Hamdard University

Department of Computing

Final Year Project

SecuroGuard

E-commerce Fake Review Detection App

(FYP-011/FL24)

Software Design Specifications

Submitted by

Faarah Khan (2578-2021)

Hafsa Nisar (1988-2021)

Supervisors

Mr. Afzal Hussain

Co- Supervisors

Mr. Maaz Ahmed

Fall 2025

1.1.1 Document Information

Project Title	SecuroGuard: E-commerce Fake Review Detection Application
Project Code	FYP-011/FL24
Document Name	Software Requirements Specifications
Document Version	2.0
Document Identifier	Project Code-SRS
Document Status	~~Draft~~ / Final
Author(s)	Faarah Khan Hafsa Nisar
Approver(s)	Mr. Afzal Hussain Mr. Maaz Ahmed
Issue Date	17/01/2025

Name	**Role**	**Signature**	**Date**
Faarah Khan	Team Lead		17/01/2025
Hafsa Nisar	Team Member 2		17/01/2025
Mr. Afzal Hussain	Supervisor		17/01/2025
Mr. Maaz Ahmed	Co-Supervisor		17/01/2025
Mr. Faheem Ahmed	Project Coordinator		17/01/2025

Revision History

Date	Version	Description	Author
15/12/2024	1.0	First Draft	Faarah Khan, Hafsa Nisar
17/01/2025	2.0	Second Draft	Faarah Khan

Definition of Terms, Acronyms, and Abbreviations

Terms	Description
SRS	Software Requirement Specification: A document that defines the functional and non-functional requirements of the software system.
API	Application Programming Interface: A set of protocols and tools that allows different software applications to communicate with each other.
HTTP	Hypertext Transfer Protocol: A protocol used for transferring web pages on the internet.
HTTPS	Hypertext Transfer Protocol Secure: An extension of HTTP that ensures secure communication by encrypting data with SSL/TLS.
ML	Machine Learning: A branch of AI that allows computers to learn and make decisions based on data without explicit programming.
NLP	Natural Language Processing: A field of AI that focuses on enabling computers to understand and process human languages.
UI/UX	User Interface / User Experience: UI refers to the design and layout of user interfaces, while UX is focused on the overall experience of the user interacting with the system.
SSL	Secure Socket Layer: A security protocol that provides encrypted communication between web servers and browsers.
URL	Uniform Resource Locator: The address used to access resources on the web.
AI	Artificial Intelligence: The simulation of human intelligence in machines designed to think and act like humans.
Fake Reviews	Reviews that are intentionally misleading or deceptive, often posted to promote a product or service or discredit competitors.
Sentiment Analysis	A technique in NLP that analyzes and determines the sentiment (positive, negative, or neutral) expressed in text data, such as reviews.

Terms	Description
Text Mining	The process of extracting useful information from unstructured text data, often used in fake review detection to find patterns or anomalies in reviews.
Anomaly Detection	A technique used to identify patterns in data that do not conform to expected behavior, often used to identify fake reviews based on unusual review patterns.
Review Spam	Reviews that are irrelevant or repetitive, often posted in large numbers to manipulate the product or service's rating or reputation.
Reviewer Profile	A profile of a user who posts reviews, including their posting history and behavior, used to detect suspicious or fake review patterns.
Crowdsourcing	The practice of obtaining data or input by soliciting contributions from a large group of people, often used in verifying the authenticity of reviews.
Authorship Attribution	The process of determining the author of a piece of text, which can help identify whether reviews come from the same or suspicious authors.
Review Verification	Techniques used to verify the authenticity of a review, such as checking for inconsistencies or using machine learning models to assess likelihood of fakeness.
Machine Learning Model	A mathematical model used to analyze and predict data trends; in fake review detection, it's used to predict whether a review is genuine or fake based on patterns in data.
Deep Learning	A subset of machine learning involving neural networks with many layers, useful in detecting complex patterns in data such as fake review behavior.
Feature Extraction	The process of identifying and selecting relevant features (like sentiment, keywords, or reviewer behavior) for use in training machine learning models for fake review detection.
Classification Algorithm	An algorithm that categorizes data into predefined groups, such as identifying reviews as real or fake, based on features extracted from the text.
Data Labeling	The process of tagging data (e.g., reviews) with predefined labels, such as "fake" or "real," which is used to train machine learning models.
Social Proof	The concept that people are influenced by others' opinions and reviews, which can be manipulated in fake review detection systems.
Review Velocity	The rate at which reviews are posted for a product or service, used to detect suspicious activity or sudden bursts of fake reviews.
Reviewer Behavior Analysis	Analyzing a reviewer's activity and history to detect patterns that might indicate fake or paid reviews, such as reviewing multiple products in a short time.

Table of Contents

1. Introduction

1.1 Purpose of Document

The purpose of this document is to define the technical and functional design of the "Ecommerce Fake Review Detection Application." It outlines the system's architecture, methodology, functionality, and implementation approach to ensure a comprehensive and efficient development process. The intended audience for this document includes project stakeholders, software engineers, data scientists, and system integrators who will be involved in the design, development, and deployment phases of the project.

The project employs an **object-oriented design methodology (OOD)** to model the system components as real-world objects, ensuring modularity, reusability, and scalability. This approach allows for effective management of system complexity and supports the development of a robust solution.

1.2 Intended Audience

The document is intended for the following groups:

- **Project Team Members:** Developers, data scientists, and system architects involved in creating and maintaining the application.
- **Quality Assurance (QA) Team:** Testers responsible for ensuring system reliability and accuracy.
- **Stakeholders:** Individuals or organizations interested in the development and impact of the system.
- **System Administrators:** Personnel responsible for managing and deploying the application.
- **Academic Supervisors:** Evaluators and instructors overseeing the project progress.

1.3 Document Convention

The document follows these formatting conventions:

- **Font:** Calibri
- **Font Size:** 12-point for body text, 14-point bold for section headers □
- **Line Spacing:** 1.15
- **Bullets and Numbering:** Used to organize lists and enhance readability
- **Figures and Tables:** Properly labeled and referenced within the text

1.4 Project Overview

The "E-commerce Fake Review Detection Application" aims to improve the reliability of online reviews by leveraging advanced machine learning and natural language processing (NLP) techniques. The system detects and flags fake reviews by analyzing review content for authenticity indicators. Users input a product URL from e-commerce platforms, and the system performs the following tasks:

- **Scrape Review Data:** Extract reviews from the product page using web scraping techniques.
- **Analyze Reviews:** Employ NLP models to distinguish between fake and genuine reviews.
- **Display Results:** Present review authenticity scores, graphs, and detailed insights to assist users in making informed purchasing decisions.

This solution promotes transparency, enhances trust in online reviews, and supports fair competition among sellers. The system adopts a modular, object-oriented design approach to ensure maintainability, scalability, and robustness against evolving fraud detection challenges.

1.5 Scope

In-Scope:

- Detect and flag fake reviews using machine learning techniques.
- Provide real-time analysis of review authenticity.
- Allow users to input product URLs for analysis.
- Display results in tables and graphs with detailed insights.
- Maintain a user-friendly interface for seamless interaction.

Out of Scope:

- Manual review moderation by administrators.
- Direct control or deletion of reviews on e-commerce platforms.
- Integration with all e-commerce platforms (focus on major ones only).
- User-provided review writing or feedback systems beyond detecting authenticity.

2. Design Considerations

This section outlines the foundational design factors to ensure the successful implementation and operation of the E-commerce Fake Review Detection Application. These considerations will address technical, functional, and system-level requirements that influence the overall system architecture and integration strategy.

2.1 Assumptions and Dependencies

The design assumes the following factors to be true for effective system implementation:

- **Web Scraping Feasibility:** The target e-commerce platforms allow non-intrusive scraping or provide APIs for accessing review data without violating terms of service.
- **NLP Model Accuracy:** The machine learning model can effectively distinguish between authentic and fake reviews based on training datasets.
- **User Authentication:** Secure login mechanisms (e.g., encrypted passwords) ensure only authorized users access the review analysis functionality.
- **Network and Server Reliability:** Continuous availability of high-performance servers to manage large volumes of data for real-time analysis.
- **Data Privacy Compliance:** The system adheres to data protection regulations, ensuring secure handling of user data and compliance with laws like GDPR.

Dependencies critical to the design include:

- Integration with NLP libraries such as Pytorch for machine learning analysis.
- Reliable access to external data sources for review collection.
- Availability of skilled personnel for maintaining the system and updating detection algorithms.

2.2 Risks and Volatile Areas

Potential risks and areas subject to frequent change include:

- **Evolving Fraud Techniques:** Fake review generators may develop more sophisticated methods, requiring continuous enhancement of detection algorithms.
  - o **Mitigation:** Implement a model update process to regularly train the system with new data. Monitor evolving patterns in fraudulent reviews to improve detection techniques.

- **Web Scraping Restrictions:** E-commerce platforms may impose stricter access restrictions or update their website structures, impacting data extraction.
    - **Mitigation:** Develop adaptive scraping mechanisms and establish fallback solutions using available APIs where possible.
- **Technology Upgrades:** Changes in software libraries, frameworks, or tools used for development may require system modifications to remain compatible and efficient.
    - **Mitigation:** Employ modular design principles to allow for easy updates and replacements of system components without affecting overall functionality.
- **User Experience Challenges:** Ensuring a user-friendly interface while maintaining robust functionality may become challenging as additional features are integrated.
    - **Mitigation:** Conduct regular usability testing and user feedback sessions to refine the interface and improve the user experience.
- **Data Privacy Risks:** Non-compliance with evolving data privacy regulations may pose legal and operational challenges.
    - **Mitigation:** Design the system with data anonymization techniques, secure data handling practices, and routine audits for regulatory compliance.

3. System Architecture

This section provides a high-level overview of how the system is structured and decomposed into subsystems or components. The architecture outlines the interactions between these elements to fulfill the required functionality of detecting and flagging fake reviews in an ecommerce environment.

3.1 System Level Architecture

The system is decomposed into functional elements, each assigned distinct responsibilities to ensure efficient operations and a seamless user experience.

Key Components:

1. **User Interface Layer:**
   - Facilitates user interactions, including login, URL input for review analysis, and display of analysis results.
   - Designed for a responsive and intuitive user experience.
2. **Data Scraping Module:**
   - Responsible for extracting review data from e-commerce platforms via scraping or API integration.
   - Ensures compliance with platform terms and conditions through controlled data retrieval.
3. **NLP Model Integration Layer:**
   - Processes extracted review data to detect fake reviews using machine learning algorithms.
   - Employs natural language processing for semantic analysis of review content.
4. **Result Processing and Visualization Module:**
   - Aggregates and formats analysis results for user-friendly display via graphs and tables.
   - Shows authenticity scores and confidence levels.
5. **Data Storage Layer:**
   - Stores user data, scraped reviews, and analysis logs.
   - Ensures data security and privacy by implementing encryption and secure access protocols.

6. **System Administration and Monitoring:**
   - Logs system activities, tracks errors, and provides performance monitoring for optimization.

Component Relationships:

- The **User Interface Layer** communicates with the **Data Scraping Module** to retrieve user-input URLs for review analysis.
- The **NLP Model Integration Layer** receives data from the scraping module, processes it, and sends the results to the **Result Processing Module** for visualization.
- The **Data Storage Layer** manages persistent storage and retrieval operations across components.

Interface Considerations:

- Secure APIs are used for interactions between components to maintain modularity and scalability.
- External system interfaces ensure compatibility with third-party e-commerce platforms.

Design Strategies:

- Error Handling: Implement global exception handling to capture, log, and resolve system errors.
- Scalability: Design system elements for scalability to handle growing data volumes and user interactions.

3.2 Software Architecture

The software architecture follows a **three-tier architecture model**, enabling separation of concerns and modular system design.

User Interface Layer:

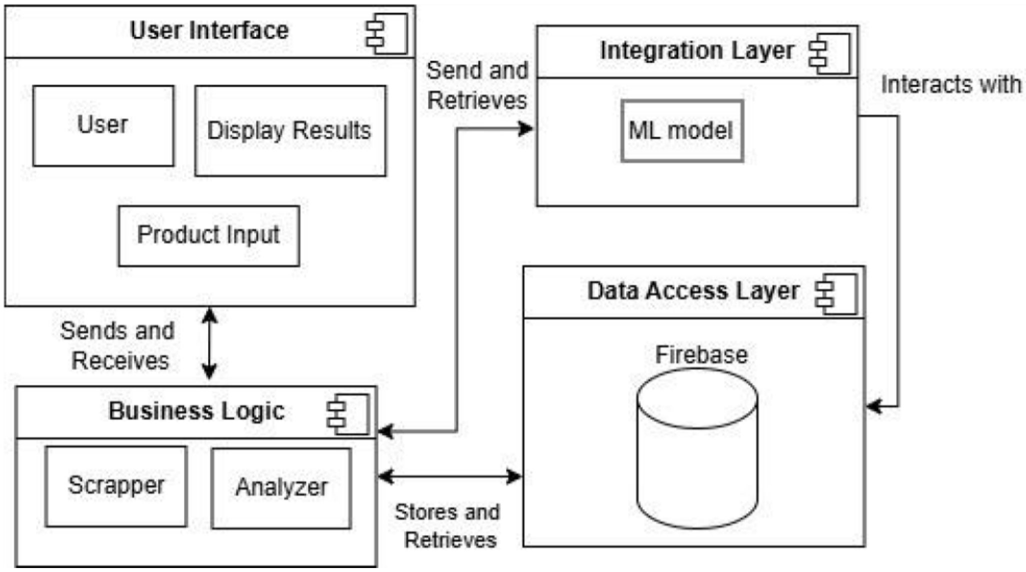
- **Function:** Manages user interaction, including login, URL input, and results display.
- **Technologies:** HTML, CSS, JavaScript, Bootstrap for front-end design.

Middle Tier:

- **Function:** Processes business logic, coordinates between the user interface and data access layers, and handles NLP operations.
- **Technologies:** Python-based APIs, TensorFlow for machine learning, Flask for application logic.

3.3. Data Access Layer:

- **Function:** Handles database interactions for storing and retrieving reviews, logs, and analysis data.
- **Technologies:** SQL databases, secure data communication protocols for encryption and access control.



4. Design Strategy

The design strategy for the E-commerce Fake Review Detection Application focuses on creating a modular, scalable, and efficient system capable of adapting to evolving e-commerce environments and fraud detection needs. The following principles and decisions drive the system design:

4.1 Future System Extension or Enhancement

- **Modular Architecture:** Components such as data scraping, NLP model processing, and result presentation are designed as independent modules to allow future updates or improvements without disrupting other system functions.
- **Model Scalability:** The system architecture supports easy integration of updated machine learning models or additional algorithms for more accurate review detection.
- **Cloud Integration:** Provisions are made for cloud-based deployment to handle increased user loads and data storage needs.

4.2 System Reuse

- **Reusable Components:** The review analysis logic and data processing modules are built to be reused across other potential platforms or projects involving text-based sentiment analysis.
- **Cross-Platform Compatibility:** Ensures compatibility with diverse e-commerce websites by abstracting the scraping logic, making it easier to adapt to changes in webpage structures.

4.3 User Interface Paradigms

- **Intuitive Design:** The user interface (UI) is designed to provide a simple and userfriendly experience, enabling users to easily input product URLs and understand the results.
- **Responsive UI:** The front end is built with responsive web design techniques to ensure accessibility across multiple devices including desktops, tablets, and smartphones.
- **Graphical Representation:** Visual elements such as graphs and tables are employed to clearly present fake vs. real review ratios and accuracy scores.

4.4 Data Management (Storage, Distribution, Persistence)

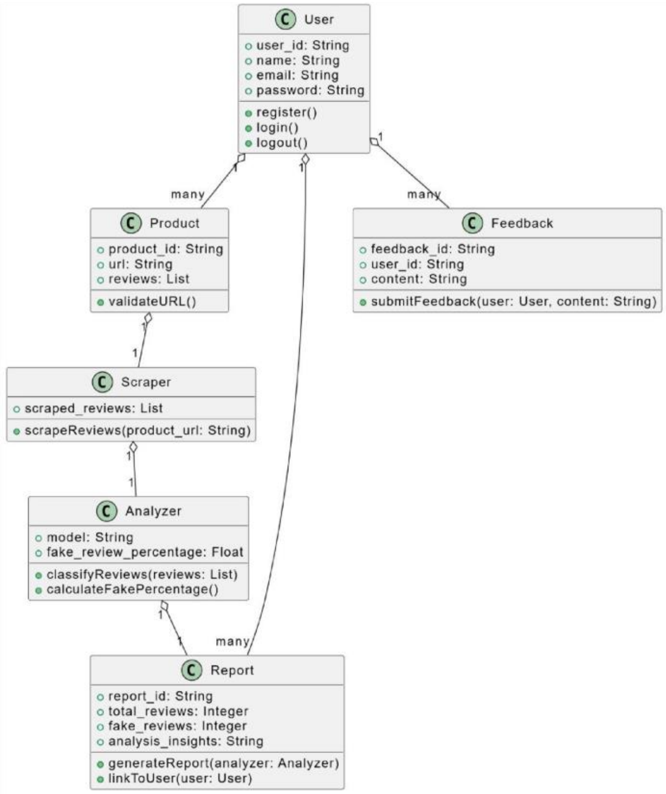
- **Centralized Data Storage:** A secure relational database stores review data, user activity logs, and analysis results, ensuring persistence and data integrity.
- **Data Security:** Strong encryption techniques are used to safeguard user data and ensure compliance with data protection regulations.
- **Efficient Data Handling:** The system implements optimized data retrieval and storage mechanisms to reduce processing latency during real-time analysis.

4.5 Concurrency and Synchronization

- **Multi-Threaded Processing:** Concurrent threads handle multiple review analysis requests, improving system performance and scalability during peak usage periods.
- **Queue Management:** A task queue system ensures efficient processing by managing simultaneous data scraping and model inference tasks.
- **Synchronization Mechanisms:** Locks and semaphores prevent data conflicts when accessing shared resources during concurrent operations.

5. Detailed System Design

5.1 Design Class Diagram



Class Diagram

5.1.1 Class Diagram Description

5.1.1.1 User:

This class represents the end-users of the application who interact with the system to analyze products for fake reviews.

5.1.1.1.1 Attributes:

user_id: A unique identifier for each user.

name, email: Basic user details for identification and communication.

password: An encrypted password for secure access.

5.1.1.1.2 Methods:

register(): Allows a new user to create an account. login():

Verifies user credentials and starts a session.

logout(): Ends the user session.

5.1.1.2 Product:

This class manages product-related data and serves as the entry point for review analysis.

5.1.1.2.1 Attributes:

product_id: Unique ID for each product. url: The product's URL,

provided by the user. reviews: A collection of reviews scraped

from the product page.

5.1.1.2.2 Methods:

validateURL(): Ensures the URL format is correct and accessible for scraping.

5.1.1.3 Scraper

The Scraper class handles the extraction of product reviews from e-commerce platforms.

5.1.1.3.1 Attributes:

scraped_reviews: Temporarily holds the data extracted from the product page.

5.1.1.3.2 Methods:

scrapeReviews(product_url): Connects to the given product URL, extracts reviews, ratings, and metadata, and stores them for analysis.

5.1.1.4 Analyzer

The Analyzer class processes the reviews to determine their authenticity using a pretrained Machine Learning Model.

5.1.1.4.1 Attributes:

model: A machine learning model trained to classify reviews as fake or real.

fake_review_percentage: The calculated percentage of fake reviews in the product’s review dataset.

5.1.1.4.2 Methods:

classifyReviews(reviews): Runs the ML model on the reviews and labels them as fake or real.

calculateFakePercentage(): Computes the proportion of fake reviews from the classified data.

5.1.1.5 Report

The Report class generates and stores the results of the review analysis.

5.1.1.5.1 Attributes:

report_id: Unique identifier for each generated report.

total_reviews: The total number of reviews analyzed. fake_reviews:

The count of reviews classified as fake. analysis_insights: Patterns

or trends derived from the review analysis, such as repetitive

language.

5.1.1.5.2 Methods:

generateReport(analyzer): Produces a report summarizing the analysis results.

linkToUser(user): Associates the report with a specific user for access or download.

5.1.1.6 Feedback

The Feedback class records user feedback regarding system performance, usability, or analysis results.

5.1.1.6.1 Attributes:

feedback_id: Unique identifier for each feedback entry. user_id:

Links the feedback to the user who submitted it. content: The

feedback text provided by the user.

5.1.1.6.2 Methods:

submitFeedback(user, content): Saves the user's feedback to the system.

5.2. Logical Data Model (E/R Model)

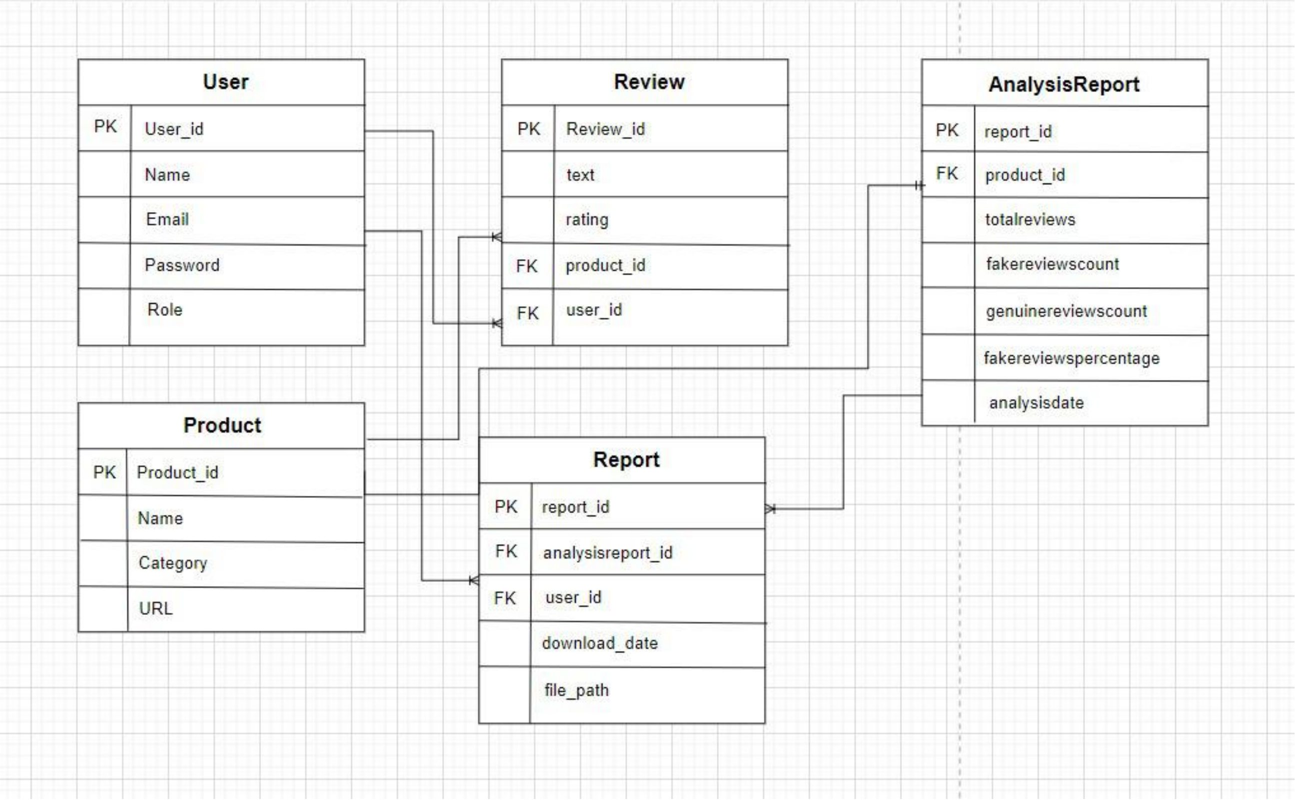
The ER model identifies entities such as User, Product, Review, and AnalysisResult, along with their relationships. This model supports efficient data storage and management.

Detailed GUIs

Home Page: Includes a text box for entering the product URL and a button to trigger the review analysis.

Results Page: Displays analysis results, including fake vs. genuine review percentages and visual representations.

5.3. Database Design



ER Diagram

ER Diagram Description

Entities and Attributes

1. **User**
   - **Purpose**: Represents the individuals using the system, specifically customers.
   - **Attributes**:
     - **UserID:** Unique identifier for each user (Primary Key).
     - **Name:** The name of the user.
     - **Role:** Indicates the role of the user, set to 'customer' by default.
2. **Product**
   - **Purpose**: Represents the items listed in the e-commerce system for which reviews are written.
   - **Attributes**:
     - **ProductID:** Unique identifier for each product (Primary Key).
     - **Name:** The name of the product.
     - Category: The category or type of the product (e.g., electronics, clothing).
3. **Review**
   - **Purpose**: Captures customer reviews of products.
   - **Attributes**:
     - **ReviewID:** Unique identifier for each review (Primary Key).
     - **ReviewText**: The actual text content of the review.
     - **ReviewRating:** Numeric rating provided for the product (e.g., 4.5).
     - **ProductID:** Foreign key linking the review to a specific product.
     - **UserID:** Foreign key linking the review to the user who wrote it.

4. **Analysis Result**
   - **Purpose**: Stores the results of the analysis performed to determine whether a review is fake or genuine.
   - **Attributes**:
     - **AnalysisID:** Unique identifier for each analysis result (Primary Key).
     - **FakeReviewCount**: The number of fake reviews identified in the analysis.
     - **GenuineReviewCount:** The number of genuine reviews identified in the analysis.
     - **ReviewID:** Foreign key linking the analysis result to a specific review.

5.4. Relationships

1. **User - Review**
    - o **Type**: One-to-Many
    - **Description**: A single User can write multiple Reviews. Each Review is associated with one specific User.

2. **Product – Review**
    - o **Type**: One-to-Many
    - **Description**: A single Product can have multiple Reviews. Each Review is associated with one specific Product.

3. **Review – AnalysisResult**
    - o **Type**: One-to-One
    - o **Description**: Each Review is linked to one AnalysisResult, which stores the results of analyzing that review.

5.5. Primary and Foreign Key Constraints

- **Primary Keys (PK)**:
    - o UserID in the User table.
    - o ProductID in the Product table.
    - o ReviewID in the Review table.
    - o AnalysisID in the
    - o AnalysisResult table.

- **Foreign Keys (FK)**:
    - o ProductID in the Review table references ProductID in the Product table.

    - o UserID in the Review table references UserID in the User table.

    - o ReviewID in the AnalysisResult table references ReviewID in the Review table.

5.6. Data Dictionary

Data 1: User Table

Column Name	Description	Type	Length	Nullable	Default Value	Key Type
UserID	Unique identifier	INT	11	No	Auto Increment	PK
Name	User name	VARCHAR	50	No	None	
Role	User role (customer)	VARCHAR	20	Yes	'customer'	

Data 2: Product Table

Column Name	Description	Type	Length	Nullable	Default Value	Key Type
ProductID	Unique product ID	INT	11	No	Auto Increment	PK
Name	Product name	VARCHAR	100	No	None	
Category	Product category	VARCHAR	50	Yes	None	

Data 3: Review Table

Column Name	Description	Type	Length	Nullable	Default Value	Key Type
ReviewID	Unique review ID	INT	11	No	Auto Increment	PK
ReviewText	Text of the review	TEXT	—	No	None	
ReviewRating	Rating of product	FLOAT	—	Yes	None	
ProductID	Linked product ID	INT	11	No	None	FK

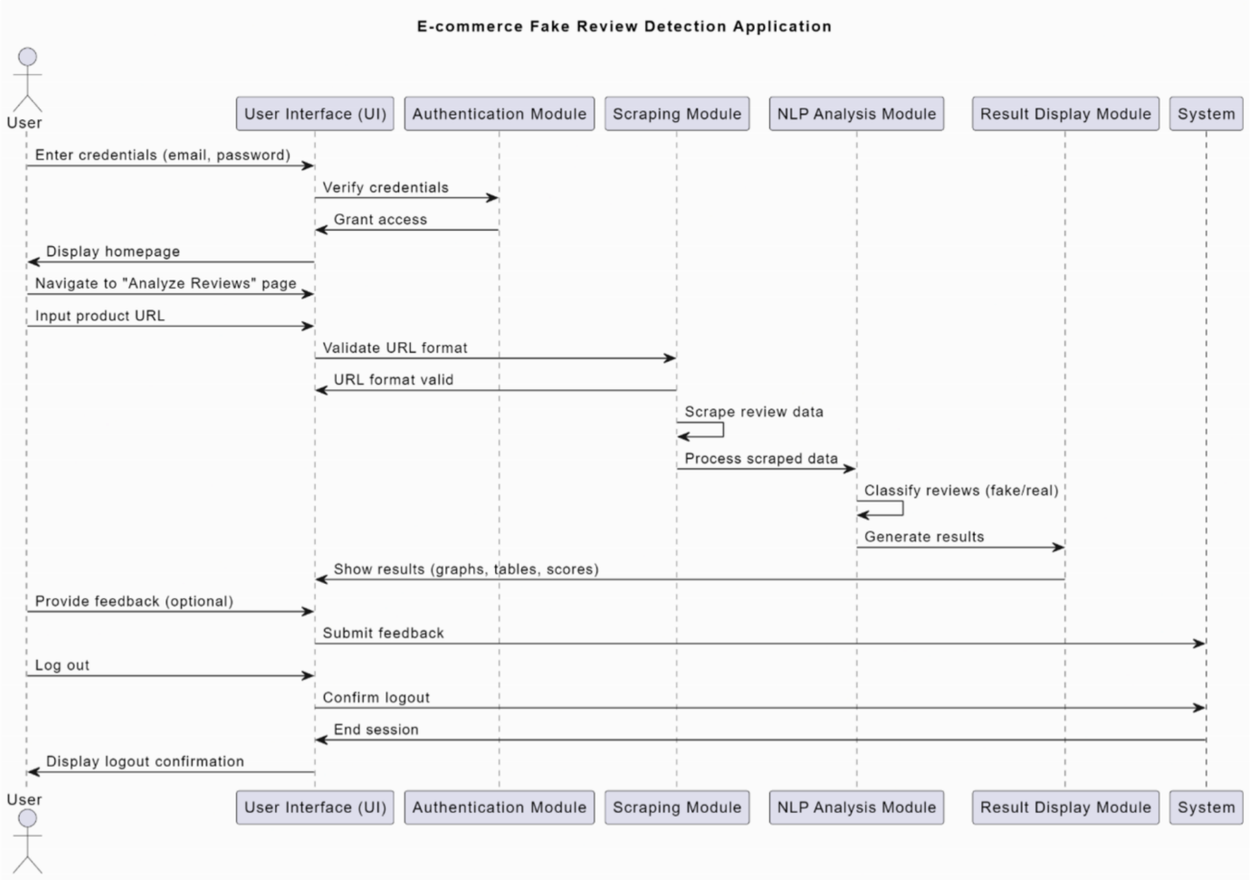
Data 4: AnalysisResult Table

Column Name	Description	Type	Length	Nullable	Default Value	Key Type
AnalysisID	Unique analysis result ID	INT	11	No	Auto Increment	PK
FakeReviewCount	Number of fake reviews	INT	—	No	None	
GenuineReviewCount	Number of genuine reviews	INT	—	No	None	
ReviewID	Linked review ID	INT	11	No	None	FK

Notation for Data Constructs

Data Construct	Notation	Meaning
Composition	=	Is composed of
Sequence	+	And
Selection	`[	]`
Repetition	{}	n repetitions of
Optional Data	()	Optional data
Comment	*...*	Delimits comments

5.7. Application Design



Sequence Diagram

Explanation:

This sequence diagram illustrates the step-by-step flow of control as a user submits a product URL, reviews are scraped from the page, and the system analyzes them for authenticity.

Actors and Entities:

- **User:** Initiates the process by providing the product page URL for analysis.
- **System UI:** Captures the URL and communicates with backend components to process the request.
- **Review Scraper:** Extracts reviews, ratings, and other relevant details from the product page URL.

- **ML Model:** Processes the extracted reviews, categorizing them as either real or fake based on pre-trained algorithms.
- **Database:** Temporarily or permanently stores the analysis results for generating reports and displaying them to the user.

Workflow:

1. The **User** enters the product URL into the system.
2. The **System UI** validates the URL format and checks accessibility.
3. Upon successful validation, the **System UI** sends the URL to the **Review Scraper**.
4. The **Review Scraper** fetches reviews, ratings, and associated data from the product page.
5. The extracted reviews are sent to the **ML Model** for classification.
6. The **ML Model** processes the reviews and identifies them as "real" or "fake."
7. Results are sent to the **Database** for temporary storage.
8. The **System UI** retrieves the analysis results from the **Database** and displays them to the

User along with options to download a detailed report.

Admin Updating the ML Model

Actors and Entities:

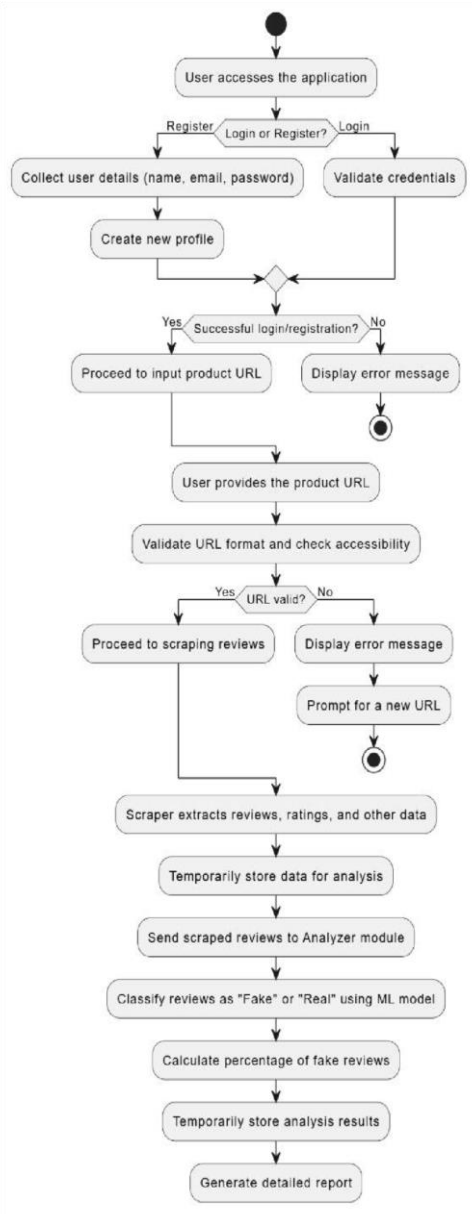
- **Admin:** The actor responsible for managing the training data and initiating the model update process.
- **System UI:** Captures the admin's input and forwards it to the backend for processing.
- **ML Trainer:** Retrains the ML model using the provided data and updates the model in the system.

Workflow:

1. The **Admin** accesses the model management interface through the **System UI**.
2. The **Admin** uploads updated training data via the **System UI**.
3. The **System UI** validates the data format and forwards it to the **ML Trainer**.
4. The **ML Trainer** processes the new training data and begins retraining the ML model.
5. Once the training process is complete, the updated model is deployed and integrated into the system.

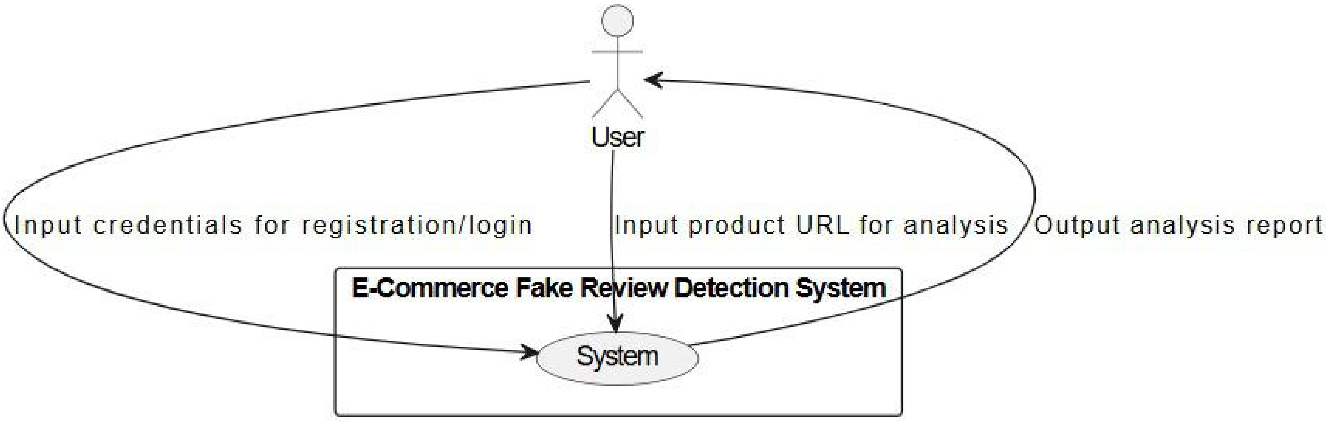
6. The **System UI** notifies the **Admin** that the ML model has been successfully updated and is ready for use.

5.8. State Transition Diagrams

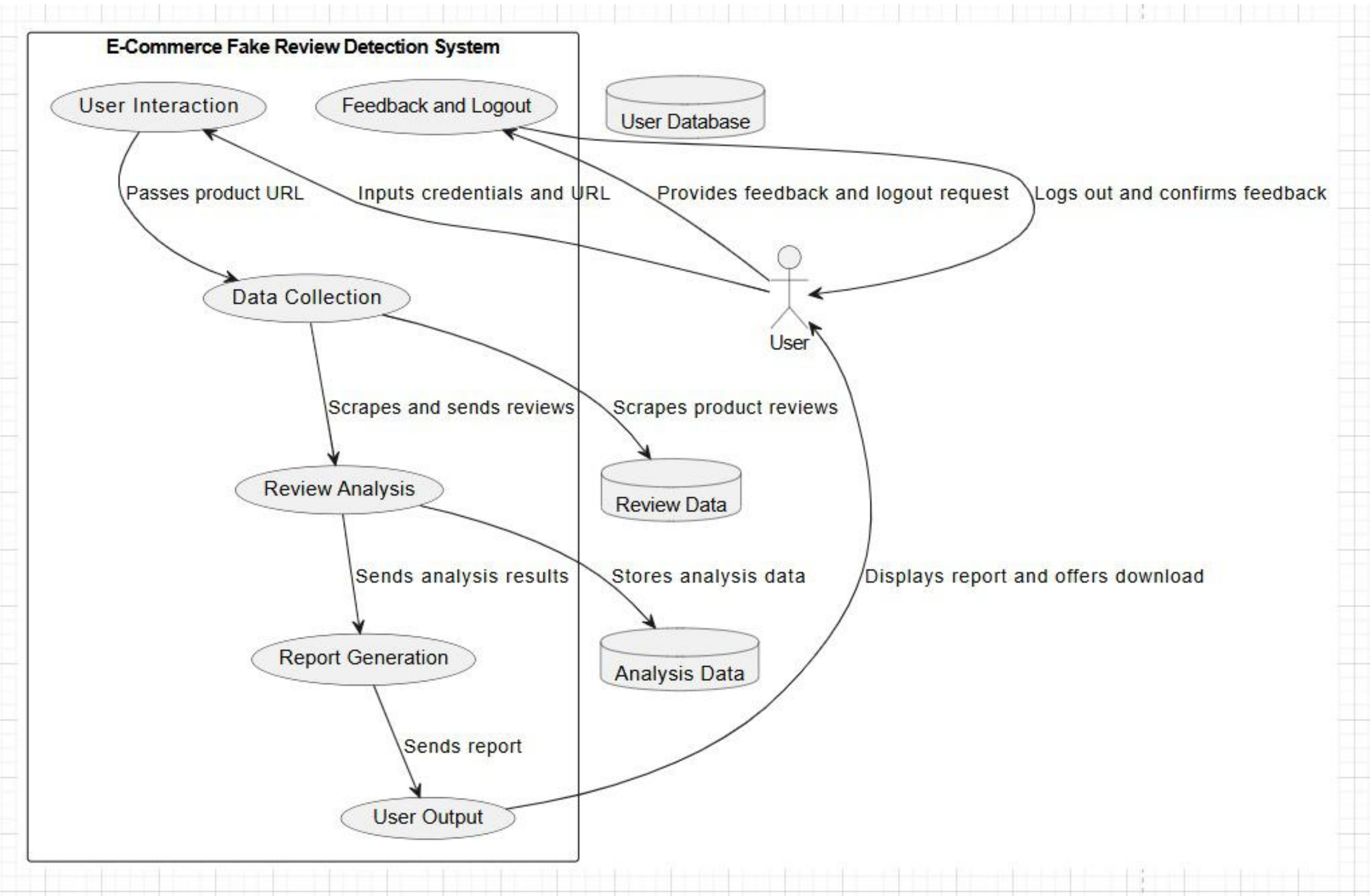


User accesses the application
Register
Login or Register?
Login
Collect user details (name, email, password)
Validate credentials
Create new profile
Yes
Successful login/registration?
No
Proceed to input product URL
Display error message
User provides the product URL
Validate URL format and check accessibility
Yes
URL valid?
No
Proceed to scraping reviews
Display error message
Prompt for a new URL
Scraper extracts reviews, ratings, and other data
Temporarily store data for analysis
Send scraped reviews to Analyzer module
Classify reviews as "Fake" or "Real" using ML model
Calculate percentage of fake reviews
Temporarily store analysis results
Generate detailed report

Level 0: Context Diagram


User
Input credentials for registration/login
Input product URL for analysis
Output analysis report
E-Commerce Fake Review Detection System
System

Level 1: High-Level DFD



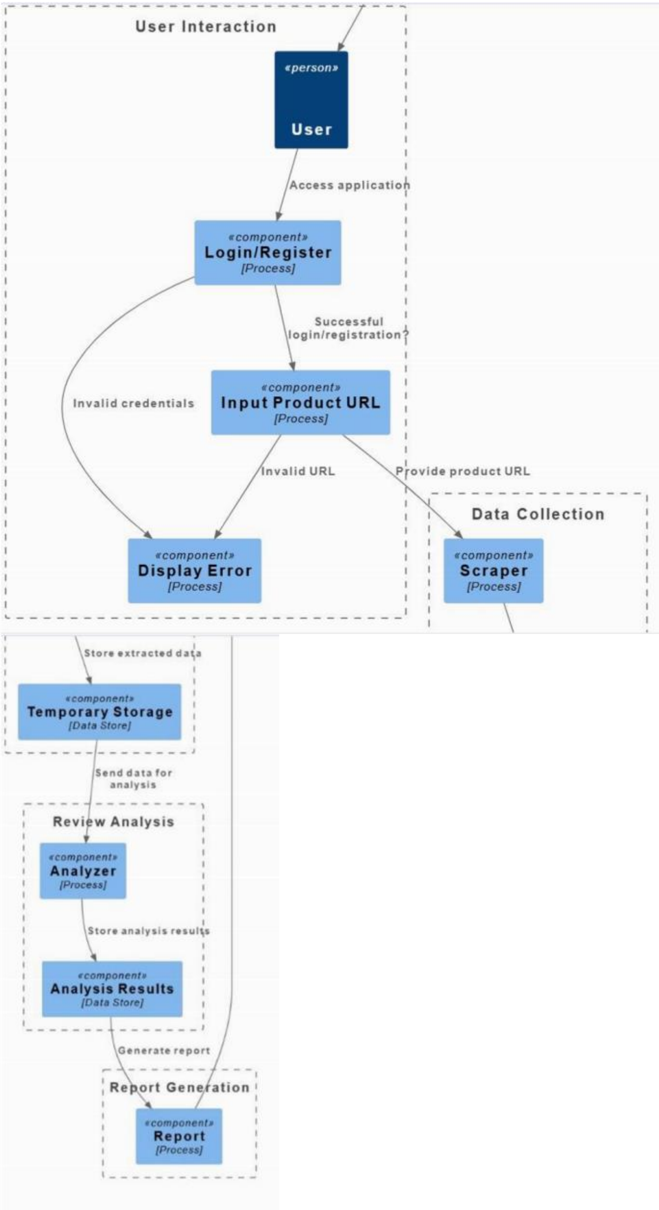
Level 2: Detailed DFD

Purpose: Provides a detailed breakdown of each high-level process.

Processes:

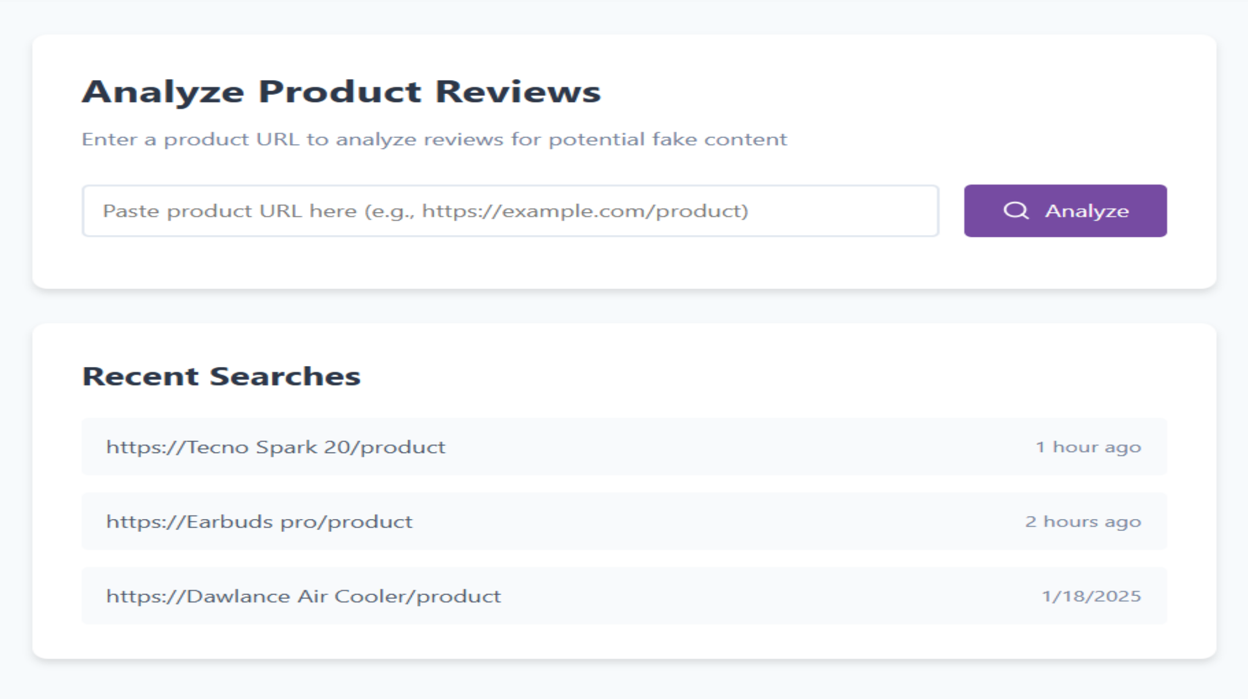
1. **Process 1: User Interaction**
    - **Input:** User credentials and product URL.
    - **Output:** Validation results for login/registration and URL validation.
2. **Process 2: Data Collection**
    - **Input:** Product URL.
    - **Output:** Scraped reviews and product details.
    - **Data Store:** Temporarily stores the review data for analysis.
3. **Process 3: Review Analysis**
    - **Input:** Reviews from the Data Collection process.
    - **Output:** Classification of reviews (fake or genuine) and the percentage of fake reviews.
4. **Process 4: Report Generation**
    - **Input:** Analysis results.
    - **Output:** A detailed report linked to the user's session, ready for display or download.
    - **Data Store:** Temporarily stores analysis results for reporting.
5. **Process 5: User Output**
    - **Input:** Analysis results.
    - **Output:** Displays the report and allows the user to download it.
6. **Process 6: Feedback and Logout**
    - **Input:** User feedback.
    - **Output:** Feedback confirmation and user logout.

5.9. DFD Diagram



5.10. GUI Design

Mock Screen 1: Home (User URL Input)



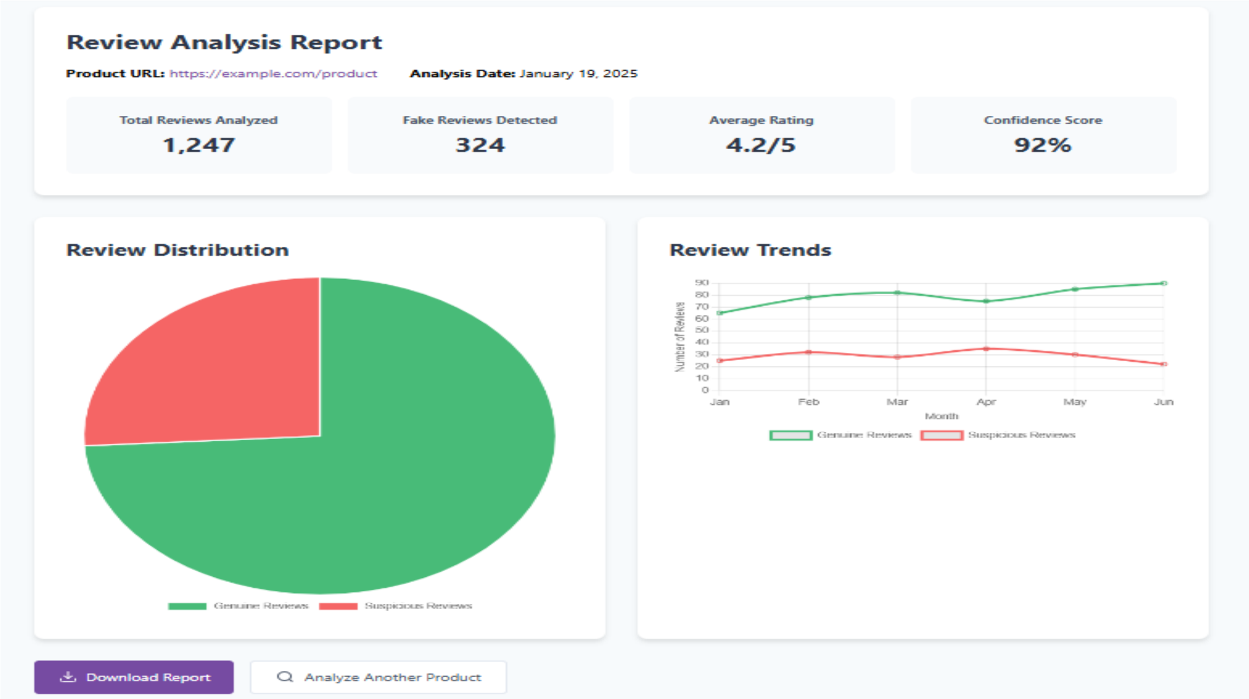
Analyze Product Reviews

Enter a product URL to analyze reviews for potential fake content

Recent Searches

https://Tecno Spark 20/product	1 hour ago
https://Earbuds pro/product	2 hours ago
https://Dawlance Air Cooler/product	1/18/2025

Mock Screen 2: Analysis Result

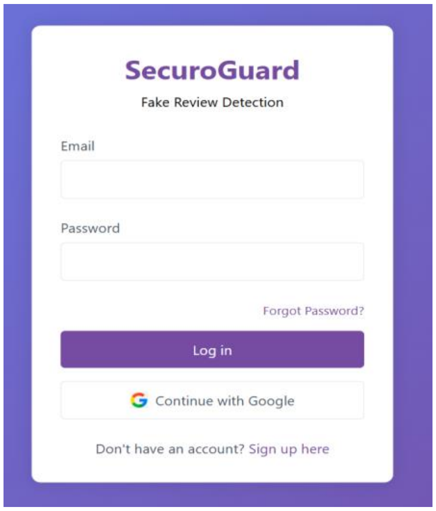


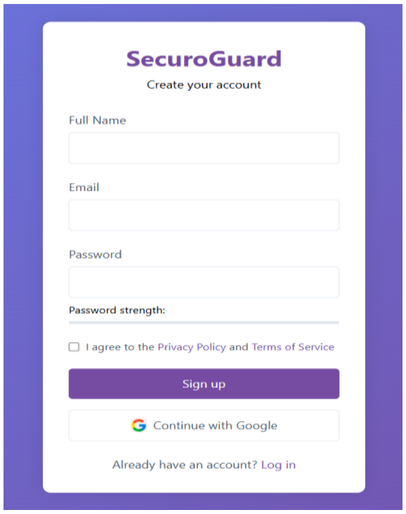
Review Analysis Report

Product URL: https://example.com/product **Analysis Date:** January 19, 2025

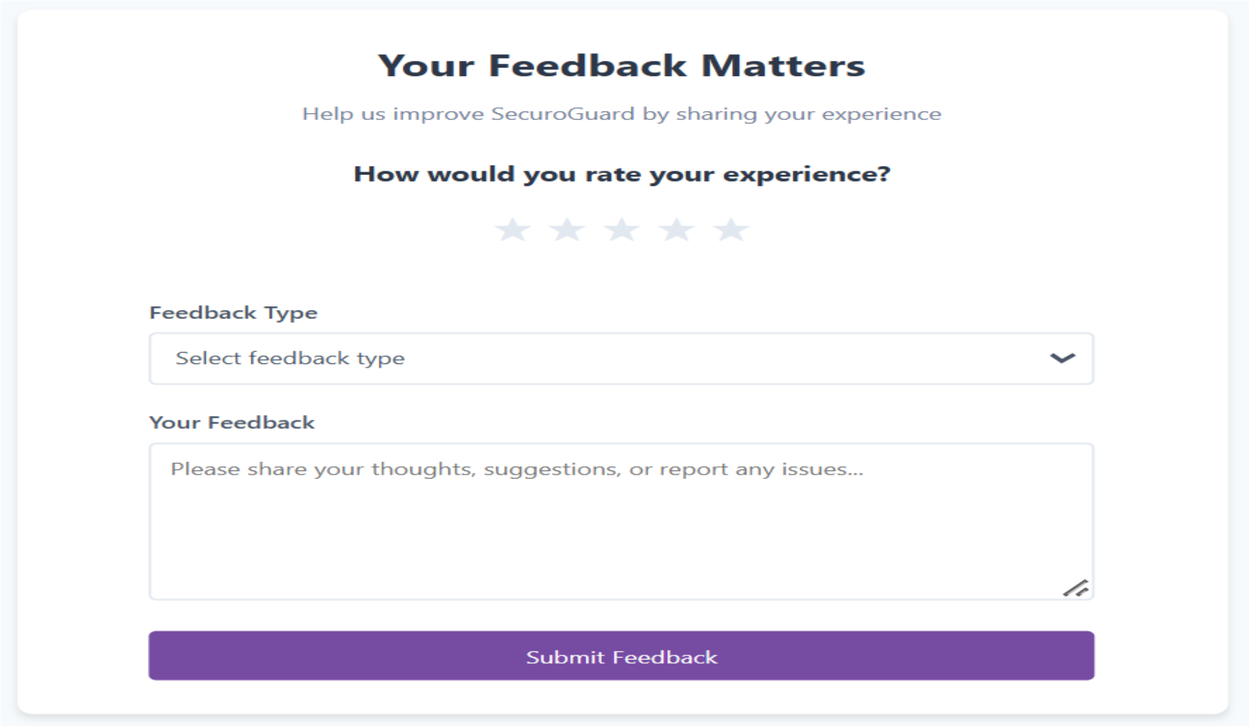
Total Reviews Analyzed	Fake Reviews Detected	Average Rating	Confidence Score
1,247	324	4.2/5	92%

Mock Screen 3: Signup/Login Page





Mock Screen 4: Feedback Page



Your Feedback Matters

Help us improve SecuroGuard by sharing your experience

How would you rate your experience?

★ ★ ★ ★ ★

Feedback Type

Select feedback type

Your Feedback

Please share your thoughts, suggestions, or report any issues...

6. References

[1] Fake review detection in e-Commerce platforms using aspect-based sentiment analysis. Journal of Business Research. November 2023, Volume 167, Pages 114143. Petr Hajek, Lubica Hikkerova, Jean-Michel Sahut. Accessed June 30, 2024. Available at: https://doi.org/10.1016/j.jbusres.2023.114143.

[2] Fake review detection system for online E-commerce platforms: A supervised general mixed probability approach. Decision Support Systems. December 2023, Volume 175, Page 114045. Jiwei Luo, Jian Luo, Guofang Nan, Dahui Li. Accessed June 30, 2024. Available at: https://doi.org/10.1016/j.dss.2023.114045.

[3] DRI-RCNN: An approach to deceptive review identification using recurrent convolutional neural network. Information Processing & Management. March 2018, Volume 54, Issue 2, Pages 255-268. Wen Zhang, Yuhang Du, Taketoshi Yoshida, Qing Wang. Accessed June 30, 2024. Available at: https://doi.org/10.1016/j.ipm.2018.03.007.

[4] Kumar, A., Gopal, R. D., Shankar, R., Tan, K. H. (2022). Fraudulent review detection model focusing on emotional expressions and explicit aspects: investigating the potential of feature engineering. Decision Support Systems, Volume 155, April 2022, 113728. Accessed June 30, 2024. Available at: https://doi.org/10.1016/j.dss.2021.113728.

[5] Opinion spam detection by incorporating multimodal embedded representation into a probabilistic review graph. November 13, 2019. Neurocomputing. Last accessed June 30, 2024. Authors: Liu Yuanchao, Pang Bo, Wang Xiaolong.

https://doi.org/10.1016/j.neucom.2019.08.013

[6] A deep learning approach for detecting fake reviewers: Exploiting reviewing behavior and textual information. March 2023. Decision Support Systems. Last accessed June 30, 2024. Dong Zhang, Wenwen Li, Baozhuang Niu, Chong Wu. https://doi.org/10.1016/j.dss.2022.113911

[7] Fake online reviews: Literature review, synthesis, and directions for future research.

May 2020. Decision Support Systems. Last accessed June 30, 2024. Yuanyuan Wu, Eric W.T.

Ngai, Pengkun Wu, Chong Wu. https://doi.org/10.1016/j.dss.2020.113280

[8] Fraud detection in online consumer reviews. 2010. Decision Support Systems. Last accessed July 1, 2024. Nan Hu, Ling Liu, Vallabh Sambamurthy.

https://doi.org/10.1016/j.dss.2010.08.012

[9] Assisting consumers in detecting fake reviews: The role of identity information disclosure and consensus. September 2016. Journal of Retailing and Consumer Services. Last accessed July 1, 2024. Andreas Munzel. https://doi.org/10.1016/j.jretconser.2016.06.002

[10] Detection of review spam: A survey. 1 May 2015. Elsevier. 1 July 2024. Atefeh Heydari, Mohammad Ali Tavakoli, Naomie Salim, Zahra Heydari. https://doi.org/10.1016/j.eswa.2014.12.029.