**Credit Fraud Detection Using the Hidden Markov Model**

Team Members:

Sergio Gabriel Jiawei Kun ; s\_kun@mail.fhsu.edu

Lee Joseph Judkins ; ljjudkins@mail.fhsu.edu

Project Github: https://github.com/kunsergio117/CreditFraudDetectionHMM.git

Instructor: Prof. Hieu Vu

**Revision History:**

|  |  |
| --- | --- |
| Part 1 | 23/09/2024 |
| Part 2 | 27/09/2024 |

# Table of Contents

[Table of Contents 2](#_Toc178373585)

[Section 3:  Use Cases 3](#_Toc178373586)

[A: Stakeholders 3](#_Toc178373587)

[B: Actors and Goals 3](#_Toc178373588)

[C: Use Cases 3](#_Toc178373589)

[B. Use Case Diagram 5](#_Toc178373590)

[Section 4:  User Interface Specification 12](#_Toc178373591)

[A. Preliminary Design 12](#_Toc178373592)

[B. User Effort Estimation 15](#_Toc178373593)

[Project Management 18](#_Toc178373594)

[References 20](#_Toc178373595)

# Section 3:  Use Cases

## A: Stakeholders

1. **Users:** The users would be the credit card holders who benefit from the protection against fraud.
2. **Fraud Analysts:** The individuals who would be responsible for reviewing flagged transactions and investigating potential fraud.
3. **Financial Institutions:** Banks, credit card companies, and other online retailers who would be responsible for managing and securing transactions.
4. **System Administrators:** Those people who are responsible for maintaining and updating system infrastructure.
5. **Developers:** The technical team responsible for development, deployment, and maintenance of the HMM-based fraud detection system.
6. **Regulatory Bodies:** Government or financial regulatory authorities ensuring compliance with fraud prevention standards.

## B: Actors and Goals

1. **Initiating Actors:** 
   1. **Credit Card Holder:** Their goal is to make secure transactions without being blocked or flagged incorrectly.
   2. **Fraud Analyst:** Their goal is to quickly and efficiently review flagged transactions. They will then confirm if it is fraud and take action.
   3. **System Administrator:** Their goal is to maintain the system, ensuring continuous functionality, and ensure regular maintenance.
2. **Participating Actors:**
   1. **Transaction System:** These processes transactions and interacts with the HMM for fraud detection.
   2. **HMM Model:** Identifies suspicious transactions based on behavioral patterns.

## C: Use Cases

UC1: **UploadTransactionData**

* Description: The user uploads a CSV file containing historical transaction data.
* Actors: Credit Card Holder, System Administrator
* Requirements: Addressed in REQ1, REQ5

UC2: **MonitorTransactionsFraud**

* Description: The HMM-based system analyzes incoming transactions in real-time, identifying anomalies that may indicate fraudulent activity.
* Actors: Transaction System, HMM Model
* Requirements: Addressed in REQ2, REQ3, REQ4

UC3: **ManualTransactionReview**

* Description: A fraud analyst reviews flagged transactions manually to determine their legitimacy.
* Actors: Fraud Analyst
* Requirements: Addressed in REQ6, REQ8, REQ13

UC4: **GenerateFraudReports**

* Description: The system generates detailed reports of fraudulent activity, including metrics such as detection accuracy and false positives.
* Actors: Fraud Analyst, System Administrator
* Requirements: Addressed in REQ5

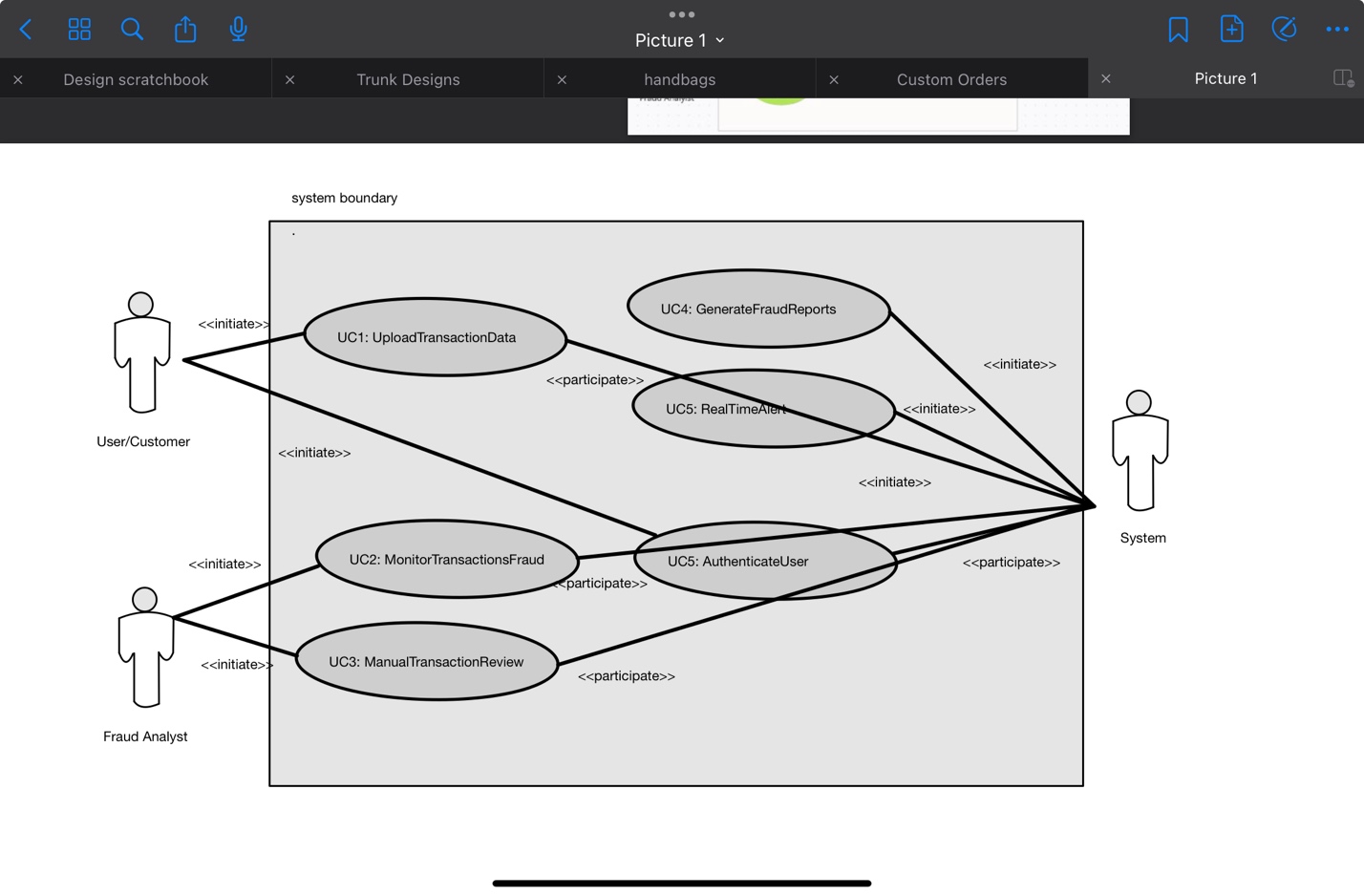
UC5: **RealTimeAlert**

* Description: The system sends alerts to users and fraud analysts when suspicious transactions are detected.
* Actors: Credit Card Holder, Fraud Analyst
* Requirements: Addressed in REQ3, REQ11, REQ12

UC6: **AuthenticateUser**

* Description: The user logs into the system by providing their credentials (username and password). The system verifies the credentials to grant access to authorized users and denies access to unauthorized users.
* Actors: Credit Card Holder, System Administrator
* Requirements: Addressed in REQ7, REQ9, REQ10

## D. Use Case Diagram

****

**C. Traceability Matrix**

|  |  |  |
| --- | --- | --- |
| **Requirement ID** | **Use Case Name** | **Priority 1-5** |
| REQ 1 | **UploadTransactionData** | **5** |
| REQ 2 | **MonitorTransactionsFraud** | **5** |
| REQ 3 | **MonitorTransactionsFraud/**  **RealTimeAlerts** | **4** |
| REQ 4 | **MonitorTransactionsFraud** | **5** |
| REQ 5 | **UploadTransactionData/**  **GenerateFraudReports** | **3** |
| REQ 6 | **ManualTransactionReview** | **4** |
| REQ 7 | **AuthenticateUser** | **5** |
| REQ 8 | **ManualTransactionReview** | **2** |
| REQ 11 | **RealTimeAlerts** | **4** |
| REQ 12 | **DataPrivacy** | **2** |
| REQ 13 | **ManualTransactionReview** | **4** |

**D: Fully Dressed Descriptions**

1. **Use Case: Upload Transaction Data**

* **Primary Actor:** Credit Cardholder, Fraud Analyst
* **Goal:** Upload CSV file containing transaction data for analysis.
* **Preconditions:** Users must be authenticated and have a valid CSV file with transaction data.
* **Main Flow of Events:**
  1. The user logs in and navigates to the upload page.
  2. The user selects the CSV file containing transaction data.
  3. The system validates the file format and structure.
  4. If the file is valid, it is uploaded, and the data is processed.
  5. If invalid, the system provides an error message indicating the problem.
* **Postconditions:** Transaction data is stored in the system for analysis.
* **Exceptions:** If the file is incorrectly formatted, the system rejects the upload and provides instructions for correction.

**2. Use Case: Real-Time Alerts**

* **Primary Actor:** Credit Cardholder, Fraud Analyst
* **Goal:** Notify users of suspicious activity when a transaction is flagged as potentially fraudulent.
* **Preconditions:** The system must be monitoring incoming transactions in real-time.
* **Main Flow of Events:**
  1. A transaction occurs and is processed by the system.
  2. The HMM flags the transaction as anomalous.
  3. The system sends an alert via email or SMS to the user and fraud analyst.
  4. The alert contains transaction details such as the amount, merchant, date, and reason for flagging.
  5. The user or analyst reviews the alert and takes appropriate action (e.g., freeze the account, investigate further).
* **Postconditions:** Users and analysts are informed of potentially fraudulent activity and can take action.
* **Exceptions:** If the system fails to send an alert (e.g., due to network failure), the transaction remains flagged for manual review.

**3. Use Case: Generate Fraud Reports**

* **Primary Actor:** Fraud Analyst
* **Goal:** Create downloadable reports that summarize system performance and fraudulent activity.
* **Preconditions:** The system must have accumulated transaction and fraud detection data.
* **Main Flow of Events:**
  1. The analyst navigates to the report generation page.
  2. The analyst selects parameters for the report (e.g., date range, transaction type).
  3. The system retrieves relevant transaction data.
  4. The system generates a report summarizing fraud detection statistics (e.g., false positives, true positives, accuracy).
  5. The analyst downloads the report in PDF or CSV format.
* **Postconditions:** A report is generated and available for download.
* **Exceptions:** If no data is available for the selected parameters, the system informs the analyst and allows them to adjust the criteria.

**Use Case: Manual Transaction Review**

* **Primary Actor:** Fraud Analyst
* **Goal:** Review and validate transactions flagged as suspicious.
* **Preconditions:** The system must have flagged transactions based on the HMM model.
* **Main Flow of Events:**
  1. The analyst accesses the list of flagged transactions.
  2. The analyst filters or sorts transactions based on criteria such as date, amount, or merchant.
  3. The analyst selects a transaction to review.
  4. The system displays transaction details and metrics that led to the flagging.
  5. The analyst makes a decision to either mark the transaction as legitimate or confirm it as fraudulent.
  6. The system updates the transaction’s status accordingly.
* **Postconditions:** The flagged transaction is resolved, and the decision is logged in the system.
* **Exceptions:** If the analyst is unable to make a decision, the transaction remains flagged for further review.

**D: System Sequence Diagrams**

**UC1:**

[Credit Card Holder] [Fraud Detection System]

| |

| 1. Upload Transaction Data (CSV) |

|-------------------------------------------------->|

| |

| 2. Validate and Store Data |

|<--------------------------------------------------|

| |

| 3. Acknowledge Success/Failure |

|-------------------------------------------------->|

**UC2:**

[Fraud Detection System] [HMM Model]

| |

| 1. Send Incoming Transaction Data |

|------------------------------------------------>|

| |

| 2. Analyze Transaction for Anomalies |

| 3. Return Fraud Detection Results |

|<------------------------------------------------|

| |

**UC3:**

[Fraud Analyst] [Fraud Detection System]

| |

| 1. Request Flagged Transactions |

|------------------------------------------------->|

| |

| 2. Provide List of Flagged Transactions |

|<-------------------------------------------------|

| |

| 3. Review Transaction Details |

| 4. Confirm/Reject Fraudulent Transaction|

|------------------------------------------------->|

| |

| 5. Update Transaction Status |

|<-------------------------------------------------|

**UC4:**

[Fraud Analyst] [Fraud Detection System]

| |

| 1. Request Fraud Report |

|----------------------------------------------->|

| |

| 2. Generate Report |

| |

| 3. Return Fraud Report |

|<-----------------------------------------------|

UC5:

[Fraud Detection System] [User]

| |

| 1. Detect Suspicious Transaction |

| 2. Send Alert Notification |

|----------------------------------------->|

**UC6:**

[User] [Fraud Detection System]

| |

| 1. Provide Credentials (username, password) |

|----------------------------------------------------------->|

| |

| 2. Validate Credentials |

|<-----------------------------------------------------------|

| |

| 3. Access Granted or Access Denied |

|----------------------------------------------------------->|

# Section 4:  User Interface Specification

## A. Preliminary Design

**UC1 - Upload Transaction Data**

Step-by-Step Interaction:

1. User Authentication Screen:

- Display Fields:

- Username (Input Field)

- Password (Input Field)

- Buttons:

- Login (Submit Button)

Navigation:

- User enters credentials and clicks the Login button.

Mock-Up:

+---------------------+

| User Authentication |

+---------------------+

| Username: [\_\_\_\_\_\_\_\_] |

| Password: [\_\_\_\_\_\_\_\_] |

| |

| [Login] |

+--------------------------------+

2. Upload Transaction Data Screen:

- Once logged in, the user is directed to the main dashboard.

- The user selects the Upload Transaction Data option.

Fields:

- File Upload Button (CSV format only)

Buttons:

- Upload (Submit Button)

- Cancel (Button to cancel upload)

Navigation:

- User navigates to the Upload page, selects a CSV file, and clicks the Upload button.

Mock-Up:

+--------------------------------+

| Upload Transaction Data |

+--------------------------------+

| Select file: [Browse...] |

| |

| [Upload] |

| [Cancel] |

+--------------------------------+

3. Result Display:

- After a successful upload, the system displays a message:

+--------------------------------+

| Upload Successful |

+--------------------------------+

| Your transaction data |

| has been successfully |

| uploaded and is under |

| analysis. |

+--------------------------------+

4. Error Display (if invalid):

- If an error occurs, an error message appears.

+--------------------------------+

| Upload Error |

+--------------------------------+

| The uploaded file is |

| incorrectly formatted. |

| Please check your CSV |

| file and try again. |

+--------------------------------+

**UC3 - ManualTransactionReview**

Step-by-Step Interaction:

1. Transaction Review Page:

- After logging in, the user navigates to ManualTransactionReview.

Display Fields:

- Filter Options (Transaction ID, Date, Amount, Merchant)

Buttons:

- Search (Submit Button)

- Reset Filters (Button to clear filters)

Navigation:

- User enters filter criteria and clicks the Search button.

Mock-Up:

+-----------------------------------------+

| Manual Transaction Review |

+-----------------------------------------+

| Transaction ID: [\_\_\_\_\_\_\_\_] |

| Date: [\_\_\_\_\_\_\_] to [\_\_\_\_\_\_\_] |

| Amount: [\_\_\_\_\_\_\_\_] |

| Merchant: [\_\_\_\_\_\_\_\_] |

| |

| [Search] |

| [Reset Filters] |

+-----------------------------------------+

2. Review Transaction Results:

- The system displays a list of filtered transactions, highlighting flagged transactions.

Mock-Up:

+-------------------------------+

| Flagged Transactions |

+-------------------------------+

| ID | Amount | Status |

|--------|-------------|-------------|

| 12345 | $200 | Suspicious |

| 67890 | $50 | Legitimate |

| 11223 | $500 | Suspicious |

+-------------------------------+

| [Select Transaction] |

+-------------------------------+

3. Detailed View:

- Upon selecting a flagged transaction, detailed metrics appear.

Mock-Up:

+--------------------------------------------------+

| Transaction Details - ID 12345 |

+--------------------------------------------------+

| Amount: $200 |

| Merchant: Apple Store |

| Date: 2024-09-27 |

| Reason for Flagging: Anomaly detected |

| |

| 1. Mark as Legitimate [Button] |

| 2. Confirm Fraudulent [Button] |

+--------------------------------------------------+

## B. User Effort Estimation

UC1 - UploadTransactionData

Step-by-Step Flow of Events:

1. User navigates to the login page.

- Mouse Clicks: 1 (Click on the login option)

- Keystrokes: 0

2. User enters Username and Password.

- Mouse Clicks: 0

- Keystrokes: 2 (Username and Password)

3. User clicks the Login button.

- Mouse Clicks: 1

- Keystrokes: 0

4. User navigates to the Upload Transaction Data section.

- Mouse Clicks: 1 (Click on Upload Transaction Data option)

- Keystrokes: 0

5. User clicks the Browse button to select the CSV file.

- Mouse Clicks: 1

- Keystrokes: 0

6. User selects the CSV file from the file dialog and clicks Open.

- Mouse Clicks: 1

- Keystrokes: 0

7. User clicks the Upload button to submit the chosen CSV file.

- Mouse Clicks: 1

- Keystrokes: 0

8. User reviews the success or error message displayed by the system.

- Mouse Clicks: 0

- Keystrokes: 0

Total Effort Calculation for UC1:

- Total Mouse Clicks: 6

- Total Keystrokes: 2

- User Interface Navigation: 5 clicks (Clicking Login, Upload Transaction Data, Browse, and Upload)

- Clerical Data Entry: 2 keystrokes (Username and Password)

Fraction of User Effort:

- User Interface Navigation: (5 clicks / 8 total actions) \* 100 = 62.5%

- Clerical Data Entry: (2 keystrokes / 8 total actions) \* 100 = 25%

UC3 - ManualTransactionReview

Step-by-Step Flow of Events:

1. User navigates to the login page.

- Mouse Clicks: 1 (Click on the login option)

- Keystrokes: 0

2. User enters Username and Password.

- Mouse Clicks: 0

- Keystrokes: 2 (Username and Password)

3. User clicks the Login button.

- Mouse Clicks: 1

- Keystrokes: 0

4. User navigates to the Manual Transaction Review section.

- Mouse Clicks: 1 (Click on Manual Transaction Review option)

- Keystrokes: 0

5. User enters filter criteria (Transaction ID, Date, Amount, Merchant).

- For each field (assuming all are filled once):

- 4 fields x 4 keystrokes (average for typing per field)

- Mouse Clicks: 0

- Keystrokes: 16 (for the four filter criteria)

6. User clicks the Search button to display filtered transactions.

- Mouse Clicks: 1

- Keystrokes: 0

7. User reviews the list of flagged transactions.

- Mouse Clicks: 0

- Keystrokes: 0

8. User selects a flagged transaction for detailed review.

- Mouse Clicks: 1

- Keystrokes: 0

9. User reviews details of the flagged transaction.

- Mouse Clicks: 0

- Keystrokes: 0

10. User casts a decision to mark the transaction as legitimate or fraudulent.

- Mouse Clicks: 1 (Click on the corresponding button)

- Keystrokes: 0

Total Effort Calculation for UC3:

- Total Mouse Clicks: 5

- Total Keystrokes: 18

- User Interface Navigation: 4 clicks (Clicking Login, Manual Transaction Review, Search, Select Transaction)

- Clerical Data Entry: 18 keystrokes (for filter criteria)

Fraction of User Effort:

- User Interface Navigation: (4 clicks / 23 total actions) \* 100 = 17.39%

- Clerical Data Entry: (18 keystrokes / 23 total actions) \* 100 = 78.26%

# Project Management

Both team members will be actively engaged in the development process, utilizing the GitHub repository to track progress and contributions clearly.

Responsibilities will be delegated based on each member's strengths and expertise, with a shared accountability structure ensuring that all aspects of the project are covered. This is because our group consists of only a pair and thus realistically we will both need input and validation from one another in all development areas.

Below is our projected milestones, with week 1 representing the week beginning in 23 Sept 2024.

| **Week** | **Task Description** | **Responsible Team Members** |
| --- | --- | --- |
| Week 1 | Project kick-off meeting, define scope and objectives | Sergio Kun, Joseph Judkins |
| Week 2 | Research HMM algorithms and relevant literature | Sergio Kun |
|  | Familiarization with data sources and datasets | Joseph Judkins |
| Week 3 | Develop initial design and architecture of the application | Sergio Kun, Joseph Judkins |
| Week 4 | Implement CSV transaction data upload feature | Sergio Kun |
|  | Initial development of the user authentication process | Joseph Judkins |
| Week 5 | Develop the manual checking function for transaction validity | Sergio Kun |
| Week 6 | Implement the alerting function for suspicious transactions | Joseph Judkins |
| Week 7 | Integrate the simulated fraudulent transaction feature | Sergio Kun |
| Week 8 | Testing functionality and debugging | Sergio Kun, Joseph Judkins |
| Week 9 | Develop and integrate reporting tools for transaction summaries | Joseph Judkins |
|  | User interface refinement and user experience enhancements | Joseph Judkins |
| Week 10 | Conduct user testing and gather feedback | Sergio Kun, Joseph Judkins |
| Week 11 | Finalize features based on user feedback | Sergio Kun, Joseph Judkins |
| Week 12 | Prepare project presentation and documentation | Sergio Kun, Joseph Judkins |
| Week 13 | Project review and adjustments based on tutor feedback | Sergio Kun, Joseph Judkins |
| Week 14 | Final project deployment and presentation to the class | Sergio Kun, Joseph Judkins |

# References