**Risk Driven V&V of**

**Bank Application Software System**

**Priyanka Deoskar**

**Devang Parekh**

**Sagar Chavan**

**Jobin Kanichirayil George**

**Liang Shen**

**CISC 593-50- B-2021/Fall**

**HU**

**Table of Contents**

Contents

**1.** **System Introduction**

**2.** **Risk Model**

**3.** **Failure Paths**

**4.** **Architecture Model**

**5.** **Context Model**

**6.** **Use – Case Models**

**7.** **Requirements V&V**

**8.** **Design V&V**

**9.** **Test Cases and V&V**

**10.** **Future work**

# 

# 

# 

# 

# 

# 

# 

# 

# **1.** **System Introduction**

**Bank Application**

A simple bank application where users can deposit money, view their balances and other transactions.

**Unique Hardware/Software needs to develop and implement this system**

**Hardware Needs :**

The user should have the app installed on their local machine. The app shall work for Python 3.9 and newer versions. The system running the application has to be maintained, monitored and patched.   
**Software Needs :**  
We need the software to be kept updated with the latest version code. The application is currently written in python and we plan on developing newer features using the same. There are going to be unit and system tests designed around it and we plan on using CI CD platforms to manage rollout and automation.

**Features/capabilities**

* **View balance:** The user should be able to view their bank account balance.
* **Deposit Cash :** The user should be able to deposit money to his account.
* **Withdraw Cash :** The user should be able to withdraw the specified amount from their account. The withdrawal amount is checked to be a valid number meaning that the withdrawal amount should be less than the account balance.
* **Transaction History :** The app keeps track of the previous transactions carried out by the user for the future potential requirements regarding credit or loan and lease needs.
* **Credit Transactions :** The application authenticates and authorizes a users/account holders credit transaction based on credit available.
* **Authentication:** Have a user enter credentials to start the application before they can use any features.

**Safety Concerns**

From user’s perspective (when this product is being used)

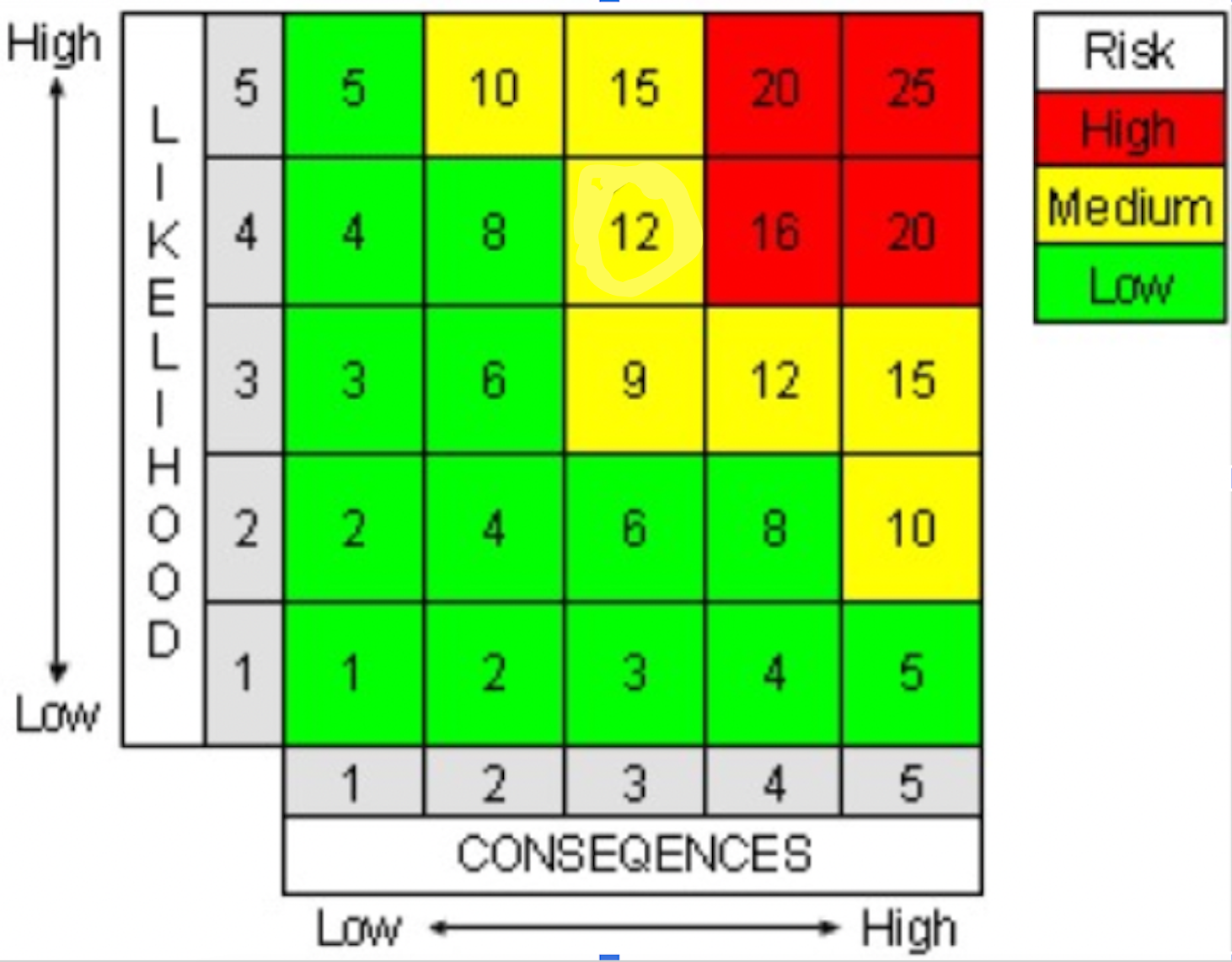
# **2.** **Risk Model**

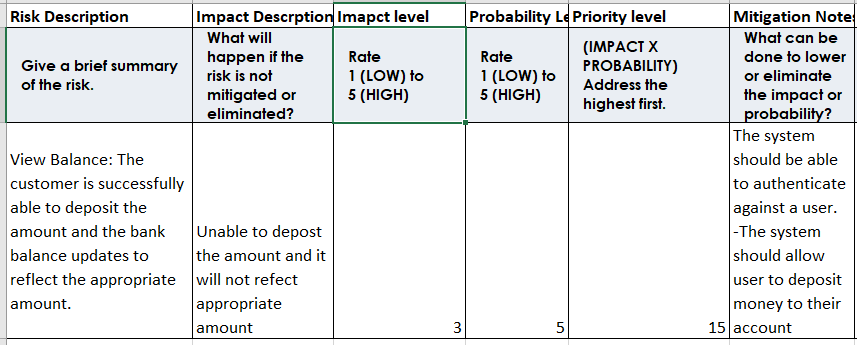
Risk assessments involve

· Identification:Determine and document the hazards associated with the use of the system

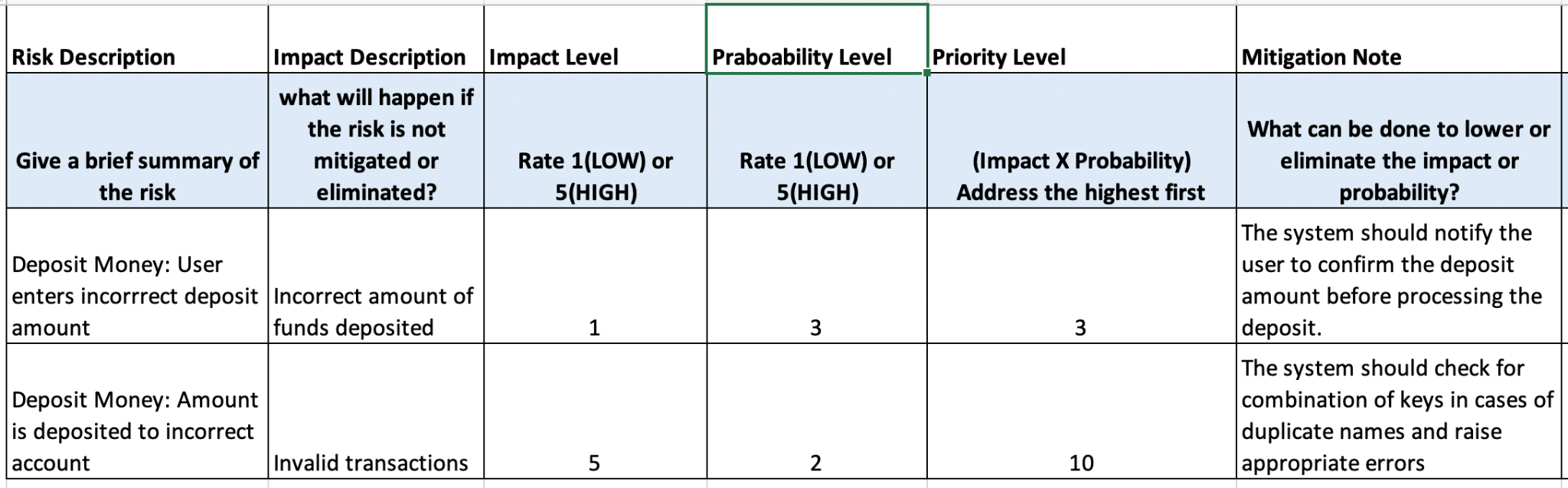
· Evaluation:Assess the severity and probability of the identified hazards.

· Classification:Categorize the risks according to severity and probability.Document the classifications.



**View Balance:**  


**Deposit Money:**



**Credit Transactions:**

| Risk # | Description | Likelihood | Impact | Likelihood x Impact Score | Mitigation Strategy |
| --- | --- | --- | --- | --- | --- |
| 1 | Card number is not valid or is not present in the system | 1 | 4 | 4 | Given the low likelihood we accept the risk, and reject the transaction |
| 2 | Card security pin is not valid | 3 | 2 | 6 | Reject the transaction and inform the user of incorrect transaction |
| 3 | Card holders name is not valid | 1 | 2 | 2 | Bank accepts the risk while informing the user to review before final approval |
| 4 | Insufficient balance | 2 | 5 | 10 | If a customer is in good standing(external logic) and less than a certain threshold, accept risk else Reject transaction. |

# **3.** **Failure Paths**

Failures that could happen

1. Users enter the wrong credentials and it does not start the application.   
   It would ask the user to enter the credentials again.
2. Withdraw more money than the balance in the account.   
   If the user tries to withdraw more money than the current balance in the account it would result in a failure path.
3. User tries to use more than the provisioned credit amount on the account.  
   If a user tries to use more credit money than the provisioned amount it should result in a failure path.
4. Card holders' security pin and/or card number is not valid then the credit transaction will be rejected.
5. Card holder doesn’t have enough credit to complete the transaction then the transaction will fail.
6. In case of multiple users with the same name, the transactions like deposit money may lead to error conditions.
7. If the user enters incorrect deposit amount or uses invalid characters then that would lead to errors or the user may need to reenter the value.

# **4.** **Architecture Model**

Software Modules - Each of the features that we discussed previously are going to act like individual software modules.

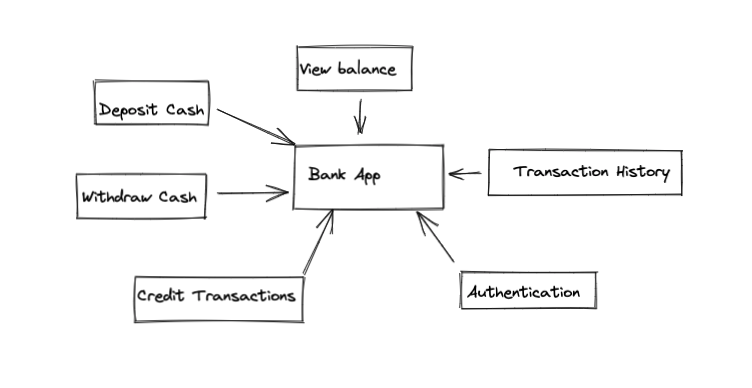
1. View Balance
2. Deposit Cash
3. Withdraw Cash
4. Credit Transactions
5. User Authentication
6. Transaction History

Assumptions:

Some of the assumptions are

1. The end user knows the credentials and will be able to enter them in three attempts
2. The system is reliable and always up and running

# **5.** **Context Model**



# **6.** **Use – Case Models**

| Name: View Balance | |
| --- | --- |
| Actors | Customer |
| Description | Allow customers to view the bank balance. |
| Main Success Scenario | The customer is successfully able to check their bank balance. |
| Exceptions |  |
| Alternative paths |  |

**Table 1**

| Name: Deposit Amount | |
| --- | --- |
| Actors | Customer |
| Description | Allow Customer to deposit a specific amount to their bank balance. |
| Main Success Scenario | The customer is successfully able to deposit the amount and the bank balance updates to reflect the appropriate amount. |
| Exceptions | If the customer enters a negative value for the deposit amount. |
| Alternative paths |  |

**Table 2**

| Name: Withdraw Amount | |
| --- | --- |
| Actors | Customer |
| Description | Allow Customers to withdraw a specific amount from their bank balance. |
| Main Success Scenario | The customer is successfully able to withdraw the amount and the bank balance deducts that value to reflect the appropriate amount. |
| Exceptions | If the customer enters an amount that is beyond the withdrawal limit. |
| Alternative paths |  |

**Table 3**

| Name: Transaction History | |
| --- | --- |
| Actors | Customer |
| Description | The app tracks the previous transactions of the user when the user starts using the app. |
| Main Success Scenario | The app records the customer’s transactions for the duration of running the application. |
| Exceptions |  |
| Alternative paths |  |

**Table 4**

| Name: Credit Transactions | |
| --- | --- |
| Actors | Customer |
| Description | Allow customer to make a credit transaction and the app approves or rejects the transaction based on a set credit limit. |
| Main Success Scenario | The customer is able to successfully make a credit transaction as long as the transaction amount is within the credit limit. |
| Exceptions | If the transaction amount goes beyond the credit limit then the transaction fails. Authentication for the card fails. |
| Alternative paths |  |

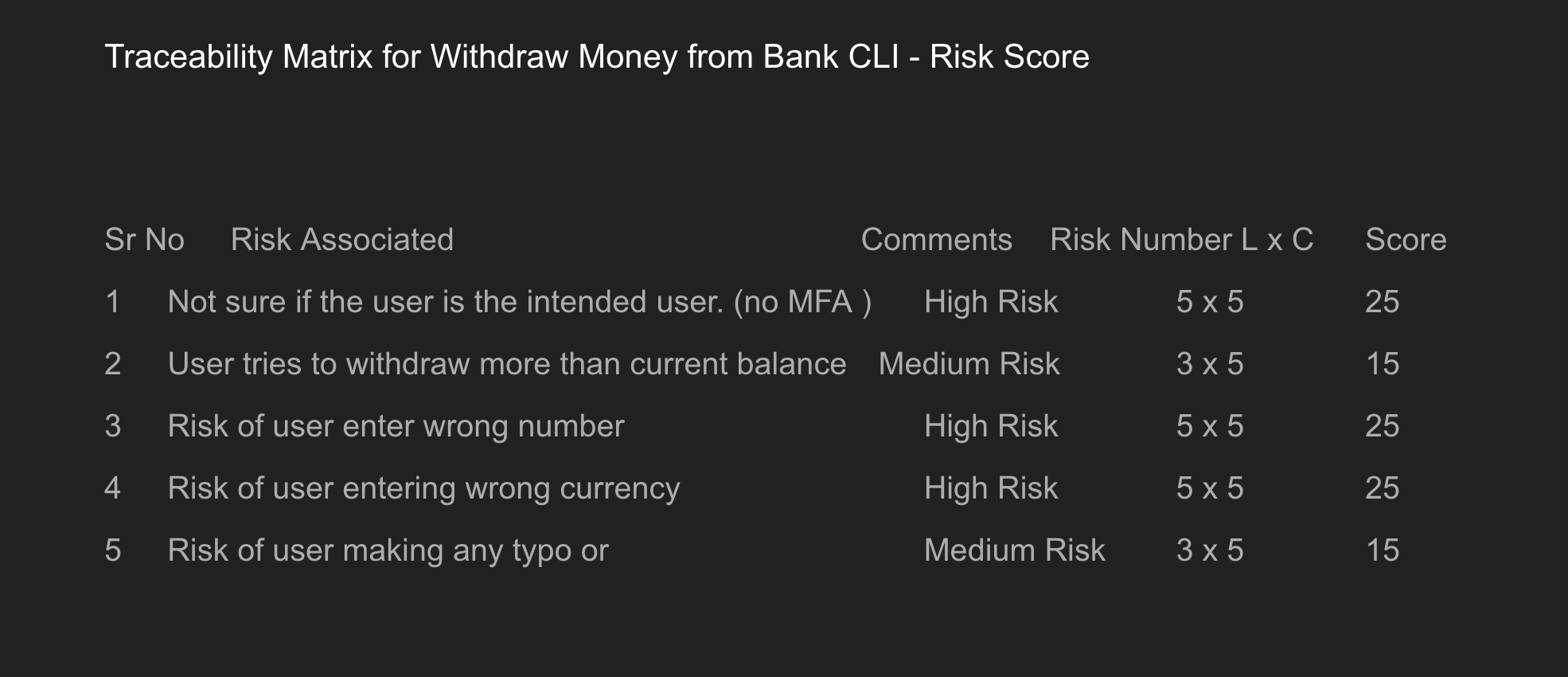
**Table 5**

| Name: Authentication | |
| --- | --- |
| Actors | Customer |
| Description | Allow customer to put a pin to access the bank application. |
| Main Success Scenario | The customer is allowed to put a password pin for 3 times. After the third time, if the customer’s input pin is incorrect then the customer is prevented from trying for the fourth time for 30 seconds. After this duration, the customer can try putting the input pin again. |
| Exceptions |  |
| Alternative paths |  |

**Table 6**

**7.** **Risk-Driven Requirements V&V**

**Capability-1: Withdraw Money**



1. Mitigation strategy and Risk Involved to Withdraw money
2. 1. We can reduce the impact of the high risk by having a multi factor authentication feature implemented in the banking application.
3. 2. It is a medium risk if the user tries to withdraw more than the balance amount form the bank. We should ideally place a check that does not permit this operation, and also does not crash the banking application.
4. 3. Another risk is that a user entering the wrong number, currency or space will either result in the wrong amount being withdrawn by the user, or it can result in a cancelled transaction. We want to avoid either of these by having enough input validations so that the user does not enter wrong text.
5. 4. Alternatively the user should be provided with an option to correct the entered text in case they change their mind or find out there was a mistake while trying to withdraw the amount.

Requirements for Withdrawing money feature.

1. 1. We need to have a hard requirement on the authentication feature set up
2. 2. We need to ensure the withdrawal operations are tested end to end and that illegal withdrawal operations are not allowed. We need to set up testing plans so that each operation can be tested.
3. 3. Ensure we have enough testing plans set-up to make sure all inputs are validated. We will have to

**Capability-2:**

2.1 - The system should allow user to view previous account balance

**Mitigation strategy and Risk Involved to View Balance**

1. Authentication is the highest risk associated with view balance.It could be reduced by using MFA.Everyone uses mobile application for banking purpose so Face recognition one method to reduce the risk. Email notification or phone message notifications are integrated for MFA. This would help to prevent unauthorized access.

2. Current View balance should work when the user click on it.Application should not be frozen. It should bring accurate balance.All transaction history should be up to date and it should be tracked

**Requirements for View Balance.**

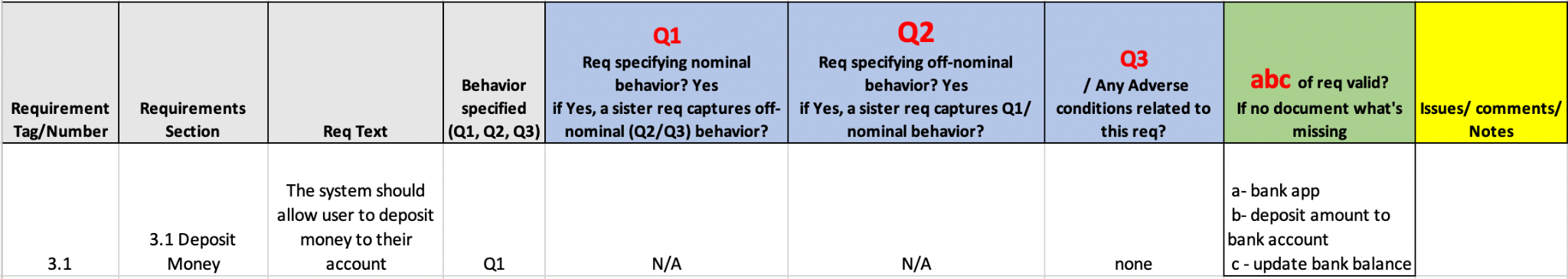
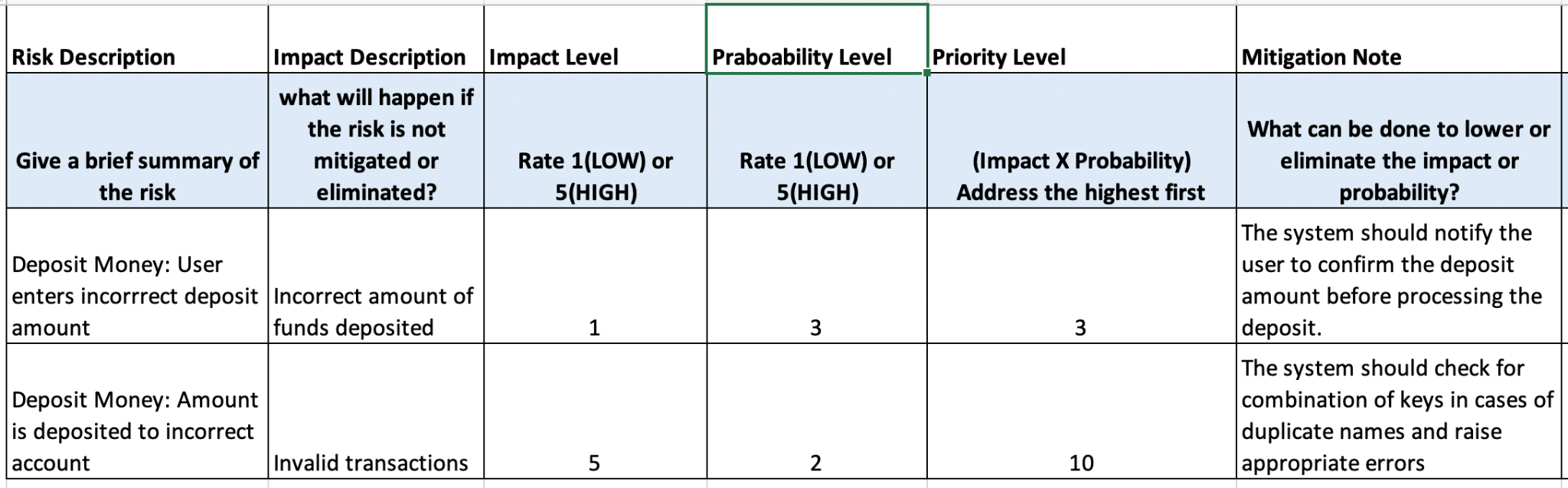
1. MFA feature should be incorporated, and it should be monitored

2. Transactions should be tracked and updated so that accurate balance would be displayed.

| SR No: | Risk Associated | Comments | Risk Number L\*C | Score |
| --- | --- | --- | --- | --- |
| 1 | User Authentication | High Risk | 5\*5 | 25 |
| 2 | Clicking View balance button | Medium Risk | 3\*5 | 15 |
| 3 | Recent Transaction related risk | Medium Risk | 3\*5 | 15 |

**Capability-3: Deposit Money  
Risk-Driven Requirement:**

3.1 - The system should allow user to deposit money to their account

**Traceability Matrix:**  
  
**Risk Model:**  


**Mitigation Strategy for Risks Involved with Deposit Money**

1. A high impact risk is related to colliding usernames that may lead to error conditions or invalid transactions. We can mitigate this risk by putting plenty of checks for a combination of keys to ensure that the input amount is deposited to the appropriate account and if need be then raise relevant errors so the user can be notified.
2. Another risk is that a user may enter the wrong value, currency or invalid characters that may result in the wrong amount being deposited by the user. The mitigation strategy is to avoid such a condition by implementing enough input validations so that the user does not enter a wrong value.

**Capability-5:**

5.1 - The system tracks user’s previous transactions

**Capability-6:**

6.1 - The system approves and rejects user’s credit transactions based on the credit limit

| Requirements Section | Req Text | Behavior specified (Q1, Q2, Q3) | Q1 | Q2 | Q3 | ABC | Issues |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | Validate the provided authentication and authorization parameters | Q1 | valid credentials | invalid credentials | None | a: Bank, b: validates credentials c: given the input credentials |  |
| 2 | Process payments for given transaction | Q2 | approve | reject | Invalid credentials, invalid balance | a: Bank b: approve transaction c: given right credentials and if enough balance is available | customer should have enough balance and right credentials |
| 3 | Maintain credit balance | Q1 | provide available | NA | NA | a: Bank b: maintains the credit balance: credentials |  |
| 4 | Show credit transaction history | Q1 | list of transaction | None |  | a: Bank b: returns the list of transaction |  |
| 5 | Ability to evaluate good behavior score for overdraft transaction from external API | Q3 | None | None |  | a: Bank b: calls external API to determine if customer is on good behavior |  |

Mitigation Strategy:

1. Reject all transactions which do not provide the right pin & card number pair for each transaction.
2. Every Transaction has to be associated with a credit card number and pin number.
3. All transactions need to be checked against available credit balance.

# **8.** **Design V&V**

We have designed the system by keeping the following things in mind:

* Updates
* Requirements
* System Analysis
* System Design
* Code Design
* Testing
* Deployment
* Maintenance

**9.** **Test Cases and V&V**

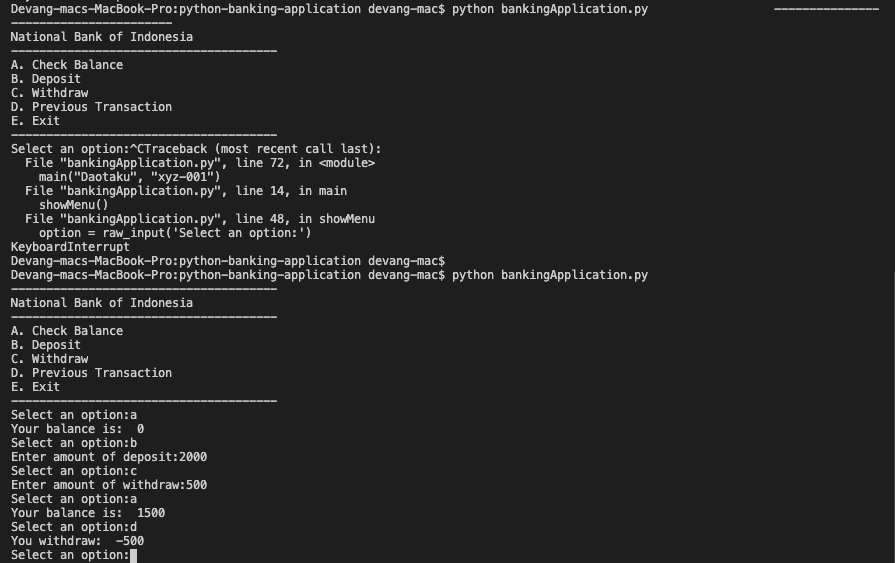
**Heartbeat:**

| Requirement number | function | complete | priority | Description | Inputs | Expected results |
| --- | --- | --- | --- | --- | --- | --- |
| 1a | User Authorization |  | 1 | User should input a password pin. | Incorrect Pin | App will prompt the user to re-enter the pin. |
| 1b | User Authorization |  | 1 | User should input a password pin. | 4th Input Pin Try | The user will be prevented to re-enter another pin for 30seconds. |
| 2a | Deposit Money |  | 2 | User should input a deposit amount. | Input includes only a numerical value and no special characters. | The account balance should get updated and reflect the new balance amount. |
| 2b | Deposit Money |  | 2 | User should input a deposit amount. | Input includes special characters. | The transaction should fail. |
| 3a | Withdraw Money |  | 3 | User should input a deposit amount. | Input includes only a numerical value and no special characters. | The account balance should get updated and reflect the new balance amount. |
| 3b | Withdraw Money |  |  | User should input a withdraw amount. | Input value exceeds the withdrawal limit. | The transaction should fail. |
| 4a | View Balance |  | 4 | User should select the view balance option | Input value is a string matching the menu option for ‘view balance’. | User should be able to view the current account balance. |
| 5a | Credit Transactions |  | 5 | User should maintain enough credit balance | Input value exceeds the credit limit | The credit transaction should fail. |
| 5b | Credit Transaction |  | 5 | A transaction should be accompanied by a valid pin and card number combination | Input pin and credit card number should match | Ther credit transaction should fail |

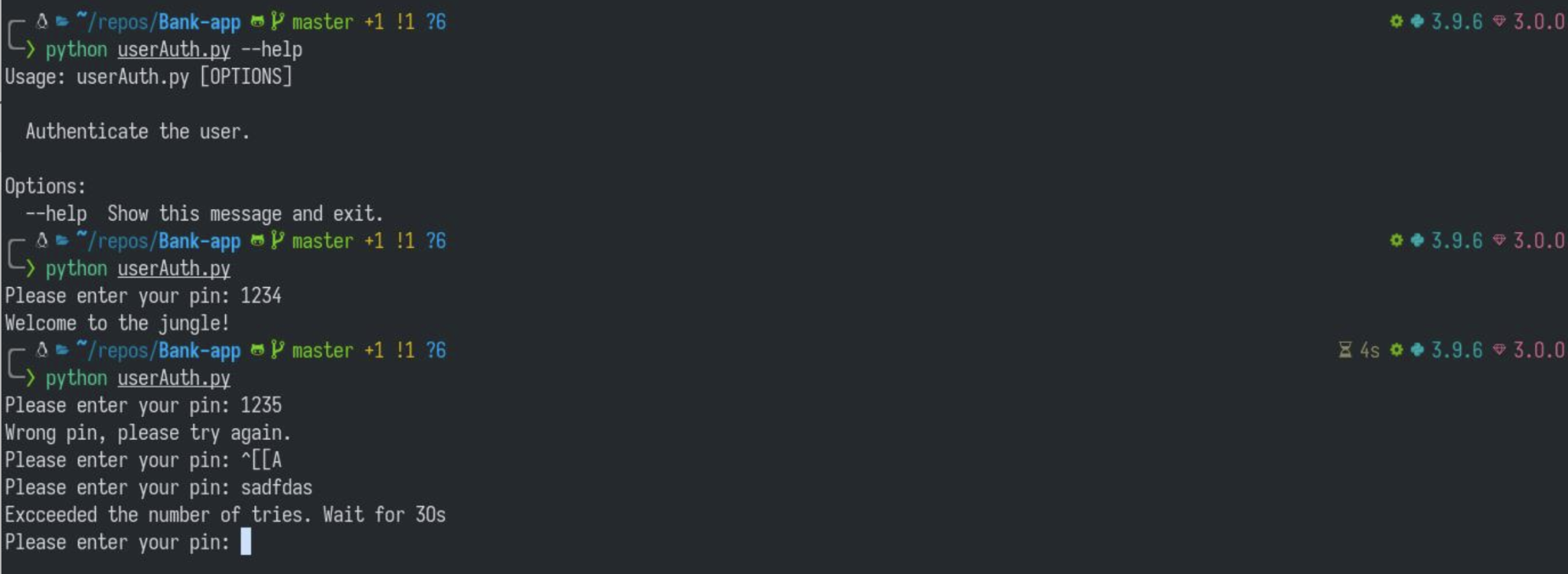
Table 9 shows the tests cases

**Screenshots and User Interaction**

**Features**

****

User Authentication:

****

**10.** **Future work**

1. We want to have user transaction history populated for the last 6 months or so
2. We want to have a user registration process setup eventually where the user upon signing up will have profile and meta data like first name, last name etc.
3. Have multiple test cases that cover corner cases.
4. Enhance security features