Sun Devil Security System

Group 14

Software Requirements Specification (SRS)

Group Members:

Vandita Venuturapalli

Saurabh Choudhary

Madhu Koteshwara Manjunath

Abdulaziz Alwashmi

Roshan Chaudhari

Pradeep Kumar Mani

Revision History

Date	Revision	Description	Author
09/22/14	1.0	Initial Version	Vandita Venuturapalli, Saurabh Choudhary, Madhu Koteshwara Manjunath, Abdulaziz Alwashmi, Roshan Chaudhari, Pradeep Kumar Mani
7/12/2014	2.0	Final Version	Vandita Venuturapalli, Saurabh Choudhary, Madhu Koteshwara Manjunath, Abdulaziz Alwashmi, Roshan Chaudhari, Pradeep Kumar Mani

Table of Contents

1. Purpose

The purpose of this document is to provide a high level requirement specification of the Secure Banking System. It captures the system requirements by analyzing the business workflow and the functionalities of the application. The intended audiences for this document are the developers, testers and the clients of the Secure Banking System. This document would enable them to understand all the aspects of the system in detail. These audiences are intended to maintain the document as well for future reference.

1.1. Scope

The scope of this document is restricted to specify the requirements and functionalities of the Secure Banking System. The system would support functionalities to secure various banking activities and intended to be used by clients that take up several roles ranging from an Admin, a Bank employee, Individual Customers and Merchants/Organizations.

1.2. Definitions, Acronyms, Abbreviations

1.2.1. Definitions

PKI (Public Key Infrastructure): Associates public keys to individuals with the help of a Certificate Authorities.

1.2.2. Acronyms

SBS: Secure Banking System

GUI: Graphical User Interface

SSL: Secure Socket Layer

PKI: Public Key Infrastructure

OTP: One Time Password

PII: Personally Identifiable Information

1.3. Overview

The Secure Banking System intends to facilitate online banking activities including credit, debit and transfers initiated by the customers, account management as well as supports functionalities that concern monitoring and reviewing of these activities intended for the system administrators or bank employees. The primary focus of this system is to secure all the transactions and account management activities initiated by any user providing a secure environment for all the operations performed on the system.

2. Overall Description

2.1. Product Perspective

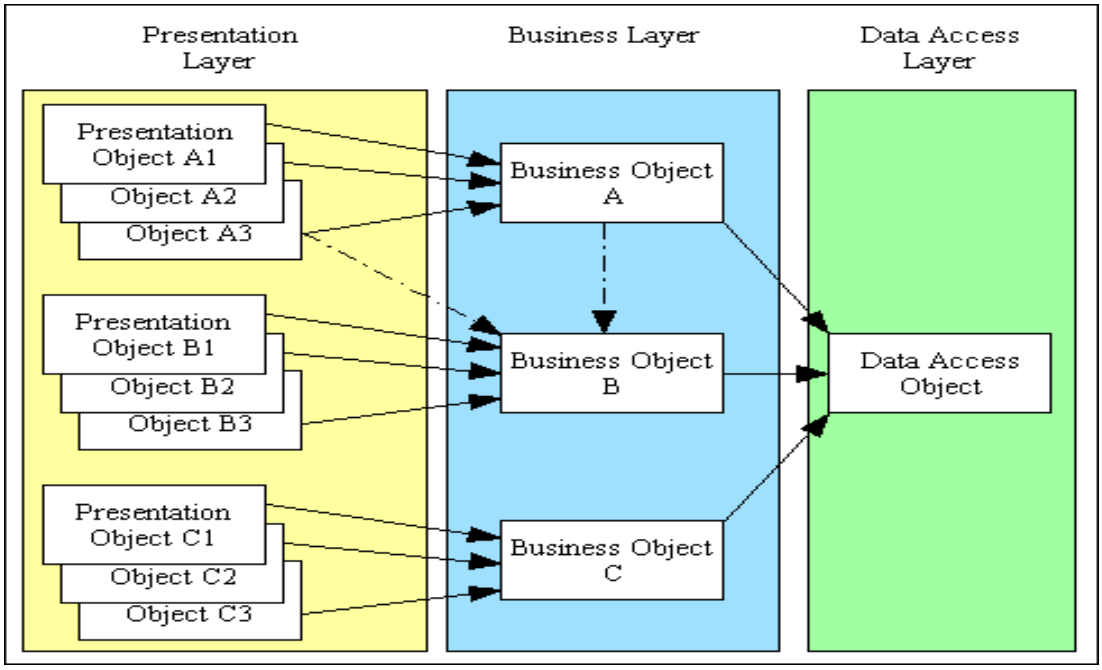
In the current world each and every product and business has online support provided for whatever is being developed. Similarly, this product is an Online Banking system service provided to handle all the Banking operations. This Product provides customers an easy way to access and manage their accounts at any moment in time. Financial transactions such as transferring money between accounts, pay bills, credit and debit fund transfer are all possible in this system.

As the Online banking system is becoming popular day by day, the numbers of threats are increasing proportionately. Hence, the need for security of valuable and confidential information and ensuring data security is very much important. In this Project, several security measures such as Public Key Infrastructure, one time password and SSL (Secure Socket Layer) have been implemented in order to ensure the security for the data within the application.

This product does not have any dependency with any other product. The end user is being provided with certain interfaces and all the data stored is created by this application is stored in an exclusive database.

2.2. Product Architecture

This Online banking system is developed as a web application and follows the 3 tier architecture. The following diagram below describes the 3 tier architecture.



- **Presentation Layer**

The user interfaces developed form the presentation layer in this application. This application can be accessed by multiple users simultaneously. The user interfaces for this Online Banking system has been designed using HTML, CSS and JSP which are compatible on most of the normal computers.

The user interfaces of this particular application consists of the Login Screen, Forgot Password, A user interface for internal users and a user interface for external users etc.,. The webpages are accessed using SSL (Secure Socket Layer) ie HTTPS connection.

- **Business layer**

The business Layer consists of Business and data rules. The components of this layer reside on a server and in this Online Banking System, Apache Tomcat Server is used as the application server. The Business layer consists of Servlets and services. Servlets act as a controller for the presentation layer and calls appropriate services based on the request coming from presentation layer. These services have the actual business logic and process the data accordingly.

- **Data Layer**

This is the layer where the database resides and can be accessed by the Business service layer. In this Online Banking system, the database is MySQL and the data is stored in a table format.

2.3. Product Functionality/Features

Following are some of the features and functionalities offered by this Online Banking System. Once, the user has been authenticated and has been logged in, the user can perform the following transactions:

- **View Account balances and Transaction history:**

The user can view his current balance and all the transaction history that was performed on his/her account.

- **Payments:**

The user (Customer) can make payments to merchants through this application. The Merchants can charge payments from other bank customers and also submit their payments to the bank. All these operations are handled securely.

- **Transfer Funds:**

Any user in the online banking system can securely transfer funds from his account to any other valid account in the bank.

- **Secure Transaction Logging**

All monetary transactions carried out by external customers or Merchants can be securely logged for external audits. The application also creates several internal transactions for various banking activities carried out by the internal bank employees such as user profile change, authorization requests, user role change etc. There internal transactions are securely logged as well.

- **Transaction Access**

An internal bank employee or an Admin can view the transactions by the bank customers with necessary authorization from the respective customers involved in the transaction.

- **Internal Transaction access**

An internal bank employee with proper privileges or an Admin can view the all the internal transactions that are being logged.

- **Account Management**

Account Management will have certain functionalities that will only be accessible to Customer if they have permission from the bank employees. They can create request for some critical activities that will be reviewed and approved by bank employees.

- **Technical account access**

Employees can access bank account of users if they sufficient permission from Customer and system admin

- **Personally Identifiable Information (PII)**

Only system admins have access to PII. Regular bank employees will have access to these only they have permission from both the customer and the system admins.

2.4. User Characteristics

The users of our application can be classified broadly into two main categories:

- Internal Users: These are the users that are bank employees. It consists of two main subcategories.
    - Employees: They carry out the day to day bank activities related to authorizing transaction, helping user with their bank account and access transactions. These users may need special permission from System Administrators to carry out certain activities.
    - Banking System Administrator: They are the people provide authorization to employees to carry out their activities. They also authorize critical transactions, can access system logs, can access PII and perform/ authorize critical activities.
- External Users: These set of users are the people who use banking services. It consists of two main subcategories

- Individual User/ Customer: View, credit, debit from bank account. They can view their and create request for transactions. They can view and edit their profile with necessary permissions.
- Merchant/Organization: They can submit payment to the bank with proper authorization and they can view credit debit transfer money from their bank account. Also they can authorize bank official request for accounts they are responsible for.

2.5. Constraints

- Users can create/view/delete a transaction only if they have permissions to do so. This applies to every user in the project.
- Some transactions may need authorization from the customer and/or bank admins.
- PII are accessible to people with proper authorization.
- Based on the type of the user there will be a constraint on what kind of user interfaces and functionalities will be available to that user.

2.6. Assumptions and Dependencies

- There will be some preexisting users in the system that can play the role of bank admins. These users will be master users and can authorize requests/Transactions.
- Functionalities such as one time password (OTP) will use email as the method of communication.
- Logs will keep track of the activities of the users of the system.
- Some functionality related to credit/debit can be handled by a separate user interface that will not be accessible to users the system. They are only provided for programmers.

3. Specific Requirements

3.1. Functional Requirements

Specify required behavior and include parameters such as response times, throughput times, other timing constraints, sequencing, accuracy, capacities, priorities, etc.

A regular bank employee:

1. Bank employee can view, create, modify, delete, and authorize transactions with proper authorization from users and merchants.

2. Bank employee can access transactions with authorization form users and system administrator.

Banking system Administrator:

1. Administrator can verify users' requests.

2. Administrator can add/delete/modify internal users' account

3. Administrator can access system log file

4. He can access PII with necessary authorization/ request from government agencies

5. Administrator can authorize critical transactions

6. An Administrator can modify/delete/add user accounts with necessary request from external user and regular bank employee.

Individual user:

1. User can view, debit, credit and transfer money from his personal bank account

2. User can initiate modification personal information change or transactional review.

3. User can authorize bank official requests to review transactions on accounts he is responsible for.

Merchant/Organization:

1. Merchant/Organization can submit merchants/users payment to the bank with proper authorization from merchants/users.

2. Merchant/Organization can view, debit, credit and transfer money from merchant's bank account

3. Merchant/Organization can initiate modification in personal information or can request transaction review.

4. Merchant/Organization can authorize bank official request to review transaction on accounts the merchant is responsible for.

3.2. External Interface Requirements

State any External interface requirements. These requirements may exist in greater detail in an Interface description document, or other requirements document. Reference these documents if they exist.

- There should be separate tabs for merchant and customer for checking account details, transferring/ making payment, transfer between the accounts, credit/debit money, view/authorize payment request, view profile and account management.
- There should be separate tabs for internal user for view and modify profile, view/authorize transactions, account management, and for viewing transaction log.
- There should be separate windows for sign in, sing up, creating new account.

3.3. Internal Data Requirements

- All the transactions of bank and other user data are stored in the database.
- Database is managed by MySQL drivers.
- All these databases are accessed by employee and administrator only.

3.4. Design and Implementation Constraints

- This application is entirely developed in Java.
- MySQL is used for database and hosted on Apache Tomcat Server.
- Git or SVN is used for versioning control.

4. Non-Functional Requirements

4.1. Security and Privacy Requirements

- The access must be funneled through a choke point where credentials are required to pass.
- SSNs, Account numbers, and sensitive information must be encrypted using a public, private key pair from both sides.
- One time password (OTP) must be used to authenticate emails, sensitive transactions, and signup process.
- OTP must only be sent to a verified email of the user.
- All communications must be encrypted using HTTP, SSL handshakes and encryption mechanisms.
- Pre-shared master key must be used to avoid MIM attacks.
- Nonce (challenge) must be used to avoid reply attacks.
- Every user must be sand boxed to be only able to access own accounts.
- All sensitive transactions must be authorized by an internal user.
- Highly sensitive transactions must authorized by an administrator.
- Request pages must expire once requests are complete.

4.2. Computer Resource Requirements

4.2.1. Computer Hardware/Software Requirements

- The requirements for our system are the same for Java 7,
- http://java.com/en/download/help/sysreq.xml plus,
- MySQL
- Apache Tomcat
- Web browser with JVM-Java7 support.

4.2.2. Computer Communication Requirements

All communications must be:

- Secure using HTTPS encryption techniques and attack prevention techniques provided by SSL.
- Continuous, reliable, and fault tolerant.
- Server should always be available to legitimate users.

4.3. Software Quality Factors

- Security: System must always be secured at all states.
- Availability: System must always be available to legitimate users.
- Reliable: database querying must be fast. Time complexity of Algorithms must be as low as possible.
- Efficiency: system must be reliable, fast, secure, and available.

- Usability: System must look and feel easy to use for novice users.

4.4. Precedence and Criticality of Requirements

- Security.
- Availability.
- Reliability.
- Usability.

5. Qualification Provisions

This section defines a set of qualification methods:

A regular bank employee: Requirements	Qualification Requirement
Bank employee can view, create, modify, delete, and authorize transactions with proper authorization from users and merchants.	Demonstration, Testing
Bank employee can access transactions with authorization form users and system administrator.	Testing
Banking system Administrator: Requirements	
Administrator can verify users' requests.	Testing
Administrator can add/delete/modify internal users' account	Testing
Administrator can access system log file	Testing
He can access PII with necessary authorization/ request from government agencies	Testing
Administrator can authorize critical transactions	Testing
Administrator can modify/delete/add user accounts with necessary request from external user and regular bank employee.	Testing
Merchant/Organization : Requirements	
Merchant/Organization can submit merchants/users payment to the bank with proper authorization from merchants/users.	Testing
Merchant/Organization can view, debit, credit and transfer money from merchant's bank account	Demonstration, Testing
Merchant/Organization can initiate modification in personal information or can request transaction review	Testing

Merchant/Organization can authorize bank official request to review transaction on accounts the merchant is responsible for	Demonstration, Testing
Individual User: Requirements	
User can view, debit, credit and transfer money from his personal bank account	Demonstration, Testing
User can initiate modification personal information change or transactional review	Testing
User can authorize bank official requests to review transactions on accounts he is responsible for.	Demonstration, Testing

6. Requirements Traceability

6.1. Upward Traceability

The below traceability relates each requirement outlined in this document with the test cases outlined in the test plan document. Please refer to the test plan for more details regarding the test cases:

SRS Section Number	Requirement	Test Cases
3.1.1.1	Regular user access control	16-21,27
3.1.2	System Admin access control	22-26
3.1.3	Regular Customer Access control	1-15
3.1.4.	Merchant access control	1-15
3.1.1.2	Authorization Requests	10,11
3.1.4.2	Credit/Debit	8,9
3.1.3	Transfer Funds	5,6
3.1.4.1	Payments	10,11
3.6.5	PKI	38
3.1.7	OTP	38
3.1.6	Technical account access	23-26
3.1.5	Transaction access	2-11
3.1.4.4	Security Features	39

7. Security functionalities

1. **One-Time-Password (OTP):** One time password (OTP) was implemented to avoid bots from signing up and to avoid illegitimate forgot password requests. The OTP functionality was incorporated in the sign up routine; after a user tries to sign up, they will be prompted to a screen to enter their SSN and email. An OTP will be sent if the preceding information is correct.

2. **Virtual Keyboard:** Virtual keyboard was implemented in the sign in routine to avoid the capture of the user's keystroke. This enhances our safeguarding system for passwords.

3. **SSL (Secure Sockets Layer):** All requests to the server are run through an HTTPS channel. No HTTP request is allowed. A PEM file was generated containing a self-signed certificate and an RSA private key. The Tomcat server was configured using this file.

4. **Recaptcha:** Re-captcha is incorporated to avoid bots from sending requests to the server; hence, avoiding DoS and DDoS attacks. This functionality was incorporated in the sign up routine.

5. P**assword hashing + salting:** All the passwords entered into the system are hashed and an additional layer of salting is being added. When a sign in request is received, the hash value of the entered password is verified against the hash value stored in the database. No plaintext password is directly stored.

6. **Authentication, Authorization, and Access Control:** Users are being authenticated via username and password pairs, as well as digital signatures in the case of PKI. Only authorized users are allowed to enter into the system. Each user is only allowed to operate within their space, based on their role.

7. **Public Key Infrastructure (PKI):** At signup, a key pair is generated. The public key is saved in the database. The private key is saved in a special space assigned for the user (We are aware that the private key is supposed to be supplied to the user, but we specified the special spaces for each user to simulate our understanding of PKI in the time we had). DSA was used to implement the signing and verification. The user's space on the server simulates the idea of the private key being safely stored within user's access only. The user signs information in the make a payment request using their private key. The signature is then verified using the stored public key. If the signature is correct the PKI interface returns true to signal the controller that the information is authenticated. Otherwise, false is returned to the controller to signal the signature was failed to be authenticated.

8. **Server side validation:** All fields in all forms are validated using well designed regular expressions. This validation is always done server side. In addition to server side validation, some forms implement client side validation to improve usability and security. However, we never have the case where only client side validation is present, server side validation is always incorporated. This is to avoid XSS and SQL injection attacks.

8. Assumptions

1. No firewall exceptions on client machine that prevents a request from being posted.
2. Internet connection is stable.
3. Vlab machine is uninterrupted.
4. Password should have the following properties:
    a. It should be of at least 8 characters long
    b. It should have at least one small case letter [a-z]
    c. It should have at least one upper all case letter [A-Z].
    d. At least one special character.
    e. At least one digit [0-9].
5. External user and internal users are different in their roles. An internal user cannot have an external user account.
6. Sign up screen has the following validations.
    a. First Name, last name of the user should be start with upper case and should have no numbers.
    b. Username should be only be alphanumeric and cannot contain special characters and spaces.
    c. Phone number is a 10 digit phones number and cannot contain – and +.
7. Access of web pages depends on the type of user logged in. For ex. An internal user cannot access pages of an external user and vice versa. A page access denied would be showed if the user tries to access pages which are not defined for his role.
8. After there is a Successful Sign up, a message is being shown "User Registered Successfully. Please Log out and Login again".
9. In the Accounts page for external user, we will not show any Account if there is not any account defined for that user. A new user account is created with a Zero balance.
10. Credit and debit works without authorization for an amount lesser than or equal to
11. After Signup, the user needs to add an account to perform all the transactions.
12. When an Internal User works on a task they should either do the necessary operation or click cancel. If an Internal User go to other screen or log out without doing any of the task that task would not be fetched again and it would be lost.
13. If the user enters a wrong OTP, he would be redirected back to entering his email without any message. This is because he has entered a wrong OTP. This is on Sign up and Forgot password when he receives the OTP.
14. In Signup, when an existing user in the database, is trying to Sign up, he is redirected back entering his email again without any message. This is only because his email already exists in the database. Once he enters an email which does not exist in the database, then will be able to Sign up.