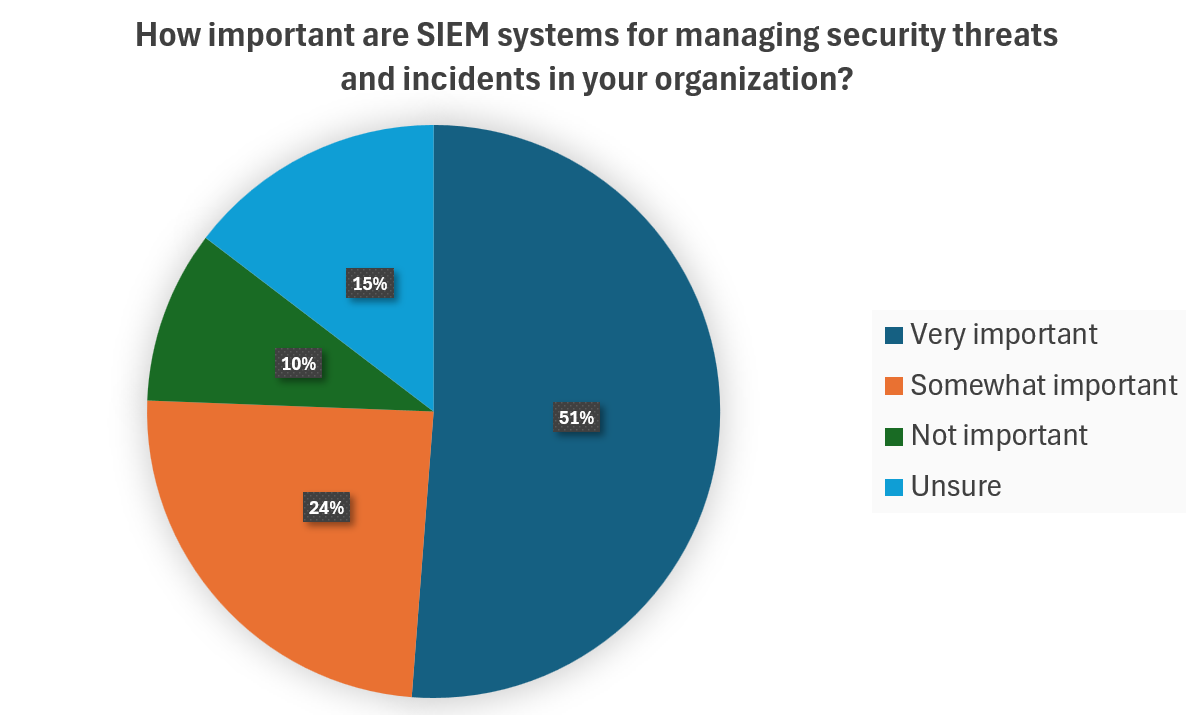
A graph of company data

Description automatically generated with medium confidence**A graph of data security

Description automatically generated**

**A graph of data being used

Description automatically generated with medium confidence**

****

### 3.3.2 Interview

We have applied methods of interviews:

*Interview Steps:*

1. Determining people to interview.

2. Asking for some needed information related to the projects.

3. Evaluation of answers of questions during interview and results.

4**.** Documentation process of interview.

1. **What’s the purpose of the project?**  
   The project aims to develop a Cloud Security Evaluation App that automatically assesses the security posture of cloud environments. It will identify vulnerabilities, check compliance with regulations, and integrate with SIEM solutions for enhanced threat detection, helping organizations secure their cloud infrastructure and reduce data breach risks.
2. **What are the problems that can be faced in the future?**  
   Future challenges include evolving cyber threats, changes in compliance standards, compatibility issues with different cloud providers, integration difficulties with various SIEM tools, and scalability concerns as data volumes increase.
3. **Who are our competitors?**  
   Competitors include Qualys, Palo Alto Prisma Cloud, Microsoft Defender for Cloud, and Trend Micro. While they offer security solutions, our app focuses on automated compliance checks, SIEM integration, and multi-cloud support.
4. **What are our differences from other related projects?**  
   Our app differentiates itself by providing automated compliance assessments, seamless SIEM integration, user-friendly reports, and scalability for multi-organization use, unlike many existing tools that focus primarily on threat detection.
5. **What changes will there be during the development of the project?**  
   Yes, changes are likely based on user feedback and testing. We may add new features, improve the interface, adjust for regulatory updates, and optimize performance for better scalability.
6. **How advertisement system will be provided?** We plan to use digital marketing, social media, content creation, and partnerships with cloud providers to promote the app. Webinars, case studies, and free trials will also help attract users and demonstrate the app's value.
7. **How are you going to deal with financial issues?** We will manage financial issues by using free resources, cloud credits, and discounts.

## 3.4 Analyze project usability, cost, benefit, schedule data

Here we are explaining each concept in detail based on our project. Primarily, usability of service, cost, benefit of that and schedule.

**Usability**: The Cloud Security Evaluation App is designed to be highly usable, focusing on accessibility for IT professionals, security analysts, and compliance officers. It features an intuitive dashboard that simplifies security assessments, real-time alerts, and compliance checks. The goal is to make the app easy to navigate even for users with limited technical expertise. By automating processes such as vulnerability scanning and compliance reporting, the app improves the efficiency of security management.

**Cost**: In terms of costs, the project is planned to be cost-effective. Free-tier cloud services, open-source tools, and existing resources will be used to minimize expenses. The primary costs include cloud service usage, which can be offset by using cloud credits from platforms like AWS, Azure, or Google Cloud. Development costs will be kept low by using free or open-source software. Hosting and maintenance expenses are expected to be minimal initially, especially by leveraging free cloud credits, while future costs will depend on scaling as the user base grows. Additionally, SIEM integration can be achieved using free trials or community editions during the development phase, making it a budget-friendly approach.

**Benefit**: The benefits of this app are significant for organizations looking to strengthen their cloud security. The automated security assessments save time by eliminating the need for manual checks, thereby increasing operational efficiency. It also helps organizations improve their compliance with industry regulations such as GDPR and HIPAA, reducing the risk of legal issues. The app enhances threat detection capabilities by providing real-time alerts through SIEM integration, helping organizations prevent breaches before they escalate. By offering scalability, the app ensures that it can grow with an organization's needs, making it a cost-effective solution in the long term. Overall, automating security evaluations can lead to considerable cost savings by reducing reliance on external audits and minimizing potential financial losses from data breaches.

**Schedule**: The project is planned to be completed within three months. The initial phase will focus on requirement analysis, design, and setup, followed by developing core features like security assessments and compliance checks in the planning phase. Afterward, focus will shift to SIEM integration, testing plans, and gathering feedback. Usability testing, final optimizations, and preparing documentation will be done before the project is handed off for development. The goal is to ensure all the groundwork is laid for smooth development, with adequate time for testing and adjustments.

## 3.5 Evaluation of Feasibility

With the help of interview questionnaires and surveys, we identified the feasibilities. The primary focus is to determine system requirements that can significantly impact the development stage.

### 3.5.1 Operational Feasibility

This project aims to provide a user-friendly tool to help IT teams and security professionals assess and improve cloud security. It will be compatible with major cloud platforms and integrate with SIEM solutions, addressing the growing need for cloud security and regulatory compliance. The tool’s simplicity and practicality make it suitable for businesses of all sizes, and there is strong demand for such a product in the current market.

**SWOT:**

* **Strengths**: Meets the urgent need for cloud security evaluation and integrates with major cloud platforms and security tools, making it highly adaptable.
* **Weaknesses**: User adoption, especially among smaller businesses without dedicated security teams, may be slow.
* **Opportunities**: As cloud adoption grows, the demand for security tools like this will increase, offering significant market potential.
* **Threats**: Competitors with broader features could impact adoption, but focusing on ease of use and targeted functionality will help differentiate the app.

### 3.5.2 Technical Feasibility

The project is technically feasible as it involves creating an app that integrates with existing cloud platforms (AWS, Azure, Google Cloud) and security tools like SIEM systems. The key challenge will be ensuring the app can scan for security issues, check compliance, and provide accurate reports. Additionally, the app must be secure, using encryption to protect data during evaluations. Given the available technologies and cloud services, the project is achievable.

**SWOT:**

* **Strengths**: The app will use widely supported cloud platforms and security tools, making integration feasible.
* **Weaknesses**: Ensuring compatibility with different cloud services and maintaining security during data evaluation could be challenging.
* **Opportunities**: The app can offer valuable automated security checks and integrate with SIEM systems to enhance cloud security.
* **Threats**: Changes in cloud services or security standards could require updates, but these can be managed over time.

### 3.5.3 Economic Feasibility

The primary expenses for this project will be related to cloud services, including AWS, Google Cloud, and Microsoft Azure. These platforms are estimated to cost at least $100 per month each, covering the essential services and tools needed for the project. As a student, we can benefit from free or discounted cloud services for up to two months, but if the project extends beyond this, additional funds will need to be allocated. Another major expense will be the cost of assembling a development team, with an estimated monthly cost of $200 per team member. Additionally, there may be costs for third-party tools and services required to enhance the functionality of the app, such as SIEM integrations or vulnerability scanning tools.

**SWOT for Economic Feasibility:**

* **Strengths**: Using student accounts for cloud services helps reduce costs, allowing us to maximize our budget.
* **Weaknesses**: Limited budget for marketing and user acquisition could delay the project's visibility and adoption.
* **Opportunities**: Potential revenue could be generated through partnerships or licensing the system to organizations in need of cloud security tools.
* **Threats**: Some cloud services and third-party APIs may not be free, leading to unanticipated costs if the project runs longer than expected.

3.5.4 Schedule Feasibility:The project is set to be completed within a three-month timeframe, with a clear breakdown of tasks. The first two monthes will focus on research, design, and requirement gathering, while the third month will be dedicated to finalizing the design, testing, and preparing for deployment. To ensure the project stays on schedule, tasks will be prioritized and aligned with the available resources. Any delay may result in missed opportunities, so sticking to the timeline is critical.

**Project Timeline:**

* **Start date**: 15.09.2024
* **Development and Planning**: 8 weeks
* **Testing and Finalization**: 4 weeks
* **Finish date**: 14.12.2024

| **Phase** | **Activities** | **Duration** |
| --- | --- | --- |
| **1. Development and Planning** | Define project scope, identify stakeholders, develop project plan, research security vulnerabilities, evaluate cloud platforms, design user interface, specify technical requirements, select SIEM tools, set project milestones, and prepare timeline. | 8 weeks |
| **2. Testing and Finalization** | Perform functionality testing, conduct integration testing, address identified bugs, improve design, finalize technical documentation, gather user feedback, prepare project report, make final adjustments, ensure compliance, and close project tasks. | 4 weeks |

**SWOT for Schedule Feasibility:**

* **Strengths**: Tasks can be completed in parallel, helping to expedite progress.
* **Weaknesses**: Delays due to lack of resources or unforeseen issues can disrupt the schedule.
* **Opportunities**: Agile methodology allows flexibility to adapt and reassign tasks if needed, ensuring continuous progress.
* **Threats**: Unforeseen challenges or resource constraints could delay project completion, requiring additional time to address issues.

## 3.6 Project Request Form

### 3.6.1 Description of the project

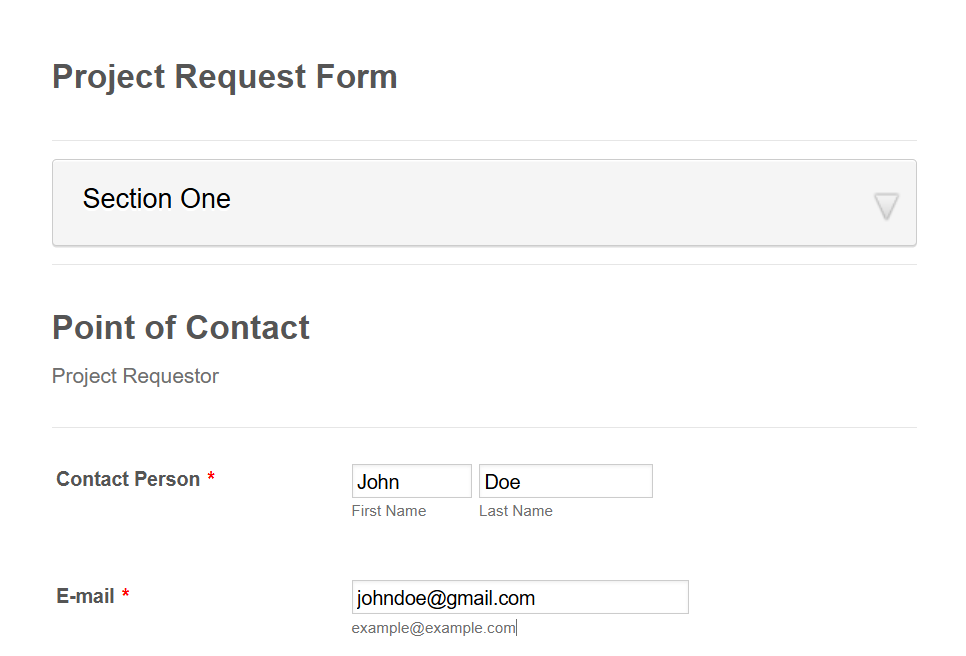
The "Security Evaluation of Cloud Services" project is designed to assist organizations in assessing the security posture of their cloud environments. With increasing reliance on cloud computing, organizations must ensure their cloud infrastructure is secure and compliant with industry standards. This project will involve developing an app that automates the process of evaluating cloud services such as AWS, Azure, and Google Cloud. The app will perform security assessments, identify potential vulnerabilities, and provide compliance checks. It will also integrate with Security Information and Event Management (SIEM) solutions to offer continuous monitoring and real-time threat detection. The goal is to deliver a comprehensive tool that simplifies cloud security assessments and helps organizations improve their security measures.

### 3.6.2 Requested Features

* **Cloud Security Assessment** – Evaluate configurations and identify vulnerabilities.
* **Compliance Check** – Assess compliance with standards like GDPR, HIPAA.
* **SIEM Integration** – Enable integration with SIEM tools for threat monitoring.
* **Automated Vulnerability Scanning** – Scan for misconfigurations and vulnerabilities.
* **Encryption Assessment** – Verify encryption for data at rest and in transit.

### 3.6.3 Main Objectives of the Project

The main objectives of the project are to evaluate the security posture of cloud environments by assessing configurations and identifying vulnerabilities, as well as ensuring compliance with industry regulations such as GDPR, HIPAA, and NIST. The project aims to integrate with SIEM systems for real-time monitoring and alerting of security events, while automating vulnerability scanning to detect misconfigurations and potential security risks. It also focuses on enhancing data protection by verifying strong encryption practices for both data at rest and in transit. Finally, the project will generate actionable insights through comprehensive reports, providing practical recommendations to improve cloud security.

**Project request form:**

A screenshot of a web page

Description automatically generated

A screenshot of a project

Description automatically generated

A screenshot of a project type

Description automatically generated

A screenshot of a contact form

Description automatically generated

## 3.7 Memo

### 3.7.1 Overview of Meeting

This meeting was held to discuss the progress of the Cloud Security Evaluation App, focusing on defining the project scope, outlining key deliverables, and addressing initial concerns related to functionality and integration with cloud platforms. The discussion also included resource allocation, timelines, and the necessary steps to ensure successful development.

### 3.7.2 Problems and Result Solutions

* **Problem:** Integration with multiple cloud providers (AWS, GCP, Azure) was unclear.
  + **Solution:** Use standardized APIs and protocols to ensure compatibility across all cloud platforms.
* **Problem:** Uncertainty around regulatory compliance.
  + **Solution:** Prioritize compliance features in the assessment phase, consult with experts for proper implementation.
* **Problem:** Limited development resources and potential budget constraints.
  + **Solution:** Leverage student accounts for cloud services to minimize initial costs and plan for additional funding if needed.
* **Problem:** Ensuring real-time security evaluations.
  + **Solution:** Implement automated, periodic scans to provide timely security assessments for users.
* **Problem:** User-friendly interface for diverse skill levels.
  + **Solution:** Focus on designing a simple, intuitive interface with easy navigation and clear feedback for non-technical users.

### 3.7.3 Result of Meeting

The team confirmed that the app can integrate with AWS, GCP, and Azure using standardized APIs and protocols. Regulatory compliance features such as GDPR and HIPAA will be prioritized in the assessment phase, with expert consultations to ensure they are correctly implemented. For the initial development phase, the team will utilize student accounts for cloud services to minimize costs, with contingency plans in place should additional funding be needed. Security evaluations will be automated and run periodically to ensure real-time, reliable results. The user interface will be designed to be simple and accessible, focusing on ease of use for all skill levels.

## 3.8 Recommendations

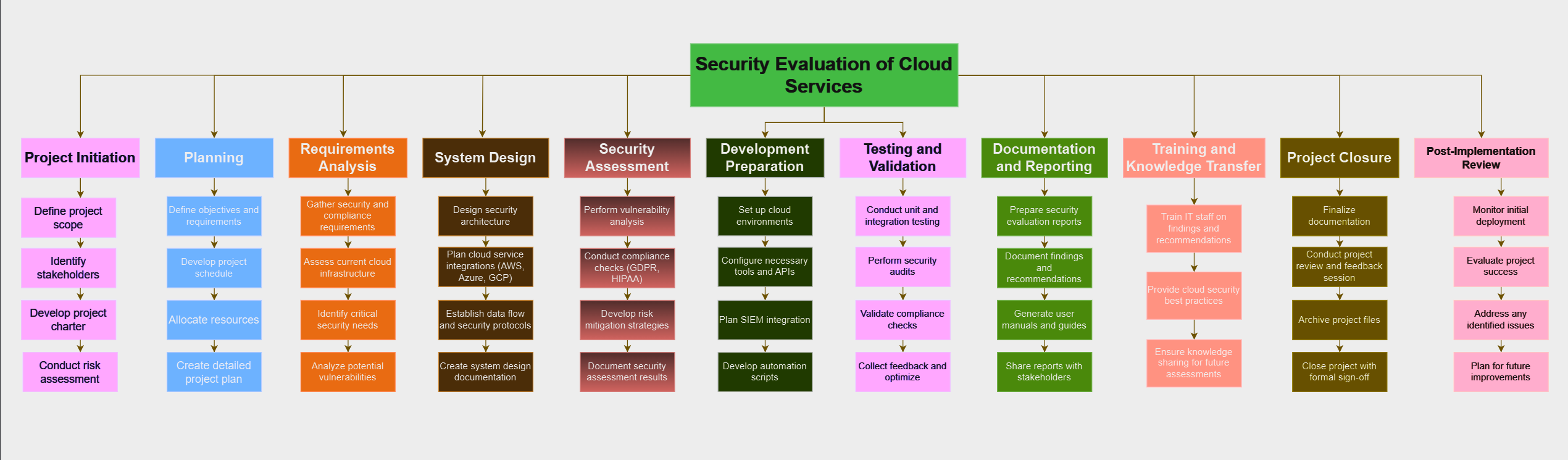
It is recommended to conduct further research on potential cloud service integrations to ensure scalability and adaptability. Expert consultations should be secured early in the process to avoid delays in implementing compliance features. A backup funding plan should be prepared in case the project extends beyond the benefits of student accounts. Clear documentation and tutorials should be developed to help users fully utilize the system’s security features. Finally, usability testing should be conducted early to gather user feedback and refine the interface for better user experience.

# Requirement Analysis

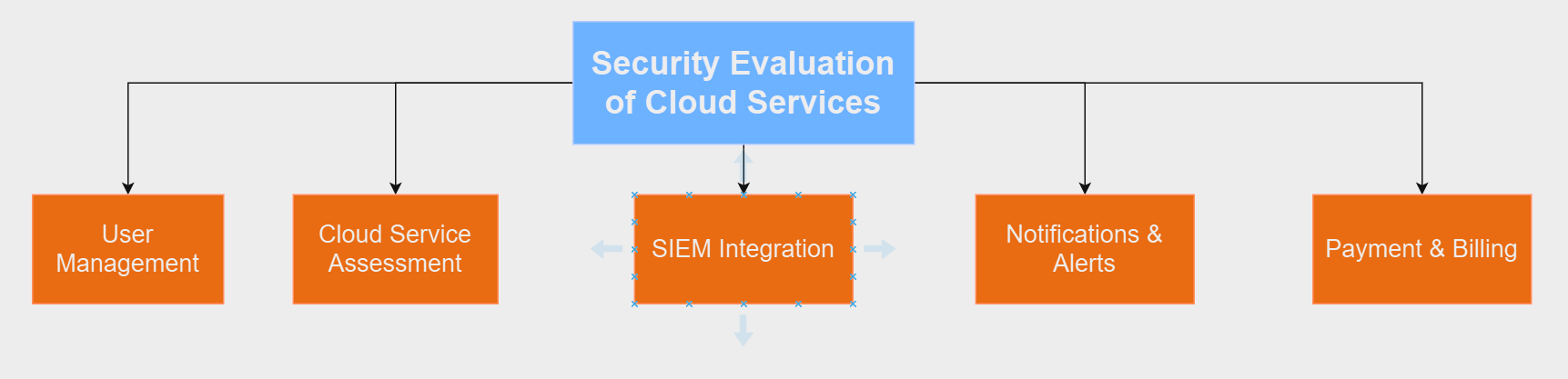
## 4.1 Work Breakdown Structure (WBS)

The Work Breakdown Structure (WBS) for our Evolution of Cloud Security is a visual, hierarchical deconstruction of the project.

**Report-based:**



**App-based:**

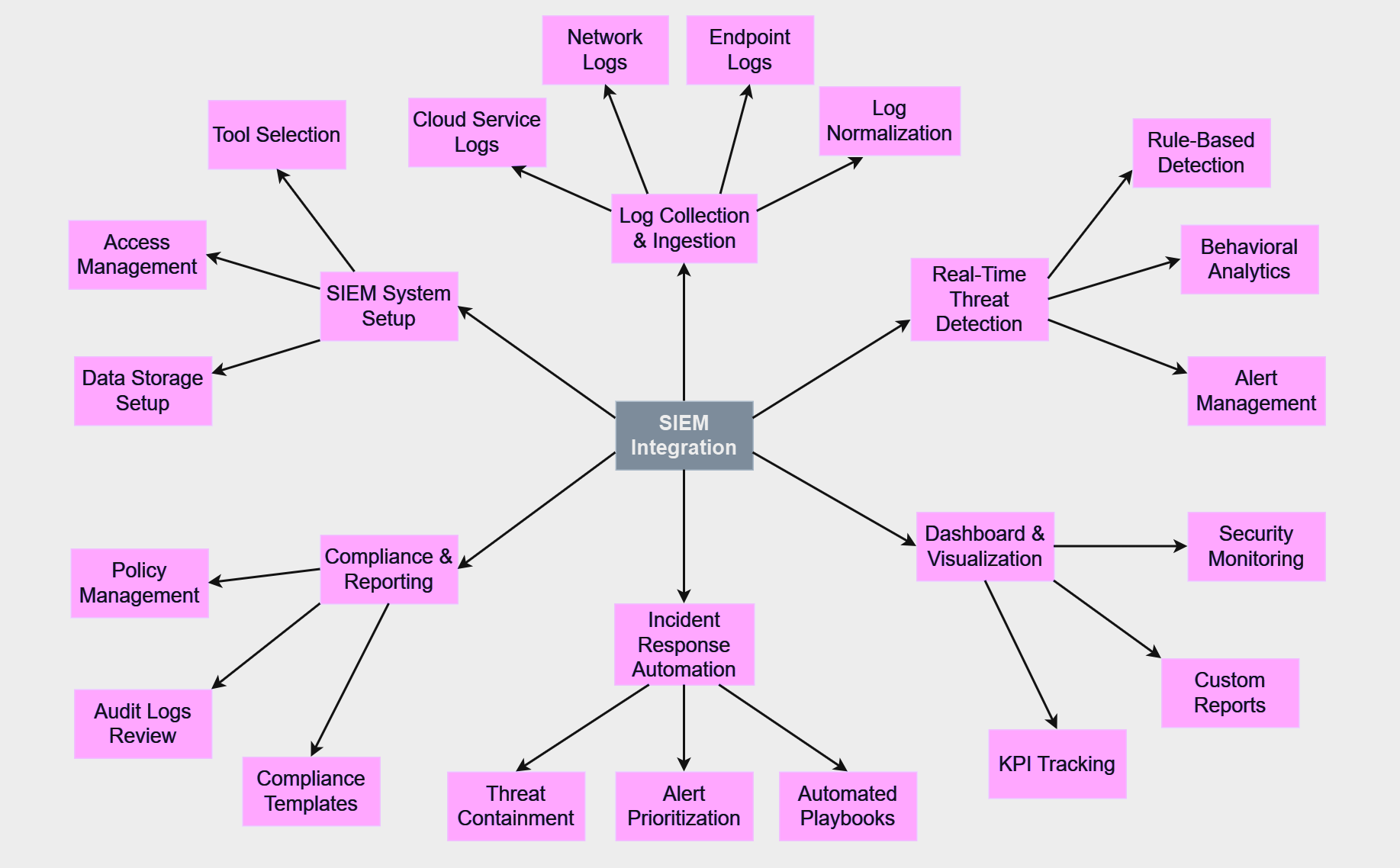


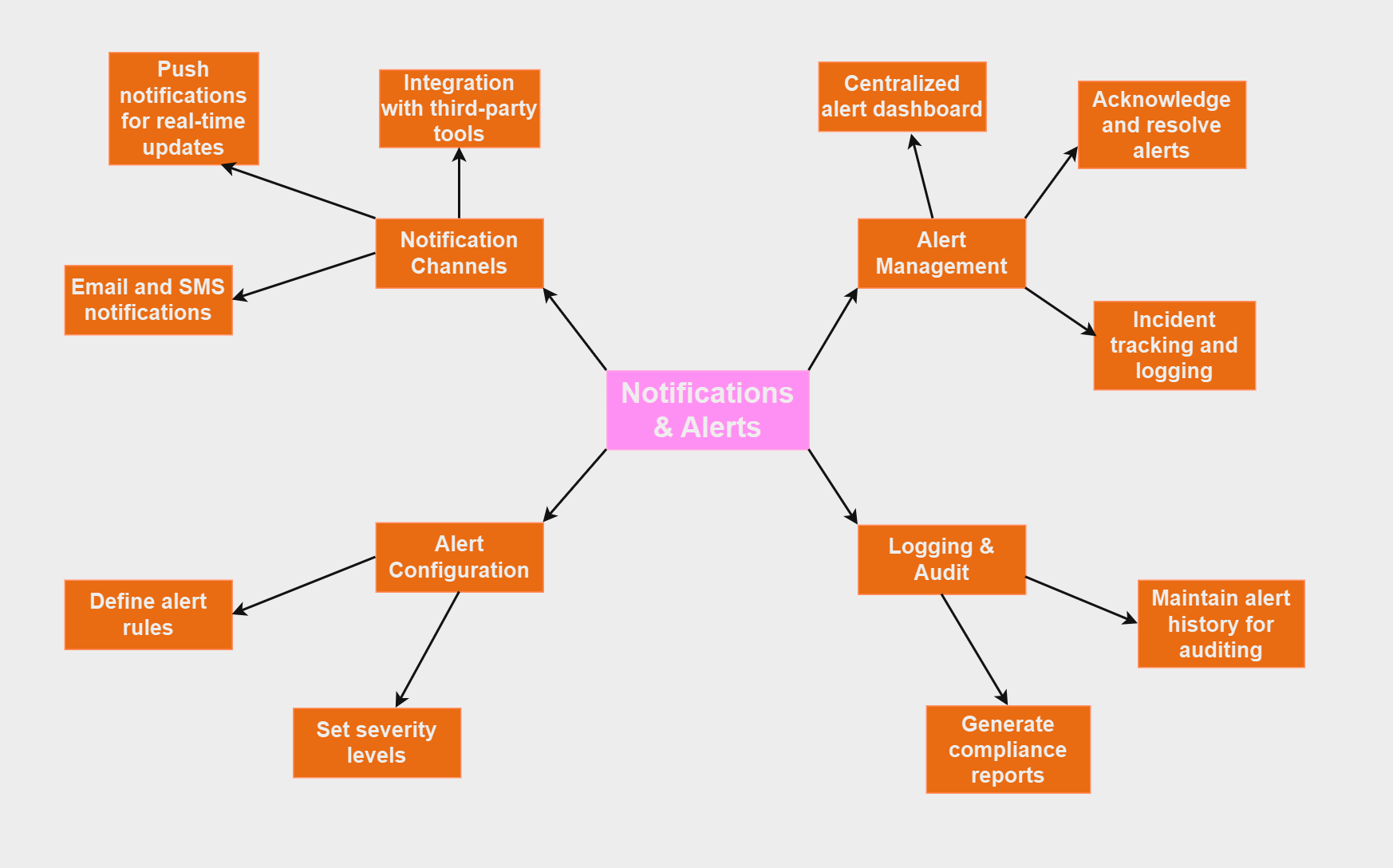
A diagram of a network

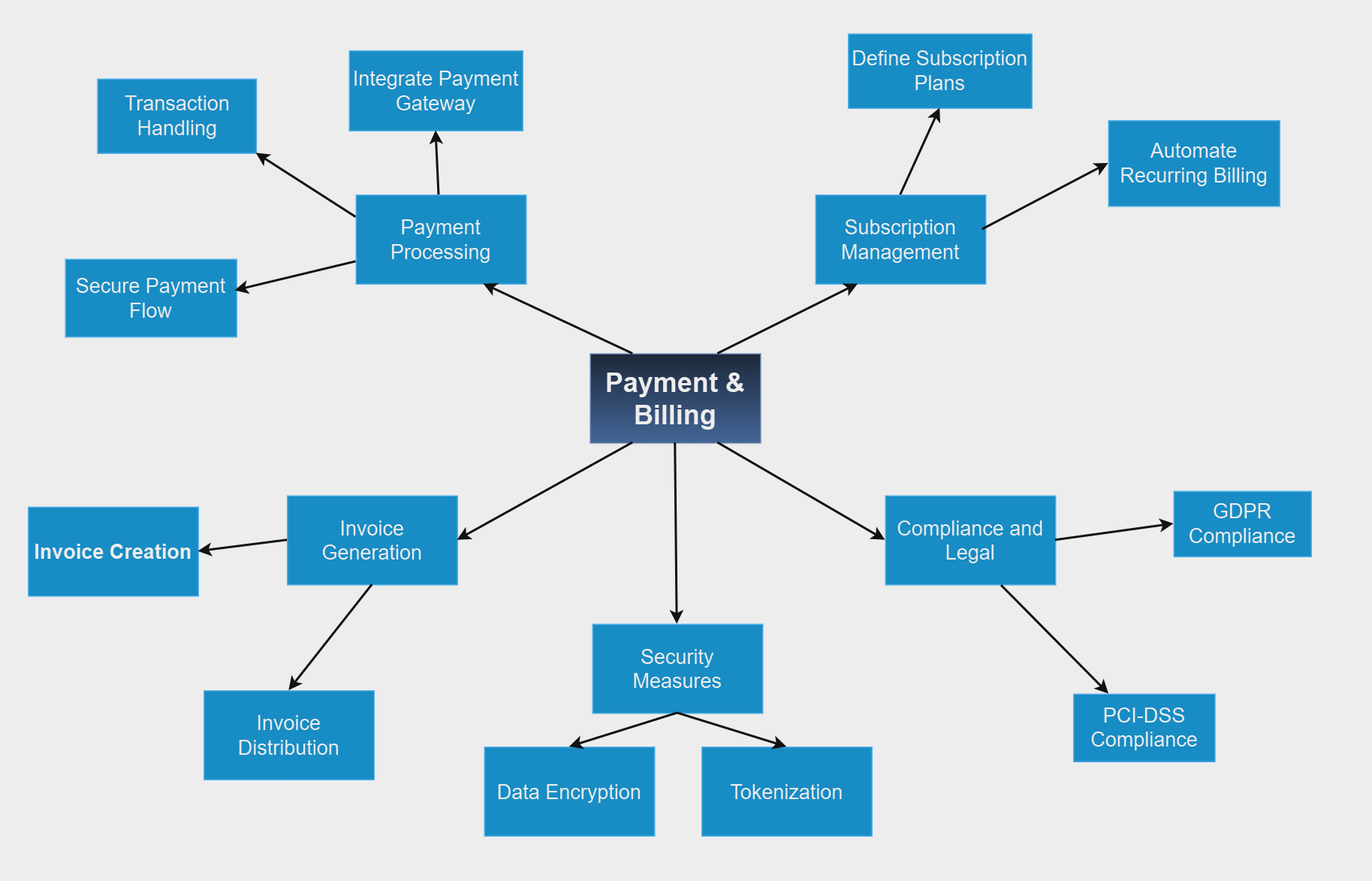
Description automatically generated

A diagram of a cloud service assessment

Description automatically generated







## 4.2 Gantt Chart

A Gantt chart is a powerful project management tool that visually represents the timeline of tasks and their dependencies within a project. It helps in organizing and scheduling tasks by showing their start and end dates, as well as the relationships between them. Each task is displayed as a horizontal bar, with the length of the bar indicating the duration. Gantt charts also allow for easy tracking of project progress, as they highlight which tasks are completed, in progress, or yet to start.

A graph of a project

Description automatically generated

A graph with a graph in the middle

Description automatically generated

A screen shot of a project

Description automatically generatedA screen shot of a graph

Description automatically generatedA graph with multiple colored lines

Description automatically generated with medium confidenceA white grid with black dots

Description automatically generated

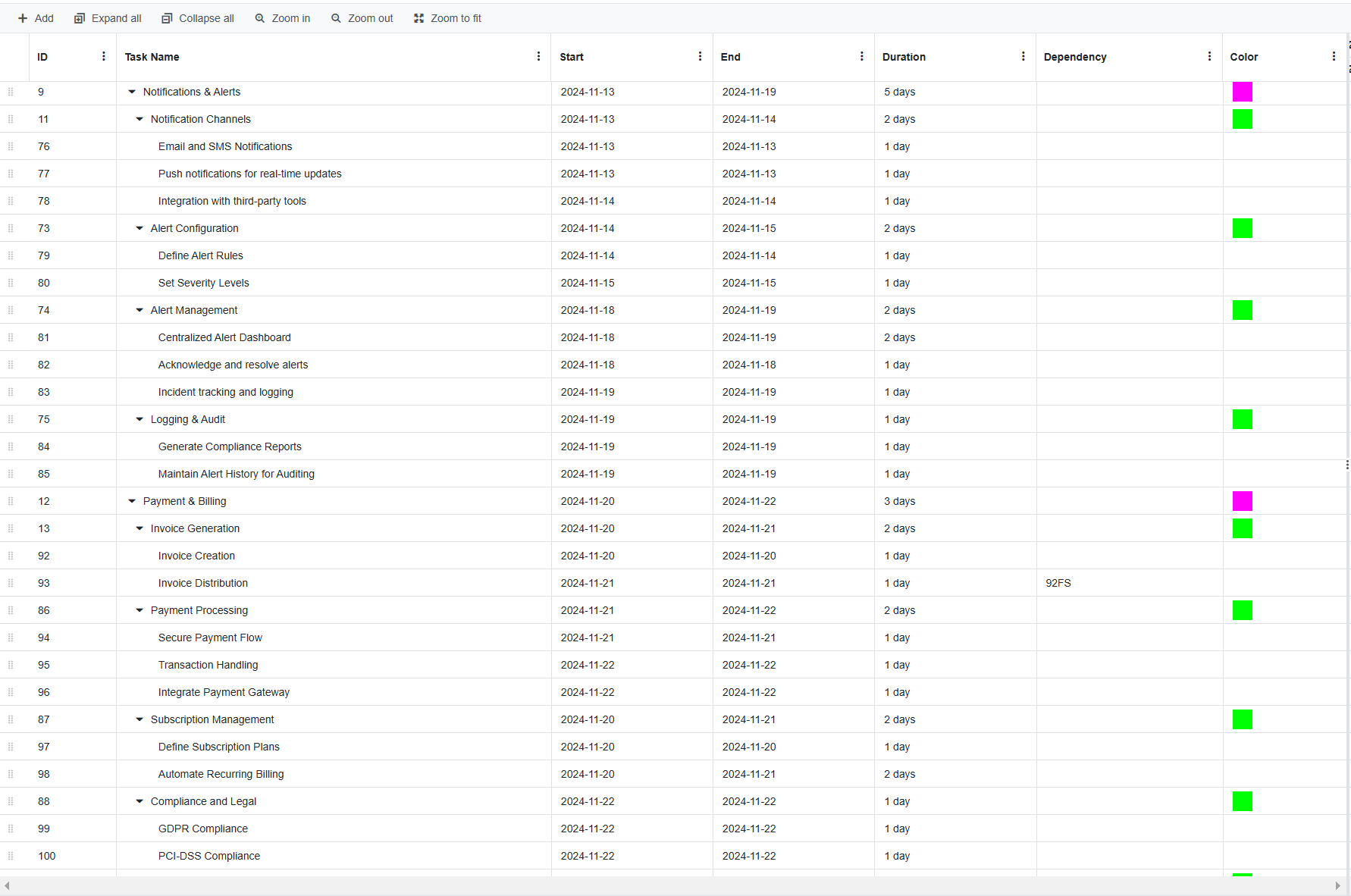
A white grid with black lines

Description automatically generated

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

## 4.3 CPM

|  |  |
| --- | --- |
| Activity | Preceding Activity |
| 1. Create Account | - |
| 2. Third-Party Registration | - |
| 3. User Login by credentials | 1, 2 |
| 4. Multi-Factor Authentication | 3 |
| 5. Profile Management | 3 |
| 6. Account Deactivation | 5 |
| 7. Cloud Environment Analysis | 6 |
| 8. Compliance Evaluation | 7 |
| 9. Security Posture Scoring | 8 |
| 10. Reporting & Recommendations | 9 |
| 11. SIEM System Setup | 10 |
| 12. Dashboard & Visualization | 11 |
| 13. Incident Response Automation | 12 |
| 14. Compliance & Reporting | 13 |
| 15. Notification Channels | 12 |
| 16. Alert Configuration | 15 |
| 17. Alert Management | 16 |
| 18. Invoice Creation | 17 |
| 19. Invoice Distribution | 18 |
| 20. Integrate Payment Gateway | 17 |
| 21. Automate Recurring Billing | 16, 20 |

A diagram of a company

Description automatically generated

## 4.4 PERT Chart

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Activity | Estimated Time (n days) | | | Expected Time  (o + 4r + p) / 6 |
| **o** | **r** | **p** |
| 1. Create Account | 1 | 2 | 3 | 3 |
| 2. Third-Party Registration | 1 | 3 | 5 | 1 |
| 3. User Login by credentials | 1 | 1 | 2 | 3 |
| 4. Multi-Factor Authentication | 2 | 3 | 5 | 3 |
| 5. Profile Management | 2 | 3 | 4 | 3 |
| 6. Account Deactivation | 2 | 3 | 4 | 3 |
| 7. Cloud Environment Analysis | 1 | 3 | 5 | 2 |
| 8. Compliance Evaluation | 1 | 2 | 3 | 2 |
| 9. Security Posture Scoring | 1 | 2 | 3 | 3 |
| 10. Reporting & Recommendations | 1 | 3 | 5 | 4 |
| 11. SIEM System Setup | 2 | 4 | 7 | 4 |
| 12. Dashboard & Visualization | 2 | 4 | 6 | 3 |
| 13. Incident Response Automation | 2 | 3 | 4 | 2 |
| 14. Compliance & Reporting | 1 | 2 | 3 | 2 |
| 15. Notification Channels | 1 | 2 | 4 | 2 |
| 16. Alert Configuration | 1 | 2 | 3 | 2 |
| 17. Alert Management | 1 | 2 | 3 | 1 |
| 18. Invoice Creation | 1 | 1 | 2 | 1 |
| 19. Invoice Distribution | 1 | 1 | 2 | 2 |
| 20. Integrate Payment Gateway | 1 | 2 | 4 | 2 |
| 21. Automate Recurring Billing | 1 | 2 | 3 | 3 |

## 4.5 List of requirements

In this section, lists of user and system requirements are provided:

### 4.5.1 User Requirements

1. **Account Creation and Management:**
   * Users must be able to register by providing basic information (email, phone number, username, password).
   * Users will receive an email confirmation to verify their account.
   * Users can reset passwords using a secure process involving email verification.
   * Account deletion must be possible, ensuring data removal in compliance with regulations like GDPR.
2. **Login and Security:**
   * Users can log in with email/password or third-party OAuth (e.g., Google, GitHub).
   * Users can enable Multi-Factor Authentication (MFA) for additional security.
   * The system sends a One-Time Password (OTP) for verification during critical actions or on new devices.
3. **User Dashboard and Profile Management:**
   * Users can update personal details and change passwords after verifying their identity.
   * A dashboard allows easy access to security settings and account info.
4. **Cloud Security Monitoring:**
   * Users should be able to monitor their cloud security posture with periodic reports on vulnerabilities, misconfigurations, and threats.
   * The system must alert users about detected risks and provide recommendations.
5. **Compliance and Auditing:**
   * The system should assess cloud services for compliance with standards like GDPR and HIPAA.
   * Audit logs must be available for tracking user actions and cloud interactions for regulatory purposes.

### 4.5.2 System Requirements

#### 4.5.2.1Functional Requirements

1. **User Registration and Login:**
   * Email and phone verification during registration.
   * Third-party authentication (OAuth) and secure password hashing (e.g., bcrypt).
   * OTP verification for sensitive actions like login from new devices.
2. **MFA and OTP:**
   * Support for enabling MFA (e.g., using Google Authenticator).
   * OTP should be sent for verification, with expiration and a limited time window for entry.
3. **Cloud Security Monitoring:**
   * Integrate with cloud services (e.g., AWS, Azure) to collect security configurations and detect vulnerabilities.
   * Generate security reports highlighting misconfigurations, risks, and compliance gaps.
   * Real-time security alerts for threats detected through SIEM integration.
4. **Compliance and Auditing:**
   * The system must provide compliance monitoring with standards like GDPR, HIPAA, and PCI-DSS.
   * Generate audit logs for every user activity and cloud interaction for traceability and audits.
5. **SIEM Integration:**
   * Collect logs from user activities, cloud events, and API calls to detect and alert on suspicious behavior.
   * Anomaly detection in cloud usage or login patterns to identify potential threats.
   * Provide incident response automation (e.g., disabling accounts after multiple failed login attempts).

#### 4.5.2.2 Non-Functional Requirements

1. **Security:**
   * Encrypt sensitive data both in transit (e.g., TLS) and at rest (e.g., AES-256).
   * Implement secure authentication mechanisms (password hashing, MFA) and integrate with SIEM for real-time monitoring.
2. **Performance and Scalability:**
   * The system should be able to handle a large number of concurrent users without performance degradation.
   * Scalable architecture to support future growth of users, cloud services, and security features.
3. **Reliability and Availability:**
   * Achieve 99.9% uptime with high availability and failover mechanisms in place.
   * Ensure reliable data backup and recovery processes.
4. **Compliance:**
   * Ensure full GDPR and HIPAA compliance in handling user data.
   * Enable data deletion and retention features to comply with privacy regulations.
5. **Logging and Monitoring:**
   * The system must provide comprehensive logging for all user actions and cloud interactions to ensure security and support compliance audits.
   * Integrate with a SIEM platform to analyze logs and trigger real-time alerts for security events.

# 5. Agile parts

## 5.1 Product Backlogs

|  |  |
| --- | --- |
| Product Backlog | User Stories |
| User Authentication and Security  Priority: High  Duration: 20 days | Send a verification email with a confirmation link to activate the account |
| Enable logging in with a user-provided email/password pair. |
| Enable 2FA as an optional security feature in user settings |
| Send real-time email/SMS alerts for suspicious actions |
| Provide a "Forgot Password" link on the login page |
| Enforce strong password rules |
| Implement an automatic account lock |
| Implement a notification system that sends alerts immediately upon detecting risks |
| User Profile and Management  Priority: High  Duration: 15 days | Implement a feature allowing users to upload profile pictures |
| Develop a role-based access control (RBAC) system where admins can assign roles |
| Implement a feature to log all user activities |
| Create a subscription management module |
| Implement automatic session expiration |
| Cloud Service Integration  Priority: High  Duration: 7 days | Develop a dashboard to display real-time statuses of cloud service integrations, including their security health (active, inactive, or error). |
| Cloud Security Posture Assessment  Priority: High  Duration: 10 days | Build a feature to conduct security assessments of connected cloud services, providing detailed reports with risk scores and improvement suggestions. |
| Implement a comprehensive audit log system to track security-related events |
| Cloud Service Risk Remediation  Priority: Medium  Duration: 5 days | Integrate automated security scans to detect configuration risks like open ports or weak encryption in cloud services |

## 5.2 User stories

The following user stories outline key features and functionalities of the system, focusing on improving user experience and operational efficiency. These stories address crucial components such as user registration, login flows, account management, integration capabilities, and system security. Each user story is accompanied by specific scenarios that detail user interactions, potential error conditions, and expected system responses. These stories serve as a foundation for both development and testing, ensuring that the final product aligns with user needs and operational requirements.

**1. User Registration with Email Verification**

**U.R.1.1: As a new user, I want to register with my email, and I expect to verify it through a confirmation link.**

* **Scenario 1:** User provides a valid email and password, receives an email with a verification link, and successfully activates their account.
* **Scenario 2:** User provides an invalid email or fails to verify within the given time frame, resulting in an error or account deactivation.

**2. Login with Email and Password**

**U.R.2.1: As a registered user, I want to be able to log in using my email and password.**

* **Scenario 1**: The user successfully logs in after entering the correct credentials.
* **Scenario 2**: Incorrectly entered credentials result in an authentication problem for the user.

**3. Login with Two-Factor Authentication (2FA)**

**U.R.3.1: As a user, I want to enable two-factor authentication (2FA) to add an extra layer of security to my account.**

* **Scenario 1:** User enables 2FA, receives a code on their phone, and successfully logs in after entering the code.
* **Scenario 2:** User attempts to log in without entering the correct 2FA code and is denied access.

**4. Profile Picture Upload**

**U.R.4.1: As a user, I want the ability to upload a profile picture to personalize my account.**

* **Scenario 1:** User selects and uploads a valid image file, and the profile picture is updated successfully.
* **Scenario 2:** User attempts to upload an unsupported file type or a file larger than the maximum size, resulting in an error message.

**5. User Roles and Permissions**

**U.R.5.1: As an admin, I want to assign different roles (e.g., user, admin) to manage the system’s access control.**

* **Scenario 1:** Admin successfully assigns the "user" role to a new member.
* **Scenario 2:** Admin attempts to assign the "admin" role to a non-admin user, but an error message prevents it.

**6. Account Security Alerts**

**U.R.6.1: As a user, I want to receive real-time security alerts if suspicious activity is detected on my account.**

* **Scenario 1:** User receives an alert for an unusual login attempt from a different geographical location.
* **Scenario 2:** User receives an email alert for a successful password change and has the option to disable it if it wasn’t initiated by them.

**7. Activity Log**

**U.R.7.1: As a user, I want to access my activity log to review all past login attempts and changes made to my account.**

* **Scenario 1:** User views a detailed activity log, including timestamps for each login, logout, and account change.
* **Scenario 2:** User tries to access the activity log but sees an error if the system cannot fetch the log due to an internal issue.

**8. Subscription Management**

**U.R.8.1: As a user, I want to manage my subscription plan directly from my dashboard.**

* **Scenario 1:** User successfully upgrades from a basic to a premium plan via their dashboard.
* **Scenario 2:** User tries to downgrade to a plan that’s incompatible with their current settings, and the system prompts a warning.

**9. Session Timeout and Auto-Logout**

**U.R.9.1: As a user, I want the system to automatically log me out after a certain period of inactivity.**

* **Scenario 1:** User is logged out automatically after 30 minutes of inactivity, and they are presented with a login screen.
* **Scenario 2:** User attempts to perform an action after being logged out, and the system prompts to log in again before proceeding.

**10. Cloud Service Integration Monitoring**

**U.R.10.1: As a cloud security administrator, I want to monitor the security integration of various cloud services within the system.**

* **Scenario 1:** The system displays the status of cloud service integrations (active, inactive, or error) and their security health.
* **Scenario 2:** A cloud service integration faces an issue, and the system generates an alert for the administrator to address the problem.

**11. Forgot Password**

**U.R.11.1:** **As a user, I want the ability to reset my password if I forget it, so that I can regain access to my account securely.**

* **Scenario 1:** User clicks on the "Forgot Password" link, receives a password reset email, and successfully sets a new password.
* **Scenario 2:** User encounters an issue with the password reset process and contacts support for assistance.

**12. Password Complexity Requirements**

**U.R.12.1: As a user, I want the system to enforce strong password policies for my account's security.**

* **Scenario 1:** User creates a password with at least 8 characters, including one special character, one uppercase letter, and one number, and the system accepts it.
* **Scenario 2:** User tries to create a password that does not meet the complexity requirements, and the system displays an error message.

**13. Cloud Security Posture Assessment**

**U.R.13.1: As a cloud administrator, I want the system to assess the security posture of connected cloud services and provide a report.**

* **Scenario 1:** User receives a security posture report with an overall score, identifying potential risks and recommended improvements.
* **Scenario 2:** User attempts to generate a report, but the system encounters a configuration error and prompts to check settings.

**14. Account Locking after Suspicious Activity**

**U.R.14.1: As a user, I want my account to be locked automatically after multiple failed login attempts to protect my account from brute-force attacks.**

* **Scenario 1:** After 5 failed login attempts, the system locks the account and sends the user an email notifying them of the lockout.
* **Scenario 2:** User attempts to log in after the account lockout, and the system displays a message stating the account is locked for security reasons.

**15. Audit Log for Cloud Security Events**

**U.R.15.1: As a security auditor, I want to access an audit log of all security-related events within the system for compliance and review.**

* **Scenario 1:** User successfully views a log of security events such as login failures, account changes, and SIEM alerts.
* **Scenario 2:** User attempts to access the audit log but is denied access due to insufficient privileges, and the system displays an error message.

**16. Real-time Alert Configuration for Security Risks**

**U.R.16.1: As a security administrator, I want to configure real-time alerts for specific cloud security risks such as unauthorized access attempts.**

* **Scenario 1:** User sets up real-time alerts for failed login attempts from unrecognized IP addresses, and the system sends an immediate notification.
* **Scenario 2:** User configures an alert but receives a delayed notification due to a system delay or misconfiguration.

**17. Cloud Service Risk Detection and Remediation**

**U.R.17.1: As a security engineer, I want the system to automatically detect cloud service configuration risks (e.g., open ports, weak encryption) and recommend remediation actions.**

* **Scenario 1:** The system detects an open port in a cloud service and automatically suggests a fix to close it.
* **Scenario 2:** The system identifies weak encryption in cloud storage and suggests implementing stronger encryption protocols.

# UML Diagram

## 6.1 Context Model Diagram

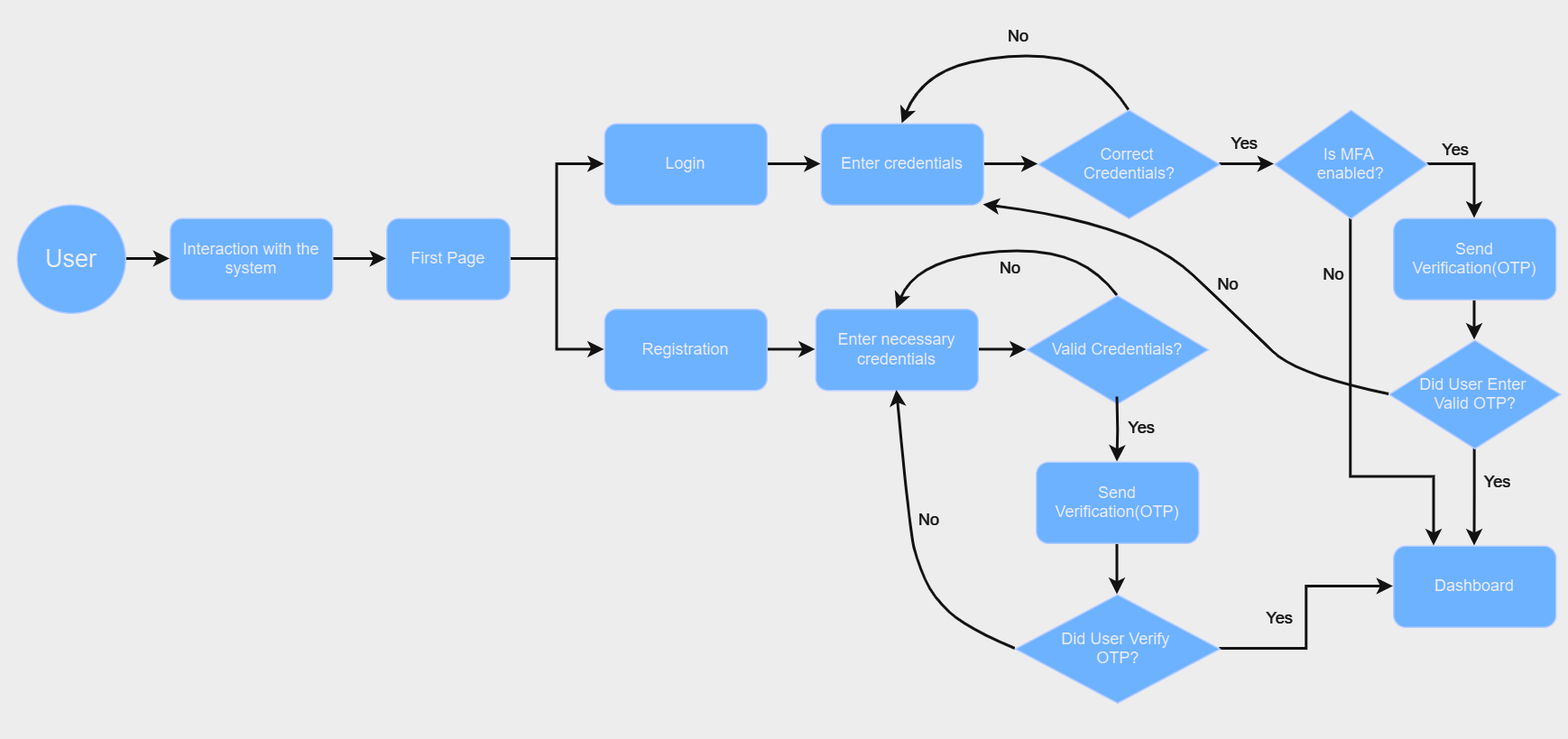
Project is divided into 5 principal portions which are User Management System, Cloud Service Assessment System, SIEM Integration System, Payment & Billing System, Notifications & Alerts System. All then will be explained in detail in the following.

A diagram of a cloud security system

Description automatically generated

## 6.2 Activity Diagram

An Activity Diagram is a type of UML diagram that represents the flow of activities or tasks in a process. Here, our activity diagram of User Management is provided:



## 6.3 User Case Diagram

A diagram of a network

Description automatically generated

## 6.4 Sequence Diagrams

This sequence diagram illustrates the process of a user registering on the platform and verifying their email. It ensures that the user’s details are correctly captured and validated before granting access to the platform.

**A diagram of a user registration

Description automatically generated**

The next sequence diagram shows the login process for a user, including OTP verification and multi-factor authentication (MFA). It ensures that only authorized users can access their accounts, adding extra layers of security for sensitive operations.

**A diagram of a user login

Description automatically generated**

This sequence diagram demonstrates how the system monitors cloud services for security vulnerabilities and generates a report. It helps organizations ensure their cloud configurations are secure and compliant by detecting potential threats.

**A diagram of a cloud security monitoring and report generation

Description automatically generated**

This sequence diagram outlines the process of a compliance audit, where the system checks the logs against industry standards (e.g., GDPR, HIPAA). It provides an audit trail for regulatory compliance and allows organizations to demonstrate their adherence to security standards.

**A diagram of a company's compliance report

Description automatically generated**

This sequence diagram shows how users select a subscription plan and complete the payment process. It handles the interaction with the payment gateway and updates the user’s subscription status, ensuring smooth billing and access management.

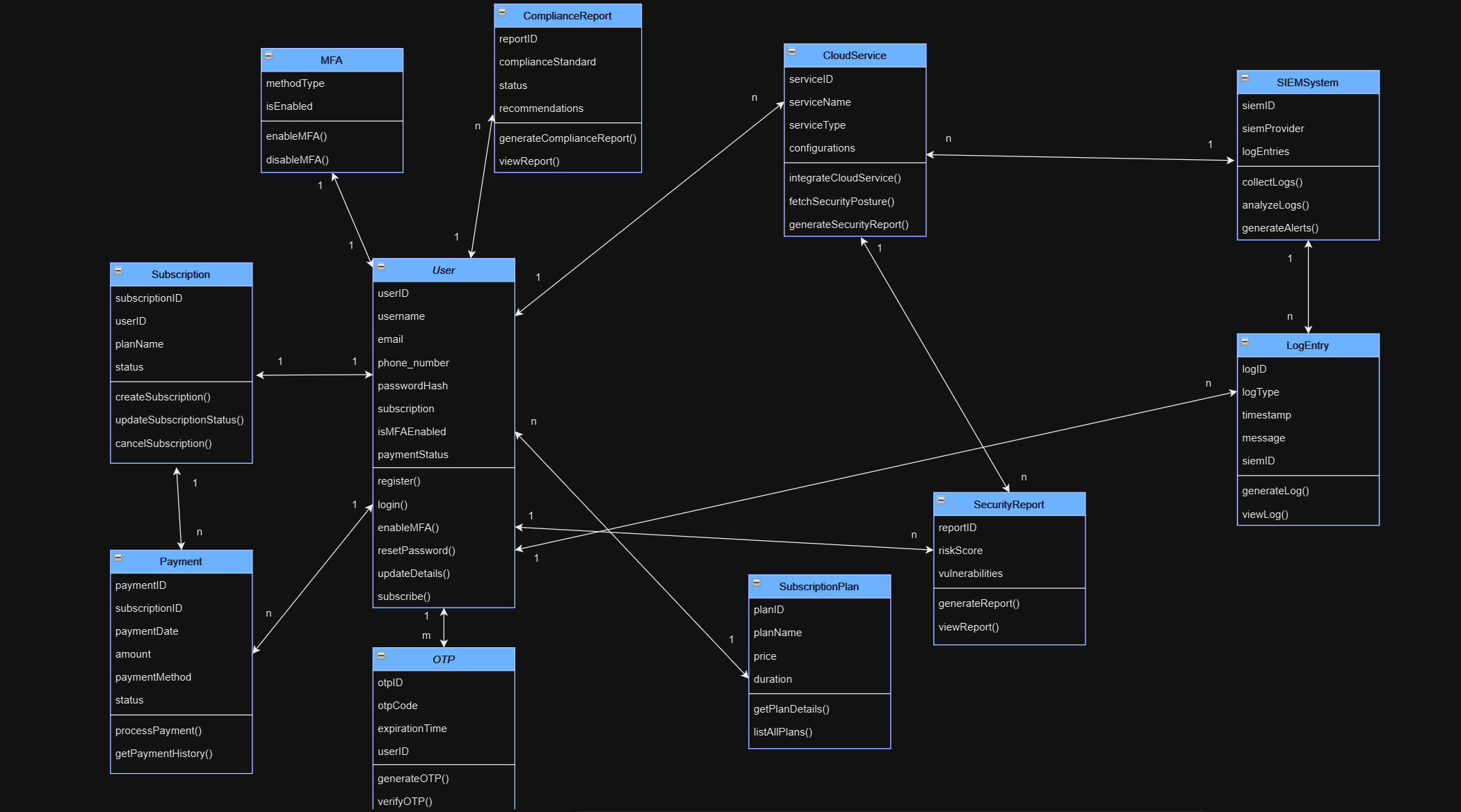
**A diagram of a diagram

Description automatically generated**

## 6.5 Class Diagram

Here, we provide class diagram which represent a cloud security evaluation system. It depicts entities like users, accounts, subscriptions, compliance and security reports, cloud services, security measures, etc.

The diagram shows relationships between these entities, such as one-to-one, one-to-many, and many-to-many relationships. This system manages user accounts, generates reports, integrates with cloud services, and implements security measures like MFA and OTP.

****

## 6.6 ER Diagram

The provided ER diagram depicts all the entities and relationships in the project.

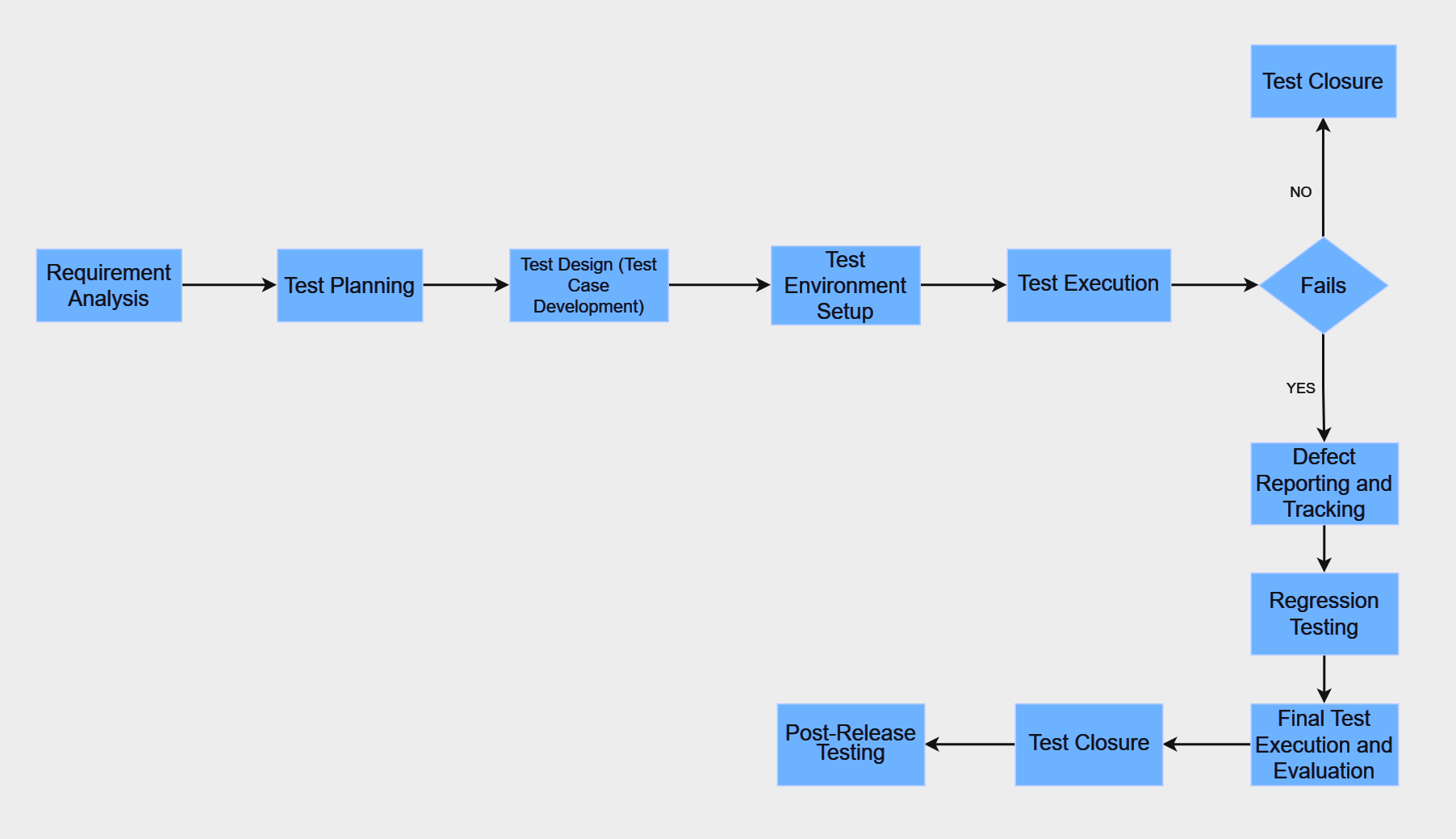
**A black screen with white dots

Description automatically generated**

# Test Strategy Documentation

## 7.1. Test Life Cycle

The Software Testing Life Cycle (STLC) is a structured approach to testing, which ensures software quality and reliability. It is a part of the broader concept - Test Life Cycle (TLC) and focuses specifically on systematically verifying and validating the software against defined requirements.



There are 10 main phases of the STLCwhich will be applied to our Cloud Security Evaluation App project. Here are the main phases of the STLC:

* **Requirement Analysis**: This phase involves a comprehensive review of the security and compliance requirements associated with cloud environments. It includes the identification of key security features to be evaluated, such as access controls, encryption standards, vulnerabilities, and compliance with regulations like GDPR, HIPAA, etc. Additionally, integration points with SIEM solutions for continuous monitoring and alerting are identified.
* **Test Planning**: During this phase, a detailed testing strategy is developed. The focus is to determine the scope of testing, required resources, timelines, and methodologies which are necessary to ensure thorough testing of security vulnerabilities, regulatory compliance, and compatibility with cloud platforms. The tools and techniques for testing, such as penetration testing, vulnerability scanning, and automated compliance checks, are also specified.
* **Test Design (Test Case Development)**: Test cases are designed to cover a variety of security scenarios. These include testing the encryption of data in transit and at rest, evaluating authentication mechanisms (such as MFA and OTP), SIEM integration, and conducting vulnerability scans. Test cases are developed to validate real-time threat detection and response capabilities, as well as compliance assessments. These test cases are aligned with the functional and security requirements of the app.
* **Test Environment Setup**: A test environment is established to simulate real-world cloud environments. The app is tested on various cloud platforms, ensuring secure monitoring of cloud services and effective integration with SIEM solutions. The environment is configured to assess user data encryption, vulnerability scanning, and compliance auditing under different conditions.
* **Test Execution**: In this phase, the designed test cases are executed. Security tests are conducted to verify the app's ability to identify vulnerabilities, misconfigurations, and compliance gaps in cloud environments. The integration with SIEM platforms is also tested to confirm its effectiveness in detecting and responding to security threats. Logs are reviewed to ensure that security events and incidents are accurately tracked.
* **Defect Reporting and Tracking**: Any defects identified during test execution, such as issues with misconfigurations, inaccurate compliance reports, or failed SIEM integrations, are logged, prioritized, and tracked. Defects are reported to the development team for resolution. The testing team verifies that the fixes address the issues without introducing new ones.
* **Regression Testing**: After defects are fixed, regression testing is performed to ensure that the changes have not impacted the security or functionality of the app in other areas.
* **Final Test Execution and Evaluation**: The final round of testing is carried out to verify that all test cases pass and that the app is secure, compliant, and functional across all supported cloud platforms. It also ensures that the app integrates effectively with SIEM tools. This phase includes validation of the disaster recovery plans, business continuity features, and user authentication mechanisms.
* **Test Closure**: Once all tests are completed, the test results are analyzed, and a test closure report is prepared. This report includes detailed findings, lessons learned, and any unresolved issues. It formally concludes the testing process, and all relevant documents, including test cases, logs, and reports, are archived for future reference.
* **Post-Release Testing**: After the app's release, post-release testing is performed to ensure that the app performs as expected in the production environment. This phase includes monitoring cloud services for unexpected security issues, validating the performance and scalability of the app, and ensuring that it continues to meet the required compliance and security standards.

## 7.2. Integration Test

**Use cases:**

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **Title** | **Description** |
| **UC-001** | **User Registration** | Verify that a new user can register by providing necessary information. |
| **UC-002** | **User Login** | Confirm that a registered user can log in using their credentials. |
| **UC-003** | **Password Reset** | Verify that a user can reset their password via an email link when they forget their password. |
| **UC-004** | **User Logout** | Evaluate that a logged-in user can log out of the application to end their session. |
| **UC-005** | **Multi-Factor Authentication (MFA) Setup** | Verify that a user can enable MFA to add an additional layer of security during login. |
| **UC-006** | **Profile Update** | Test that users can update their personal details, including email, password, or contact information. |
| **UC-007** | **Cloud Service Registration** | Verify that users can register their cloud services for monitoring and security evaluation. |
| **UC-008** | **Security Configuration Check** | Confirm that the app can check the security configurations of the user’s cloud services for vulnerabilities and misconfigurations. |
| **UC-009** | **Compliance Check** | Verify that the app can assess the cloud service's compliance with regulatory standards such as GDPR and HIPAA. |
| **UC-010** | **Vulnerability Scanning** | Verify that the app can scan the cloud environment for known vulnerabilities, outdated software, and misconfigurations. |
| **UC-011** | **Real-time Security Alerts** | Test that the app provides real-time alerts when potential security threats or breaches are detected. |
| **UC-012** | **SIEM Integration** | Verify that the app integrates with a SIEM platform to collect and analyze security logs from cloud environments. |
| **UC-013** | **Access Control Check** | Test that the app can check the cloud service’s access control policies to ensure proper configuration. |
| **UC-014** | **Incident Response** | Confirm that the app provides guidance for users on how to respond to security incidents such as unauthorized access. |
| **UC-015** | **Security Report Generation** | Verify that users can generate detailed security reports, including vulnerabilities, risks, and compliance status of their cloud environments. |
| **UC-016** | **Data Encryption Check** | Test that the app can check if encryption standards are applied to data in transit and at rest. |
| **UC-017** | **Audit Log Management** | Verify that the app maintains comprehensive logs of user actions and cloud interactions for auditing purposes. |
| **UC-018** | **Cloud Resource Monitoring** | Test that the app continuously monitors cloud services for security risks, misconfigurations, and potential threats. |
| **UC-019** | **Security Dashboard** | Test that the user dashboard provides an overview of the cloud environment's security status, incidents, and recommended actions for improvement. |

**Test Cases:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Use Case ID** | **Test Case ID** | **Test Case Title** | **Test Case Description** | **Prerequisites for Test Case Execution** | **Execution Steps** | **Expected Results** | **Result** |
| UC-001 | TC-001 | User Registration | Verify that a new user can register by providing necessary information. | User is on the registration page. | 1. Navigate to the registration page. 2. Enter required details. 3. Submit the form. | User account is created and a confirmation email is sent. | Pass |
| UC-002 | TC-002 | User Login | Confirm that a registered user can log in using their credentials. | User has a registered account. | 1. Navigate to the login page. 2. Enter valid credentials. 3. Click login. | User is successfully logged in and redirected to the dashboard. | Pass |
| UC-003 | TC-003 | Password Reset | Verify that a user can reset their password via an email link when they forget their password. | User has a registered email address. | 1. Navigate to the login page. 2. Click on "Forgot Password". 3. Enter registered email. 4. Follow the email link to reset the password. | User successfully resets the password and can log in with the new one. | Pass |
| UC-004 | TC-004 | User Logout | Evaluate that a logged-in user can log out of the application to end their session. | User is logged in. | 1. Click the logout button. 2. Confirm logout. | User is logged out and redirected to the login page. | Pass |
| UC-005 | TC-005 | MFA Setup | Verify that a user can enable MFA to add an additional layer of security during login. | User is logged in and on the security settings page. | 1. Navigate to the security settings. 2. Select MFA option. 3. Follow the MFA setup steps. | MFA is enabled and user can log in using both password and authentication code. | Pass |
| UC-006 | TC-006 | Profile Update | Test that users can update their personal details, including email, password, or contact information. | User is logged in. | 1. Navigate to the profile page. 2. Modify profile details (email, password, contact). 3. Save changes. | User's profile is updated successfully. | Pass |
| UC-007 | TC-007 | Cloud Service Registration | Verify that users can register their cloud services for monitoring and security evaluation. | User is logged in. | 1. Navigate to cloud service registration page. 2. Enter cloud service details. 3. Submit the registration form. | Cloud service is successfully registered and ready for monitoring. | Pass |
| UC-008 | TC-008 | Security Configuration Check | Confirm that the app can check the security configurations of the user’s cloud services for vulnerabilities and misconfigurations. | Cloud service is registered. | 1. Navigate to the security configuration section. 2. Run the security configuration check. 3. Review the results. | Vulnerabilities or misconfigurations are identified and listed. | Pass |
| UC-009 | TC-009 | Compliance Check | Verify that the app can assess the cloud service's compliance with regulatory standards such as GDPR and HIPAA. | Cloud service is registered. | 1. Navigate to the compliance check section. 2. Select relevant compliance standards. 3. Run the compliance check. | Compliance status for each standard (GDPR, HIPAA) is displayed. | Pass |
| UC-010 | TC-010 | Vulnerability Scanning | Verify that the app can scan the cloud environment for known vulnerabilities, outdated software, and misconfigurations. | Cloud service is registered. | 1. Navigate to vulnerability scanning section. 2. Run the scan. 3. Review the identified vulnerabilities. | Vulnerabilities, outdated software, and misconfigurations are identified. | Pass |
| UC-011 | TC-011 | Real-time Security Alerts | Test that the app provides real-time alerts when potential security threats or breaches are detected. | Cloud service is registered and monitored. | 1. Simulate a security threat (e.g., unauthorized login attempt). 2. Check for alert notifications. | Real-time alerts for the security threat are received. | Pass |
| UC-012 | TC-012 | SIEM Integration | Verify that the app integrates with a SIEM platform to collect and analyze security logs from cloud environments. | SIEM platform is configured. | 1. Log in to the app. 2. Navigate to SIEM integration settings. 3. Verify that the SIEM system is collecting and analyzing logs. | Security logs from the cloud environment are displayed and analyzed in the SIEM platform. | Pass |
| UC-013 | TC-013 | Access Control Check | Test that the app can check the cloud service’s access control policies to ensure proper configuration. | Cloud service is registered. | 1. Navigate to the access control section. 2. Run the access control check. 3. Review the results. | Access control misconfigurations are identified and displayed. | Pass |
| UC-014 | TC-014 | Incident Response | Confirm that the app provides guidance for users on how to respond to security incidents such as unauthorized access. | User is logged in and incident response guidance is enabled. | 1. Simulate an incident (e.g., unauthorized access). 2. Review the guidance provided. | Appropriate incident response guidance is provided. | Pass |
| UC-015 | TC-015 | Security Report Generation | Verify that users can generate detailed security reports, including vulnerabilities, risks, and compliance status of their cloud environments. | Cloud services are monitored. | 1. Navigate to the report generation section. 2. Select desired parameters (vulnerabilities, risks, compliance). 3. Generate the report. | A detailed report is generated with vulnerabilities, risks, and compliance status. | Pass |
| UC-016 | TC-016 | Data Encryption Check | Test that the app can check if encryption standards are applied to data in transit and at rest. | Cloud service is registered. | 1. Navigate to the encryption check section. 2. Run the encryption check. 3. Review the encryption status. | Encryption is confirmed for data in transit and at rest. | Pass |
| UC-017 | TC-017 | Audit Log Management | Verify that the app maintains comprehensive logs of user actions and cloud interactions for auditing purposes. | User is logged in and cloud services are configured. | 1. Perform some actions in the app (e.g., login, profile update). 2. Check the audit logs. | Comprehensive audit logs are generated and stored. | Pass |
| UC-018 | TC-018 | Cloud Resource Monitoring | Test that the app continuously monitors cloud services for security risks, misconfigurations, and potential threats. | Cloud services are registered. | 1. Log in to the app. 2. Navigate to the cloud resource monitoring section. 3. Review the monitoring data. | Continuous monitoring results in identified risks and misconfigurations. | Pass |
| UC-019 | TC-019 | Security Dashboard | Test that the user dashboard provides an overview of the cloud environment's security status, incidents, and recommended actions for improvement. | User is logged in and has registered cloud services. | 1. Navigate to the security dashboard. 2. Review security status, incidents, and recommended actions. | A summary of security status, incidents, and recommendations is displayed. | Pass |

**Smoke test for Integration test phase:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Use Case ID** | **Test Case ID** | **Test Case Title** | **Test Case Description** | **Prerequisites for Test Case Execution** | **Execution Steps** | **Expected Results** | **Result** |
| UC-001 | ST-001 | User Registration Integration | Verify that a new user can register by providing necessary information. | User is on the registration page, and the system is running. | 1. Open the registration page. 2. Enter valid user details (name, email, password). 3. Click on "Register" button. 4. Wait for confirmation. | User is successfully registered, and a confirmation message appears. The system navigates to the login page. | Pass |
| UC-002 | ST-002 | User Login Integration | Confirm that a registered user can log in using their credentials. | User has an active account and is on the login page. | 1. Open the login page. 2. Enter valid credentials (username, password). 3. Click on "Login". 4. Wait for dashboard to load. | User is successfully logged in, and the dashboard is displayed. | Pass |
| UC-003 | ST-003 | Password Reset Integration | Verify that a user can reset their password via an email link when they forget their password. | User has registered an account and has access to the email. | 1. Go to the login page. 2. Click on the "Forgot Password" link. 3. Enter registered email address. 4. Check email inbox for reset link. 5. Click on the link and reset password. | User successfully resets the password, and a confirmation message is shown. The user is redirected to the login page. | Pass |
| UC-004 | ST-004 | User Logout Integration | Evaluate that a logged-in user can log out of the application to end their session. | User is logged in to the application. | 1. Click on the "Logout" button. 2. Wait for the session to end. | User is successfully logged out and redirected to the login page. The session is terminated. | Pass |
| UC-005 | ST-005 | Multi-Factor Authentication (MFA) Setup Integration | Verify that a user can enable MFA to add an additional layer of security during login. | User is logged in and has access to the profile page. | 1. Go to profile settings. 2. Select "Enable MFA" option. 3. Follow instructions to set up MFA (e.g., using a mobile app). 4. Complete the MFA setup. | MFA is successfully enabled, and the user is prompted to use MFA on the next login. | Pass |
| UC-006 | ST-006 | Profile Update Integration | Test that users can update their personal details, including email, password, or contact information. | User is logged in and on the profile page. | 1. Go to profile settings. 2. Modify personal details (email, contact info, or password). 3. Click "Save" to update changes. 4. Verify changes. | User’s details are updated successfully, and a confirmation message is shown. | Pass |
| UC-007 | ST-007 | Cloud Service Registration Integration | Verify that users can register their cloud services for monitoring and security evaluation. | User is logged in and on the cloud services page. | 1. Go to the cloud services registration page. 2. Enter valid service details (e.g., cloud provider, access keys). 3. Click "Register Cloud Service". 4. Wait for confirmation. | Cloud service is registered successfully, and a confirmation message appears. | Pass |
| UC-008 | ST-008 | Security Configuration Check Integration | Confirm that the app can check the security configurations of the user’s cloud services for vulnerabilities and misconfigurations. | User has registered a cloud service and is on the security settings page. | 1. Open the security configurations page. 2. Select the registered cloud service. 3. Click "Check Security Configurations". 4. Wait for the scan to complete. | The system displays a summary of vulnerabilities and misconfigurations in the cloud service’s security settings. | Pass |
| UC-009 | ST-009 | Compliance Check Integration | Verify that the app can assess the cloud service's compliance with regulatory standards such as GDPR and HIPAA. | User has registered a cloud service and is on the compliance check page. | 1. Open the compliance check page. 2. Select the registered cloud service. 3. Click "Check Compliance". 4. Wait for the results. | Compliance status is displayed, showing whether the cloud service complies with regulatory standards like GDPR, HIPAA, etc. | Pass |
| UC-010 | ST-010 | Vulnerability Scanning Integration | Verify that the app can scan the cloud environment for known vulnerabilities, outdated software, and misconfigurations. | User has registered a cloud service and is on the vulnerability scan page. | 1. Go to the vulnerability scan page. 2. Select the cloud service to scan. 3. Click "Start Scan". 4. Wait for the scanning process. | The system scans the environment and displays a report with vulnerabilities, outdated software, and misconfigurations. | Pass |
| UC-011 | ST-011 | Real-time Security Alerts Integration | Test that the app provides real-time alerts when potential security threats or breaches are detected. | User has a registered cloud service with active security monitoring. | 1. Simulate a security event (e.g., unauthorized access). 2. Monitor the app for alerts. | A real-time alert is triggered and displayed in the security alerts section of the dashboard. | Pass |
| UC-012 | ST-012 | SIEM Integration | Verify that the app integrates with a SIEM platform to collect and analyze security logs from cloud environments. | SIEM platform is configured and integrated with the app. | 1. Trigger a security event in the cloud service. 2. Verify that the SIEM platform receives the event logs. 3. Check for analysis in the SIEM platform. | The SIEM platform receives the event logs and analyzes them for security threats. | Pass |
| UC-013 | ST-013 | Access Control Check Integration | Test that the app can check the cloud service’s access control policies to ensure proper configuration. | User has registered a cloud service and access control policies are configured. | 1. Go to the access control settings page. 2. Select the cloud service to review. 3. Click "Check Access Control". 4. Review the result. | Access control policies are displayed with any potential misconfigurations or issues highlighted. | Pass |
| UC-014 | ST-014 | Incident Response Integration | Confirm that the app provides guidance for users on how to respond to security incidents such as unauthorized access. | User is on the incident response page and has a registered cloud service. | 1. Simulate a security incident (e.g., unauthorized access). 2. Check for response guidance. | The system displays clear instructions and steps for mitigating the security incident. | Pass |
| UC-015 | ST-015 | Security Report Generation Integration | Verify that users can generate detailed security reports, including vulnerabilities, risks, and compliance status of their cloud environments. | User has registered cloud services and access to report generation features. | 1. Go to the report generation page. 2. Select cloud services to include in the report. 3. Click "Generate Report". 4. Wait for the report to be displayed. | A detailed security report is generated, showing vulnerabilities, risks, and compliance status for the selected cloud services. | Pass |
| UC-016 | ST-016 | Data Encryption Check Integration | Test that the app can check if encryption standards are applied to data in transit and at rest. | User has a registered cloud service and data encryption settings available. | 1. Go to the encryption settings page. 2. Select the registered cloud service. 3. Click "Check Encryption". 4. Wait for the result. | The system confirms if data encryption is properly applied in transit and at rest for the selected cloud service. | Pass |
| UC-017 | ST-017 | Audit Log Management Integration | Verify that the app maintains comprehensive logs of user actions and cloud interactions for auditing purposes. | User has logged in and accessed cloud services. | 1. Go to the audit logs page. 2. Review user actions and cloud interactions. 3. Verify the presence of logged activities. | The system displays logs of user actions and cloud interactions, with complete and accurate timestamps. | Pass |
| UC-018 | ST-018 | Cloud Resource Monitoring Integration | Test that the app continuously monitors cloud services for security risks, misconfigurations, and potential threats. | User has active monitoring set up for cloud services. | 1. Go to the monitoring page. 2. Verify active monitoring for cloud services. 3. Check for any security risks or threats displayed. | The system continuously monitors cloud services and alerts if any security risks, misconfigurations, or threats are detected. | Pass |
| UC-019 | ST-019 | Security Dashboard Integration | Test that the user dashboard provides an overview of the cloud environment's security status, incidents, and recommended actions for improvement. | User is logged in and on the security dashboard page. | 1. Open the security dashboard page. 2. Review security status, incidents, and recommended actions. | The security dashboard displays an overview of the cloud environment’s security status, incidents, and recommended actions. | Pass |

**Regression tests for Integration testing phase:**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test and Use Case ID** | **Test Description** | **Test Steps** | **Expected Results** | **Actual Results** | **Status** | **Severity** | **Priority** | **Assigned to** | **Comments/Notes** | **Remediation Steps** | **Validation Steps** |
| TC-001  UC-001 | User Registration | 1. Open the registration page. 2. Enter valid user details (name, email, password). 3. Click on "Register". 4. Wait for confirmation. | User is successfully registered, and a confirmation message appears. The system navigates to the login page. | The registration was successful, and the system redirected to the login page as expected. | Pass | Major | High | Tester | Ensure smooth registration without errors. | Re-test after fixing any issues found. | Verify registration success after issue resolution. |
| TC-002  UC-002 | User Login | 1. Open the login page. 2. Enter valid credentials (username, password). 3. Click on "Login". 4. Wait for dashboard to load. | User is successfully logged in, and the dashboard is displayed. | Login failed due to incorrect authentication logic after recent updates. | Fail | Major | High | Developer | Authentication logic needs to be fixed. | Fix the authentication flow. | Validate that the login process works post-fix. |
| TC-003  UC-003 | Password Reset | 1. Go to the login page. 2. Click on the "Forgot Password" link. 3. Enter registered email address. 4. Check email inbox for reset link. 5. Click on the link and reset password. | User successfully resets the password, and a confirmation message is shown. The user is redirected to the login page. | Password reset worked successfully and the confirmation was shown. | Pass | Critical | High | Tester | Ensure the password reset process is tested after changes. | Re-test after updates are made. | Confirm successful reset after issue resolution. |
| TC-004  UC-004 | User Logout | 1. Click on the "Logout" button. 2. Wait for the session to end. | User is successfully logged out and redirected to the login page. The session is terminated. | Logout failed after recent session handling updates. | Fail | Major | Medium | Developer | Investigate issues with session termination. | Re-test once session management is fixed. | Ensure user is logged out and session is terminated. |
| TC-005  UC-005 | MFA Setup | 1. Go to profile settings. 2. Select "Enable MFA" option. 3. Follow instructions to set up MFA (e.g., using a mobile app). 4. Complete MFA setup. | MFA is successfully enabled, and the user is prompted to use MFA on the next login. | MFA was successfully enabled, and the user was prompted during login. | Pass | Critical | High | Tester | Verify if MFA setup still works after new feature integration. | Re-test after fixing any issues related to MFA. | Validate MFA setup by testing its prompt during login. |
| TC-006  UC-006 | Profile Update | 1. Go to profile settings. 2. Modify personal details (email, contact info, or password). 3. Click "Save" to update changes. 4. Verify changes. | User’s details are updated successfully, and a confirmation message is shown. | Profile update worked and confirmation appeared. | Pass | Major | Medium | Developer | Confirm that profile update works as intended. | Re-test after issues with save functionality are fixed. | Confirm that details are correctly updated. |
| TC-007  UC-007 | Cloud Service Registration | 1. Go to the cloud services registration page. 2. Enter valid service details (e.g., cloud provider, access keys). 3. Click "Register Cloud Service". 4. Wait for confirmation. | Cloud service is registered successfully, and a confirmation message appears. | Cloud service registration worked as expected. | Pass | Major | Medium | Tester | Ensure smooth registration without errors. | Re-test after fixing any errors found. | Verify service is properly registered. |
| TC-008  UC-008 | Security Configuration Check t | 1. Open the security configurations page. 2. Select the registered cloud service. 3. Click "Check Security Configurations". 4. Wait for the scan to complete. | The system displays a summary of vulnerabilities and misconfigurations in the cloud service’s security settings. | Security configuration scan completed successfully and results were displayed correctly. | Pass | Major | Medium | Tester | Verify that scanning works correctly after updates. | Re-test after any scanning logic issues are fixed. | Confirm that vulnerabilities and misconfigurations are properly displayed. |
| TC-009  UC-009 | Compliance Check | 1. Open the compliance check page. 2. Select the registered cloud service. 3. Click "Check Compliance". 4. Wait for the results. | Compliance status is displayed, showing whether the cloud service complies with regulatory standards like GDPR, HIPAA, etc. | Compliance check passed and displayed regulatory compliance status. | Pass | Critical | High | Developer | Ensure the compliance check feature works after changes. | Re-test after fixing any issues related to regulatory standards. | Confirm that compliance results are shown correctly. |
| TC-010  UC-010 | Vulnerability Scanning | 1. Go to the vulnerability scan page. 2. Select the cloud service to scan. 3. Click "Start Scan". 4. Wait for the scanning process. | The system scans the environment and displays a report with vulnerabilities, outdated software, and misconfigurations. | The scan completed, and the vulnerability report was displayed correctly. | Pass | Critical | High | Tester | Test the scan after bug fixes to ensure proper functionality. | Re-test after scanning logic is fixed. | Verify that vulnerabilities are accurately listed in the report. |
| TC-011  UC-011 | Real-time Security Alerts | 1. Simulate a security event (e.g., unauthorized access). 2. Monitor the app for alerts. | A real-time alert is triggered and displayed in the security alerts section of the dashboard. | Alert was triggered and displayed as expected. | Pass | Critical | High | Tester | Check if real-time alerting is functioning as expected. | Re-test after real-time event simulation. | Ensure alerts appear promptly during incidents. |
| TC-012  UC-012 | SIEM Integration | 1. Trigger a security event in the cloud service. 2. Verify that the SIEM platform receives the event logs. 3. Check for analysis in the SIEM platform. | The SIEM platform receives the event logs and analyzes them for security threats. | SIEM integration worked successfully, logs were transmitted, and analyzed correctly. | Pass | Major | Medium | Developer | Ensure logs are being properly transmitted. | Re-test after resolving transmission issues. | Validate logs are correctly transmitted and analyzed in the SIEM system. |
| TC-013  UC-013 | Access Control Check | 1. Go to the access control settings page. 2. Select the cloud service to review. 3. Click "Check Access Control". 4. Review the result. | Access control policies are displayed with any potential misconfigurations or issues highlighted. | Access control check completed successfully with no issues. | Pass | Major | Medium | Tester | Ensure that access control policies are verified correctly. | Re-test if any access control issues arise. | Confirm that the access control check is working. |
| TC-014  UC-014 | Incident Response | 1. Simulate a security incident (e.g., unauthorized access). 2. Check for response guidance. | The system displays clear instructions and steps for mitigating the security incident. | Incident response displayed the correct instructions, guiding the user. | Pass | Critical | High | Developer | Verify that incident response guidance works as expected. | Re-test after any issues with response display are fixed. | Confirm that incident response instructions are accurate and clear. |
| TC-015  UC-015 | Security Report Generation | 1. Go to the report generation page. 2. Select cloud services to include in the report. 3. Click "Generate Report". 4. Wait for the report to be displayed. | A detailed security report is generated, showing vulnerabilities, risks, and compliance status for the selected cloud services. | Report generation failed due to formatting issues. | Fail | Minor | Low | Developer | Fix formatting issues with the report generation. | Re-test after fixing the formatting bug. | Ensure that the report is correctly generated with accurate data. |
| TC-016  UC-016 | Data Encryption Check | 1. Go to the encryption settings page. 2. Select the registered cloud service. 3. Click "Check Encryption". 4. Wait for the result. | The system confirms if data encryption is properly applied in transit and at rest for the selected cloud service. | Encryption status was correctly displayed. | Pass | Critical | High | Tester | Verify encryption functionality after updates. | Re-test after addressing any encryption validation issues. | Confirm that encryption settings are correctly displayed. |
| TC-017  UC-017 | Audit Log Management | 1. Go to the audit logs page. 2. Review user actions and cloud interactions. 3. Verify the presence of logged activities. | The system displays logs of user actions and cloud interactions, with complete and accurate timestamps. | Logs were correctly displayed, showing user actions and cloud interactions. | Pass | Minor | Low | Developer | Ensure that logs are correctly displayed and accessible. | Re-test after addressing any log retention issues. | Confirm that user actions are correctly logged. |
| TC-018  UC-018 | Cloud Resource Monitoring | 1. Open the resource monitoring page. 2. Select the cloud service. 3. Click "Monitor". 4. Check the displayed monitoring results. | The system shows real-time monitoring data, including security risks, misconfigurations, and threats. | Monitoring results were displayed correctly, with no issues. | Pass | Major | Medium | Tester | Ensure resource monitoring continues functioning post-integration. | Re-test after any updates to monitoring logic. | Confirm real-time monitoring data is accurate. |
| TC-019  UC-019 | Security Dashboard | 1. Open the dashboard page. 2. Review the security status overview. 3. Verify incidents and recommended actions. | The dashboard displays the security status, incidents, and recommended improvements. | The dashboard displayed correct data, including incidents and recommendations. | Pass | Major | Medium | Developer | Ensure the dashboard displays all relevant security information. | Re-test after updating dashboard display logic. | Confirm that the dashboard shows correct and current data. |

## 7.3. Penetration Test

**Use Cases**

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **Title** | **Description** |
| **PT-001** | SQL Injection | Test if the application is vulnerable to SQL injection by inserting malicious SQL queries into input fields. |
| **PT-002** | Cross-Site Scripting (XSS) | Verify if the application is susceptible to XSS attacks by injecting malicious scripts into user input fields. |
| **PT-003** | Broken Authentication | Assess the authentication mechanisms to ensure they prevent unauthorized access and brute force attacks. |
| **PT-004** | Sensitive Data Exposure | Verify if sensitive data like passwords and session tokens are transmitted securely using encryption. |
| **PT-005** | Cross-Site Request Forgery (CSRF) | Test if the application is protected against unauthorized actions initiated from third-party websites. |
| **PT-006** | Insecure Direct Object References (IDOR) | Check if unauthorized access to resources can be achieved by manipulating input values. |
| **PT-007** | Security Misconfigurations | Identify if any default configurations, unnecessary services, or insecure permissions exist in the application or server setup. |
| **PT-008** | Denial of Service (DoS) | Simulate excessive traffic or resource-intensive operations to evaluate if the system is resistant to DoS attacks. |
| **PT-009** | Privilege Escalation | Assess whether users can elevate their privileges and gain access to restricted areas of the application. |
| **PT-010** | Weak Session Management | Verify if session tokens are unique, secure, and properly invalidated upon logout or timeout. |
| **PT-011** | API Security | Test APIs for vulnerabilities such as unauthorized access, improper data validation, and insecure communication. |
| **PT-012** | Input Validation | Ensure the application validates and sanitizes user inputs to prevent injection or buffer overflow attacks. |
| **PT-013** | File Upload Vulnerabilities | Test if the application prevents malicious file uploads that could compromise the server. |
| **PT-014** | Open Redirects | Check if the application is susceptible to redirecting users to malicious websites. |
| **PT-015** | Directory Traversal | Test if attackers can access restricted files or directories by manipulating file paths. |

**Test Cases:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Case Title** | **Test Case Description** | **Prerequisites for Test Case Execution** | **Execution Steps** | **Expected Results** | **Result** |
| TC- 001 | SQL Injection Vulnerability Test during User Registration | Verify that the application prevents SQL injection in input fields. | Access to input fields and database monitoring tools. | 1. Enter SQL payloads such as ' OR '1'='1 into registration fields. 2. Monitor database logs and application behavior. 3. Verify if any unauthorized queries are executed. | The application should sanitize inputs and prevent SQL injection attacks. No unauthorized queries should be executed. |  |
| TC- 002 | XSS Vulnerability Test in Comments Section | Verify that the application prevents malicious script injection in input fields. | Access to a user account with permission to comment. | 1. Submit scripts like <script>alert('XSS')</script> in the comments field. 2. Refresh the page and check if the script executes. 3. Inspect backend logs for script execution traces. | The application should sanitize inputs to prevent script execution. The submitted script should be neutralized. |  |
| TC- 003 | Authentication Vulnerability Test in Login | Verify authentication mechanisms against brute force and token theft attacks. | Access to a test account and login page. | 1. Perform brute force attempts with weak credentials. 2. Test token validity after logout or expiration. 3. Simulate login bypass using stolen session tokens. | The application should enforce account lockouts, invalidate old tokens, and block unauthorized access attempts. |  |
| TC- 004 | Sensitive Data Transmission Test during Login | Verify that sensitive data is encrypted during transmission. | Tools to intercept network traffic (e.g., Wireshark). | 1. Log in and capture network traffic. 2. Check for plaintext transmission of credentials or session tokens. 3. Verify the use of encryption protocols such as TLS. | Sensitive data should be encrypted and transmitted securely. No plaintext credentials should be visible in traffic. |  |
| TC- 005 | CSRF Protection Test in Password Change Functionality | Verify that the application is protected against CSRF attacks. | Access to a user account and CSRF testing tools. | 1. Generate a malicious CSRF form targeting password change functionality. 2. Host the form on a third-party site. 3. Simulate a victim clicking the malicious link. | The application should require CSRF tokens for sensitive actions. Unauthorized actions should be rejected. |  |
| TC- 006 | Insecure Direct Object Reference Test in Resource Access | Verify that unauthorized access to resources is not possible. | Access to valid resource identifiers. | 1. Access a resource using a valid identifier. 2. Modify the identifier to access unauthorized resources. 3. Observe if access is granted. | The application should validate user permissions and deny access to unauthorized resources. |  |
| TC- 007 | Security Misconfigurations Test in Server Setup | Identify insecure configurations, unnecessary services, or open ports. | Tools like Nmap or Nessus for configuration analysis. | 1. Scan the server for open ports and unused services. 2. Inspect server configuration files for insecure settings. 3. Check for unnecessary features like directory indexing. | The server should only have essential services running. All insecure configurations should be eliminated. |  |
| TC- 008 | DoS Resilience Test using High Traffic Simulation | Evaluate if the system withstands high traffic without crashing. | Tools to generate high traffic, such as LOIC. | 1. Simulate high traffic using stress testing tools. 2. Monitor application response times and server stability. 3. Inspect logs for dropped requests or performance issues. | The application should handle high traffic gracefully without crashing or slowing down excessively. |  |
| TC- 009 | Privilege Escalation Vulnerability Test in User Roles | Verify that unauthorized privilege escalation is not possible. | Access to accounts with different privilege levels. | 1. Attempt to access admin functionalities from a regular user account. 2. Modify API requests to elevate privileges. 3. Observe the system's behavior. | Unauthorized privilege escalation attempts should be denied, and proper logging should occur. |  |
| TC- 010 | Session Management Test during User Logout | Verify session tokens are invalidated after logout. | Access to a test account and browser tools. | 1. Log in and capture session token. 2. Logout and attempt to reuse the session token. 3. Observe if access is granted using the old token. | The application should invalidate session tokens immediately after logout. |  |
| TC- 011 | API Security Test for Authorization Vulnerabilities | Verify APIs enforce authorization rules for accessing data. | Access to API endpoints and testing tools like Postman. | 1. Call an API endpoint with a valid token. 2. Attempt the same call with an invalid or expired token. 3. Observe responses for unauthorized access attempts. | The API should block unauthorized requests and provide meaningful error messages. |  |
| TC- 012 | Input Validation Test during Data Submission | Verify the application sanitizes and validates user inputs. | Access to input forms and backend logs. | 1. Submit invalid and malicious inputs in forms. 2. Check for errors or buffer overflows in the system. 3. Inspect logs for unhandled exceptions. | The application should sanitize all inputs and reject malicious or invalid data. |  |
| TC- 013 | File Upload Test for Malicious Content | Verify the application blocks malicious file uploads. | Access to file upload functionality and server logs. | 1. Attempt to upload files with malicious extensions or content (e.g., .exe, .php). 2. Observe if the upload is rejected or sanitized. 3. Check server directories for uploaded files. | The application should reject malicious file uploads and log attempts. No harmful files should be executed on the server. |  |
| TC- 014 | Open Redirect Test during URL Handling | Verify the application prevents open redirection attacks. | Access to redirect functionalities. | 1. Craft a URL redirecting to a malicious site. 2. Submit the URL to the application. 3. Observe if redirection to untrusted sites occurs. | The application should validate redirection URLs and reject untrusted destinations. |  |
| TC- 015 | Directory Traversal Test in File Paths | Verify attackers cannot access restricted files via directory traversal. | Access to file path-related functionalities. | 1. Submit file paths like ../../etc/passwd in input fields. 2. Monitor responses for file disclosures. 3. Check logs for unauthorized access attempts. | The application should block directory traversal attempts and restrict access to sensitive files. |  |

# Defect Management

## Defect Management Life Cycle

A diagram of a flowchart

Description automatically generated

In this section, a detailed explanation of the Defect Management Lifecycle is provided, maintaining the flow logic in order:

**1. Defect Found**

* **Status: New**  
  A defect is identified and logged into the system, with details such as steps to reproduce, severity, and environment.

**2. Defect Assignment**

* **Status: Assigned**  
  The defect is assigned to the appropriate developer or team for resolution.

**3. Fix Implementation**

* **Status: Open**  
  The developer works on fixing the defect. The defect remains "Open" until it is resolved.

**4. Defect Fixed**

* **Status: Fixed**  
  Once resolved, the developer marks the defect as "Fixed" and prepares it for retesting.

**5. Retesting**

* **Status: Retest**  
  The testing team verifies that the defect has been fixed and that the issue no longer exists.

**6. Defect Verification**

* **Status: Verified**  
  After successful retesting, the defect is confirmed as fixed by the testing team.

**7. Defect Closure**

* **Status: Closed**  
  The defect is closed, indicating the issue is fully resolved.

**8. Reopen (if required)**

* **Status: Reopen**  
  If the issue reappears, the defect is reopened for further investigation and fixes.

**Other Possible Statuses:**

* **Duplicate**: The defect is a duplicate of an existing issue and is closed.
* **Deferred**: The defect is not urgent and is deferred to a later release.
* **Not A Bug**: The issue is determined not to be a bug and is closed.
* **Rejected**: The defect is invalid or not reproducible and is rejected.

This lifecycle ensures that defects are tracked, resolved, and closed in a structured manner, ensuring software quality.

# Conclusion

In conclusion, the security evaluation app for cloud services serves as a comprehensive tool for assessing the security and compliance of cloud environments. The application is intended to test a variety of essential factors, including vulnerability scans, security configuration checks, compliance assessments, and data encryption requirements. It also sends real-time security alerts and interfaces with other security platforms such as SIEM to ensure ongoing monitoring.

By allowing users to register cloud services, modify security settings, and create thorough reports, the software guarantees that cloud environments are safe from potential attacks and meet relevant legal requirements. The app's functionalities are designed to be user-friendly and scalable, assuring its effectiveness across several cloud service setups.

Future enhancements could concentrate on improving automation capabilities, integrating AI models, and broadening integration options with other security platforms.

# Remediation Steps

|  |  |
| --- | --- |
| **№** | **Remediations** |
| **1** | Ensure that all user passwords meet complexity requirements, including minimum length and a combination of uppercase, lowercase, numbers, and special characters. |
| **2** | Only allow specific file types and scan uploaded files for malware to prevent security risks related to file uploads. |
| **3** | Perform regular penetration tests to identify and resolve potential vulnerabilities before attackers can exploit them. |
| **4** | Limit access to sensitive data by defining precise user roles and permissions, ensuring that only authorized personnel can access or modify critical information. |
| **5** | Use strong encryption protocols such as AES and TLS to protect sensitive data both while stored and during transmission, ensuring that unauthorized parties cannot access or tamper with the data. |

# References

<https://peakin.in/defect-life-cycle-in-software-testing/>

[https://www.crowdstrike.com/en-us/cybersecurity-101/cloud-security/cloud-security-assessment/#:~:text=By%20reviewing%20the%20organization's%20network,improve%20capabilities%20in%20the%20future.](https://www.crowdstrike.com/en-us/cybersecurity-101/cloud-security/cloud-security-assessment/%23:~:text=By%20reviewing%20the%20organization's%20network,improve%20capabilities%20in%20the%20future.)

<https://aimconsulting.com/insights/evaluate-cloud-service-provider-security-benefits-criteria/>

<https://www.techtarget.com/searchsecurity/tip/How-to-conduct-a-cloud-security-assessment>

<https://www.mimecast.com/blog/what-is-cloud-siem/>