KING SAUD UNIVERSITY
COLLEGE OF COMPUTER AND INFORMATION SCIENCES
DEPARTMENT OF SOFTWARE ENGINEERIN

SWE314: Software Security Engineering

Project phase 2

Course Code / Title: SWE 314

Assessment: Project
Deliverable Semester / Year: Spring 2023

Submission Date: 5/6/23

Prepared by

	Name	ID
1	Sultan Al-enzi	442106994
2	Khalid Al-harbi	442103477
3	Abdulmajeed Al-romaih	442101425

Supervision

by

Dr. Noureddine Abbadeni

Contents

1. Summary

The report focuses on enhancing security in a customer banking system by establishing security requirements, analyzing risks and threats, and presenting effective solutions. It will cover a use case diagram that will be transformed into a misuse case diagram, highlighting potential misuse scenarios and corresponding mitigation strategies. Additionally, the report involves exploring software architecture and design considerations to develop a secure software product.

2. Introduction

The objective of this report is to conduct a comprehensive analysis and evaluation of the security requirements for a banking system. This banking system enables customers to engage in a range of online banking activities, including login procedures, account information access, transactions, messaging, and statement printing. Due to the sensitive nature of the information involved, such as passwords, account details, and transaction data. In this report, each requirement of the banking system is meticulously examined, along with its associated risks and threats. The aim is to identify potential vulnerabilities and potential attack vectors that may compromise the system's security.

To provide a visual representation of the system's functionality and potential security vulnerabilities, a use case diagram is created as part of this report. It is further refined and transformed into a misuse case diagram, showcasing various misuse scenarios that malicious actors could exploit. Moreover, the report also emphasizes the development of corresponding mitigation methods to address and minimize the impact of these misuse scenarios.

In addition to addressing security requirements and vulnerabilities, the report includes an overview of the software architecture and design choices.

3. List of functional requirements

The following section outlines the functional requirements for the 'User' component of the system, detailing the specific features and capabilities that users should have access to.

3.1 The user shall be able to log in to the system.

3.1.1 input: username and password.

3.1.2 output: none

3.2 The user shall be able to view transaction information.

3.2.1 input: none.

3.2.2 output: transaction information.

3.3 The user shall be able to view profile information.

3.3.1 input: none.

3.3.2 output: profile info.

3.4 The user shall be able to view the remaining amount of their money.

3.4.1 input: none.

3.4.2 output: none.

3.5 The user shall be able to perform payment.

3.5.1 input: amount.

3.5.2 output: remaining amount .

3.6 The user shall be able to perform local transactions.

3.6.1 input: amount, beneficiary account

3.6.2 output: none.

3.7 The user shall be able to perform national transaction.

3.7.1 input: amount, beneficiary account.

3.7.2 output: none.

3.8 The user shall be able to perform international transaction.

3.8.1 input: amount, beneficiary account.

3.8.2 output: none.

3.9 The user shall be able to send message to the bank.

3.9.1 input: message.

3.9.2 output: none.

3.10 The user shall be able to send inquiry to the bank.

3.10.1 input: inquiry.

3.11.2 output: none.

3.11 The user shall be able to print Account statement for a given period of time.

3.11.1 input: period time.

3.11.2 output: account statement.

4. Analysis of software security requirements

This step will be done using:

1. Requirement.
2. Fail case analysis.
3. Consequence of failure.
4. Associated risks/threats.

4.1: The user shall be able to login to the system.

4.1.1 Fail case:

1- user can’t login error in username/password.

2- Username and password used by different users.

4.1.2 Consequences:

1- User will not be happy with the banking system

2- non-authorized user can access the system.

4.1.3 Associated risks/threats:

1- DDOS attack.

2- a malicious user can access sensitive data like the amount of money.

3- SQL injection.

4.2: The user shall be able to view transaction information.

4.2.1 Fail case:

1-User can’t view transactions

2-transcation viewed by another user

4.2.2 Consequences:

1-User can’t identify the amount he sent, and beneficiary

2-non-authorized user can view transaction information like bank account’s number, and account’s name

4.2.3 Associated risks/threats:

1- DDOS attack network data

2- packet sniffer

4.3: The user shall be able to view profile information.

4.3.1 Fail case:

1- User can’t view profile information.

2-non-authorized user can view profile information.

4.3.2 Consequences:

1- User won’t be happy to see certain information in certain time

2- non-authorized user can view name, and account number.

4.3.3 Associated risks/threats:

1- Sensitive date of profile information can be viewed which harm confidentiality which leads to a lawsuit against the bank

2- Spyware attack to steal sensitive information

- anonymity attack

4.4 **The user shall be able to view the remaining amount of their money.**

4.4.1 Fail case:

1- User can’t see the amount of money

2- Remaining amount of money is wrong

4.4.2 Consequences:

1- User can’t make any process because of not knowing the amount of money.

2- User can’t see the expected remaining amount of money.

4.4.3 associated risks/threats:

1- MITM Attack

2-Date integrity of the amount which is not true

4.5 The user shall be able to perform payment.

4.5.1 Fail case:

1- payment is not accepted

2- Payment is sent to the wrong beneficiary

3- a malicious website receives payment

4.5.2 Consequences:

1- User is unhappy with the payment process

2- The user made payment to the wrong person

3-The payment is not sent to the right person

4.5.3 Associated risks/threats:

1- Phishing attack

2- man in the middle attack alteration of payment

3- Data integrity

4.6 The user shall be able to perform local transaction.

4.6.1 Fail case:

1- Transaction processing error.
2- Transaction by lockout account.

4.6.2 Consequences:

1- Wrong transaction to wrong destination.
2- Non-authorized transaction.

4.6.3 Associated risks/threats:

1- Payment fraud: Users may be victims of payment fraud, where attackers exploit vulnerabilities in the payment infrastructure or payment processing systems.
2- Transaction errors.
3- Man-in-the-middle attacks.
4- Phishing attacks.

4.7 The user shall be able to perform national transaction.

4.7.1 fail case:

1- Transaction processing error.

2- Transaction by lockout account.

4.7.2 Consequences:

1- Wrong transaction to wrong destination.

2- non-authorized transaction.

4.7.3 Associated risks/threats:

1- Payment fraud: Users may be victims of payment fraud, where attackers exploit vulnerabilities in the payment infrastructure or payment processing systems.

2- Transaction errors.

3- Man-in-the-middle attacks.

4- phishing attacks.

4.8 The user shall be able to perform international transaction.

This functional requirement necessitates huge security constraints. This is due to the complex nature of tracking the money flow, particularly when dealing with substantial amounts reaching into the millions.

4.8.1 Fail case:

1- Invalid or unsupported destination country.

2- Currency conversion.

4.8.2 Consequences:

1- Restricted international transactions.

2- Lack of transaction traceability.

4.8.2 Associated risks/threats:

1- Payment fraud: Users may be victims of payment fraud, where attackers exploit vulnerabilities in the payment infrastructure or payment processing systems.

2- Transaction errors.

3- Man-in-the-middle attacks.

4.9 The user shall be able to send message to the bank.

4.9.1 Fail case:

1- Message sent by the user didn't receive to the back.

2- Message sent by the bank didn't receive to the user.

3- Send the message by another user who is not supposed to send the message.

4.9.2 Consequences:

1- non-authorized User send messages

2- User will not be able to receive message

3- The message sent by the user may be intercepted and stolen before it is received by the recipient.

4.9.3 Associated risks/threats:

1- Man-in-the-middle attacks

2- Phishing attacks

3- Spoofing attacks

4- Social engineering attacks

4.10 The user shall be able to send an inquiry to the bank.

4.10.1 Fail case:

1- enquiries sent with wrong information.

4.10.2 Consequences:

1- bank response with wrong action

4.10.3: Associated risks/threats:

No risks or threats to be mentioned in This F.R

4.11 The user shall be able to print an Account statement for a given period of time.

4.11.1 Fail case:

1- user is not able to print the account statement.

4.11.2 Consequences:

1- User dissatisfaction

4.11.3 associated risks/threats:

No risks or threats to be mentioned in This F.R

5. Proposed solutions to the identified threats/risks

To address the identified threats/risks, here are some proposed solutions:

	Functional Requirement Number	Security threat	Mitigation
1	4.1	1- a malicious user can access sensitive data like the amount of money 2- DDOS attack 3- SQL injection	- 2-step authentication - Server Farm - Configure Firewall
2	4.2	1- DDOS attack on network data 2- packet sniffer	- Server Farm - Configure Firewall - Encryption
3	4.3	1- sensitive date of profile information can be viewed which harm confidentiality which leads to a lawsuit against the bank 2-Spyware attack to steal sensitive information - anonymity attack	- Encryption of sensitive data - Intrusion detection - Mixing Technique
4	4.4	1- Man-in-the-middle Attack 2-Date integrity of the amount which is not true	- Monitor network. - Access control - Regular auditing
5	4.6, 4.7 and 4.8	1- Payment fraud 2- Man-in-the-middle attacks. 3- Phishing attacks.	- Configure Firewall - Two factors Authentication - Certificate Validation - Regular Security Audits
6	4.9	1- Spoofing 2- Man-in-the-middle attacks. 3- Phishing attacks. 4- Social engineering attacks	- Configure Firewall. - Awareness. - Two factor authentication

6. Use Case Diagram

The use case diagram presented below outlines the different roles and responsibilities of the user interacting with the system.

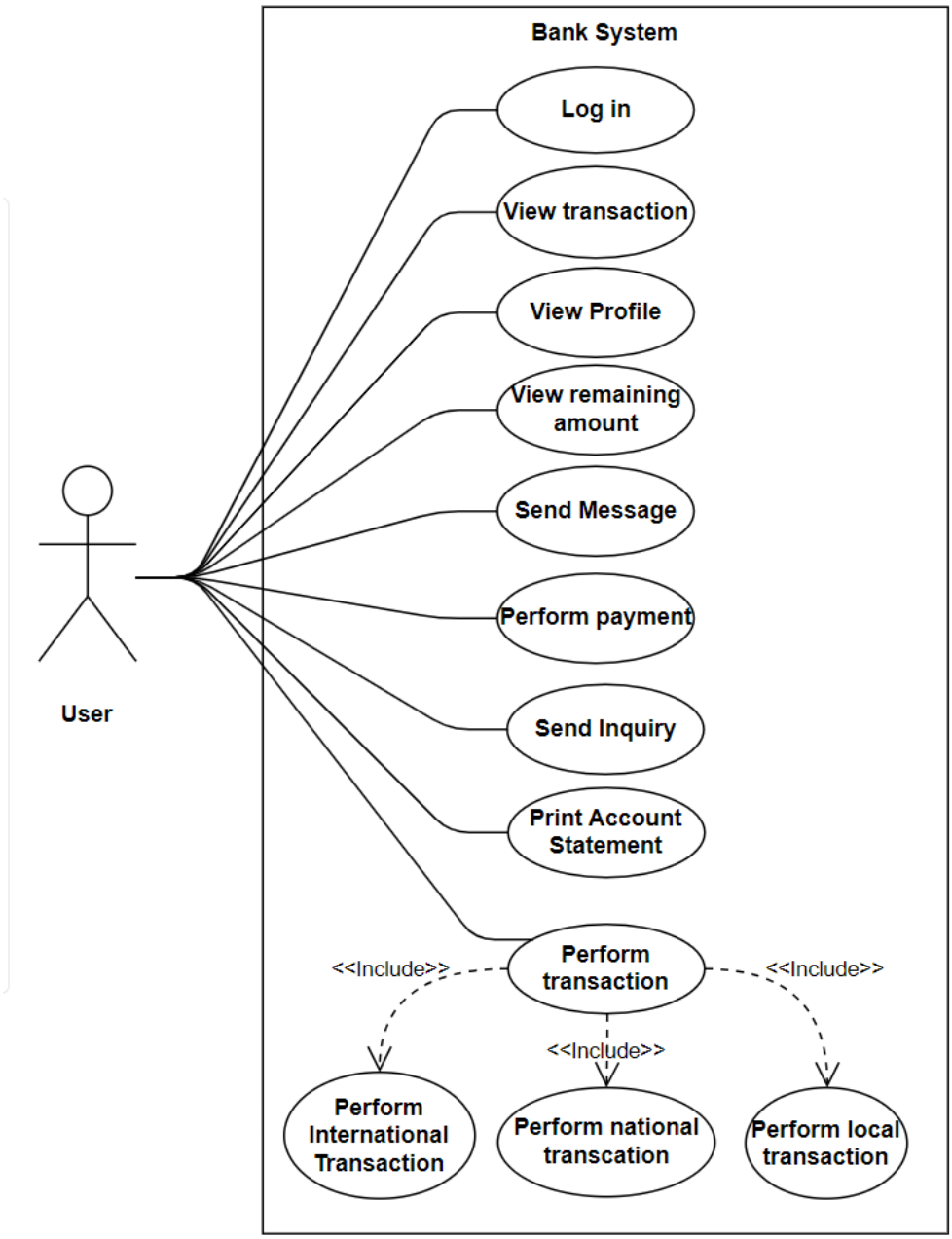

Figure 1 The shown use case model shows the functionalities performed by the user.

7. Misuse Case Diagram

Below is the Misuse Case Diagram illustrating potential misuses and malicious actions involving both users and attackers within the system.

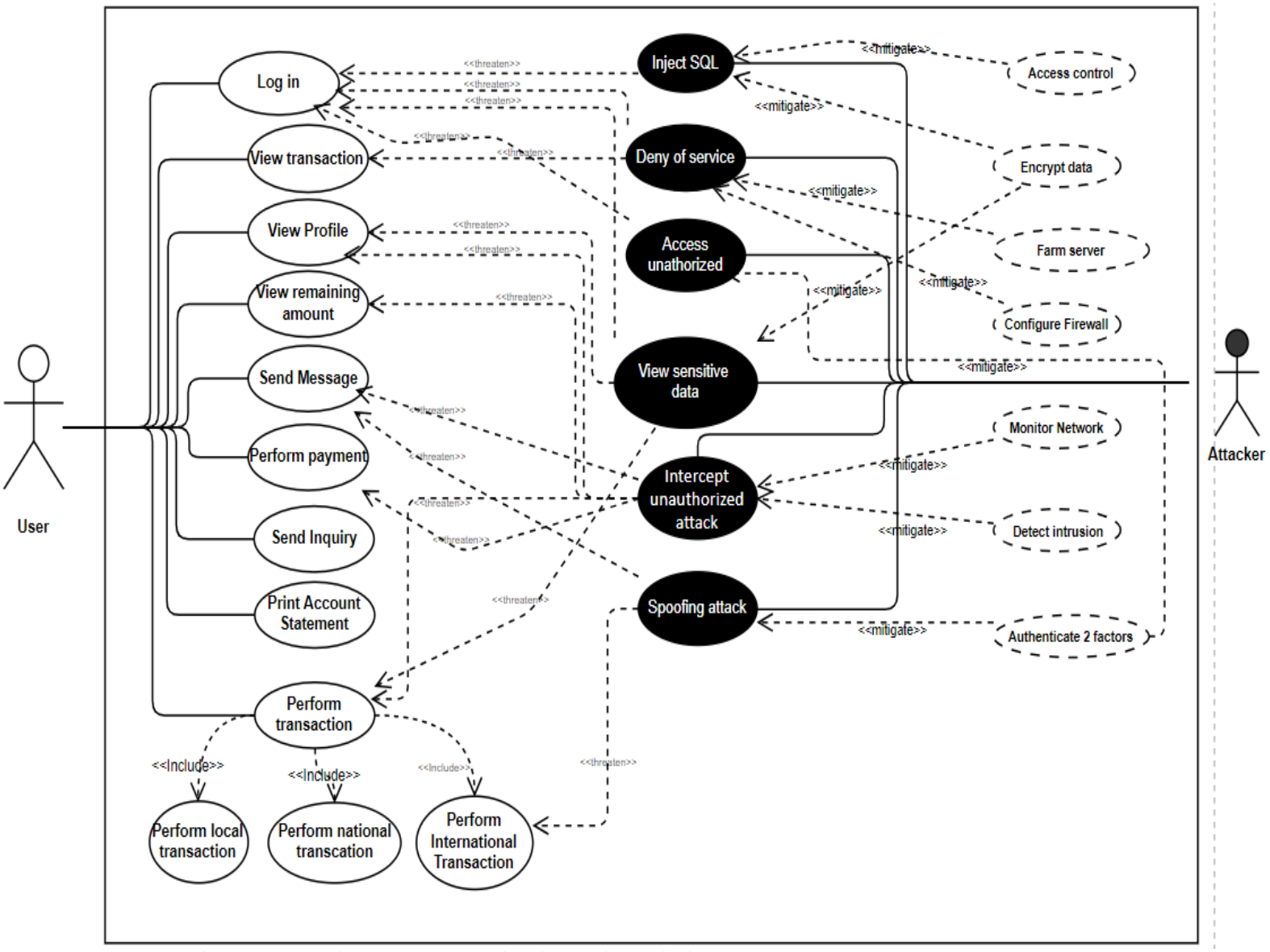

Figure 2 In this mis-use case diagram represents what the attacker might do.

8. Misuse case descriptions

The following section provides a concise overview of various misuse case descriptions related to potential security vulnerabilities. These misuse cases highlight specific threats that an application or system may face, including Denial of Service (DoS) attacks, SQL injection, viewing sensitive data, spoofing, and unauthorized access. Understanding these misuse cases is crucial for identifying and mitigating potential risks, protecting valuable data, and maintaining integrity and availability.

8.1 deny of service.

Misuse case: deny of service	**Use case:** Log in, view transaction	**Mitigated use case:** farm server, configure firewall	
Misuser: Attacker			
Context: This misuse case attack floods the target with an enormous volume of data or traffic, consuming its available network resources and causing it to become unresponsive.			
Threat:	**Description:**	**Motivation:**	**Mitigation:**
lie in their ability to disrupt services, cause financial losses, damages reputations, and potentially expose vulnerabilities leading to data breaches.	deny of service is malicious attack that overwhelms a target system with a flood of traffic or requests, causing disruption and rendering it inaccessible to legitimate users.	The motivation behind deny of service attacks can vary, including seeking financial gain or revenge or simply engaging in malicious behavior for personal satisfaction.	- farm sever (external) - configure firewall (external)

8.2 Access unauthorized.

Misuse case: Access unauthorized	**Use case:** Log in	**Mitigated use case:** Two factor authentication	
Misuser: Attacker			
Context: This misuse case involves breaching its security to gain entry without proper authorization.			
Threat:	**Description:**	**Motivation:**	**Mitigation:**
The threat of unauthorized access to a bank system login involves potential risks such as unauthorized financial transactions, unauthorized access to customer accounts, theft of sensitive financial information, compromise of customer privacy, identity theft, reputational damage to the bank, regulatory non-compliance, and potential legal consequences.	Accessing a bank system without authorization refers to the unauthorized entry or login into the bank's network or applications by individuals or entities who do not have legitimate access rights, potentially leading to unauthorized financial transactions, unauthorized access to customer accounts, theft of sensitive financial information, compromise of customer privacy, reputational damage to the bank.	The motivation behind attempting to access a bank system without authorization can vary, including financial gain through fraudulent transactions, theft of funds or sensitive financial information, gathering intelligence for future attacks, sabotage, revenge, personal satisfaction, or as part of a larger criminal enterprise seeking to exploit weaknesses in the banking system for illicit purposes.	- authenticate in two-factor authentication (internal): 1-Require 2 authentication methods during login. 2- Securely store authentication data • Monitor and update security measures

8.3 View sensitive data

Misuse case: View sensitive data	**Use case:** View profile, perform transaction	**Mitigated use case:** Encryption data	
Misuser: Attacker			
Context: This misuse case any confidential, private, or personally identifiable information that, if accessed or disclosed without authorization, could potentially harm individuals.			
Threat:	**Description:**	**Motivation:**	**Mitigation:**
The threat of viewing sensitive data lies in the potential for unauthorized access, leading to privacy breaches, identity theft, financial loss, reputational damage.	Viewing sensitive data refers to accessing or examining confidential or personally identifiable information that requires appropriate authorization and safeguards to protect the privacy and security of individuals and organizations.	The motivation behind viewing sensitive data can include malicious intent to exploit or misuse the information for financial gain, personal advantage, identity theft, espionage, or to expose vulnerabilities for blackmail, sabotage, or disruption.	- Encryption data (internal) - conducting regular security audits, and employing encryption techniques to ensure that only authorized individuals with a legitimate need can access the sensitive information.

8.4 intercept authorization attack

Misuse case: intercept authorization attack	**Use case:** Send message, perform payment, perform transaction	**Mitigated use case:** Monitor network, Detect intrusion	
Misuser: Attacker			
Context: This misuse case Intercepting unauthorized attacks entails the proactive identification and prevention of malicious activities.			
Threat:	**Description:**	**Motivation:**	**Mitigation:**
The threat of intercepting unauthorized attacks in a bank system includes potential financial losses, compromise of sensitive customer information, and disruption of critical banking service.	The user data is spied on during Send message, perform payment, perform transaction.	The motive is to steal user data, tamper with it, and financial theft	-Monitor network (external) -Detect intrusion (external)

8.5 Spoofing attack

Misuse case: Spoofing attack	**Use case:** Send message, perform international transaction	**Mitigated use case:** Authenticate 2 factors	
Misuser: Attacker			
Context: This misuse case allows the misuser to change the destination IP address			
Threat:	**Description:**	**Motivation:**	**Mitigation:**
The threat of spoofing involves the malicious act of impersonating a person, device, or system to deceive and gain unauthorized access.	Spoofing is receiving data sent in the network without permission. By changing the IP address of the destination device.	The motivation behind spoofing includes attempting to gain unauthorized access to sensitive financial information potentially causing financial losses for both the bank and its customers.	- authenticate in two-factor authentication (internal): 1-Require 2 authentication methods during login. 2- Securely store authentication data • Monitor and update security measures.

8.6 Inject SQL

Misuse case: Inject SQL	**Use case:** Log in	**Mitigated use case:** Access control, Encryption data.	
Misuser: Attacker			
Context: This misuse case common web application vulnerability where malicious SQL code is inserted into an application's input fields,			
Threat:	**Description:**	**Motivation:**	**Mitigation:**
Attackers could manipulate the login process by injecting malicious SQL code, extracting sensitive financial information, tampering with transaction records.	SQL injection in a login occurs when an attacker exploits vulnerabilities in the login mechanism by injecting malicious SQL code into input fields, allowing them to manipulate database queries	The motivation behind attempting SQL injection in a is typically driven by the desire to gain unauthorized access to customer accounts, extract valuable financial information, conduct fraudulent transactions, exploit system vulnerabilities for personal gain or financial benefit.	-Encryption data (internal): Encrypt sensitive data, both in transit and at rest, using industry-standard encryption algorithms, which adds an extra layer of protection in case of a successful SQL injection attack. - Access control(internal)

9. Design Choices

In this section, we will discuss the architectural decisions made for the system and how they address the project requirements.

9.1 Design Architecture

We will be implementing a 3-tier architecture for our system, consisting of the Data Tier, Logic Tier, and Client Tier. The utilization of a 3-tier architecture is highly regarded for bank systems due to its inherent security benefits. This architecture enables the separation of the presentation, application, and database layers, allowing for optimization of each layer's specific functionality. Additionally, it facilitates easy scalability as the bank's requirements evolve, support security through the implementation of firewalls and access controls.

The adoption of a 3-tier architecture also contributes to streamlined maintenance and efficient development processes. It facilitates parallel development across different parts of the system, enhancing productivity and reducing development time. With its focus on security without compromising functionality.

9.2 Threats

While the 3-tier architecture provides significant security advantages for bank systems, it is important to address potential threats that may still exist within this system. One such threat is eavesdropping, which refers to the unauthorized interception and monitoring of sensitive data transmitted between the tiers of the architecture.

9.3 Class diagram

The class diagram depicts the organization of classes and their relationships within the system, particularly focusing on the handling of sensitive data. The sensitive data, such as usernames, passwords, dates of birth, and amounts, needs to be treated with utmost privacy and security. To ensure this, the following measures we shall consider:

1- Data Encryption

2- Data Integrity and Authentication

3- Privacy of Sensitive Data

4- Inheritance and Access Control

In this system, sensitive data is handled with strict security measures. First, data encryption is implemented, ensuring that sensitive information is protected and cannot be directly manipulated. Second, data integrity and authentication mechanisms are in place to verify the authenticity and integrity of any updates or modifications to sensitive data. Third, privacy of sensitive data is maintained by making the relevant fields private and granting access only with proper authorization. Finally, inheritance and access control restrictions are enforced to prevent unauthorized access or modification of sensitive data, enhancing the overall security posture of the system.

10. Conclusion

In conclusion, this report has emphasized the importance of addressing software security requirements to ensure the integrity and confidentiality of the system. By analyzing potential threats, proposing effective solutions, and considering misuse scenarios, the report has laid the groundwork for developing a robust and secure software system. It is essential to implement the recommended measures and continuously evaluate and update security practices to stay ahead of evolving threats in the ever-changing cybersecurity landscape. The findings and recommendations presented in this report emphasize the necessity of prioritizing software security to protect against potential threats.