

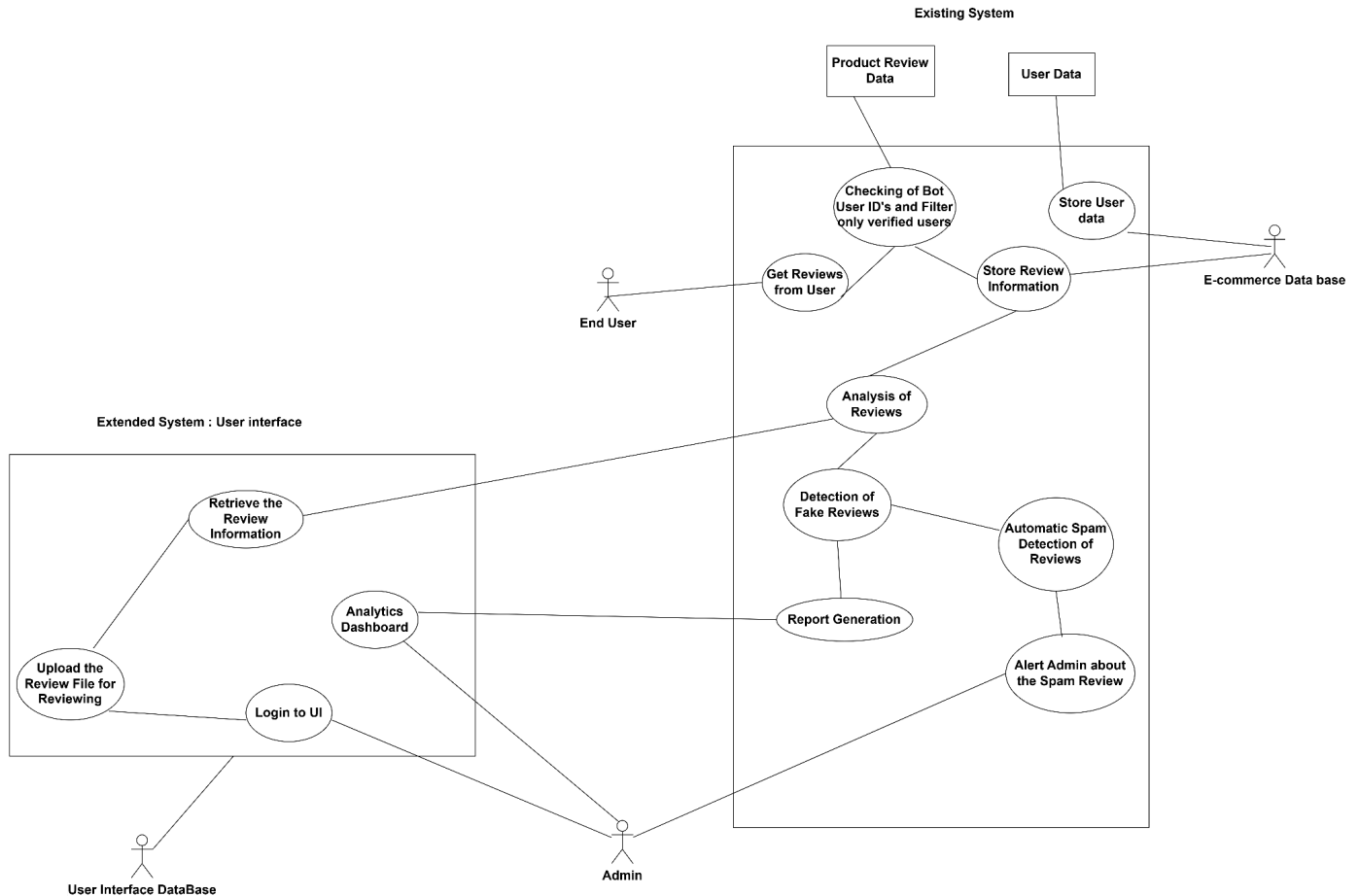
PHASE 3 :EXTENDING THE ARCHITECTURE OF AN EXISTING ML-BASED SYSTEM

Project Link : <https://github.com/anubhavs11/Fake-Product-Review-Monitoring>

Section 1: Identification of architecture drivers

Task 1.1 :UML Use Case Diagram

Our extended system is an User Interface system where the admin interacts with the system for Review Monitoring.



Description about the new use cases.

USE CASE	DESCRIPTION
Login to UI	Admin Logins to the UI page for analyzing his reviews
Upload Review File for Reviewing	This is an input file used by the admin for analyzing the spam reviews
Retrieve Review Information	The UI captures only the information required for analysis
Checking of Bot User IDs	Filters the users from bots and bot reviews will be dropped.
Spam detection	Automatic detection of Fake reviews when an end-user provide the new response

Alert to Admin	When a new response is identified as a Spam review, Sending notification to Admin about the Fake review detected.
User Interface Database	A Database for storing the information related to UI

Task 1.2 : Quality Attributes for the new use cases

Scenario 1: The admin login to the UI web page and upload the file for monitoring. Once uploaded, He gets the analytics of the Reviews submitted.

USE CASE	QUALITY ATTRIBUTES
Login to UI	Security : Authenticated users will only be able to access the webpage Usability : Simple and interactive login interface, Offer error messages for failed login attempts without exposing sensitive information. Scalability : Multiple users can parallelly use the webpage for analyzing their Reviews
Upload Review File for Reviewing	Performance : Enable fast and efficient file uploads for processing large datasets. Fault tolerance : If unsupported file type is uploaded, instead of failing the system completely, It generated an error message
Retrieve Review Information	Scalability : Handle large volumes of reviews without compromising user experience

Scenario 2: When new reviews are provided by the end-user, It automatically detect the Spam reviews and Send alerts to the Admin about the fake reviews

USE CASE	QUALITY ATTRIBUTE
Spam detection	Accuracy : It reduces the risk of unsolicited reviews Modifiability : Updates the conditions of Spam detection based on real time monitoring
Alert to Admin	Responsiveness : Deliver alerts to admins in real-time or within an acceptable delay Reliability : Ensure notifications are sent to admin within acceptable delay.

Scenario 3: When new feedback is provided by the end-user, Only those reviews which are from verified users, We store those review information in our database. The reviews generated from Bot user ID's are ignored.

USE CASE	QUALITY ATTRIBUTE
Checking of Bot User ID's	Accuracy : Using heuristics for identifying bot accounts, minimizing false accounts. Security : Protects the system from malicious users

Task 1.3 : Constraints

CONSTRAINTS	DESCRIPTION
Programming Language	The system uses python as primary language for Natural language processing methods and frameworks like NLTK, StopWords and Lemmatization is compatible with only python.
Network Constraints	Internet connectivity is crucial for the working of UI webpage

Data Constraints	The system is limited to .csv files. Any other formats are not recognized
Cost	The need of cloud hosting services (e.g., AWS) for running NLP models, storing Review data, requirement of multiple servers for concurrent processing is expensive.
Resource Constraints	Limited resources restrict the implementation of advanced features, such as high-end analytics dashboards

Task 1.4 : Concerns

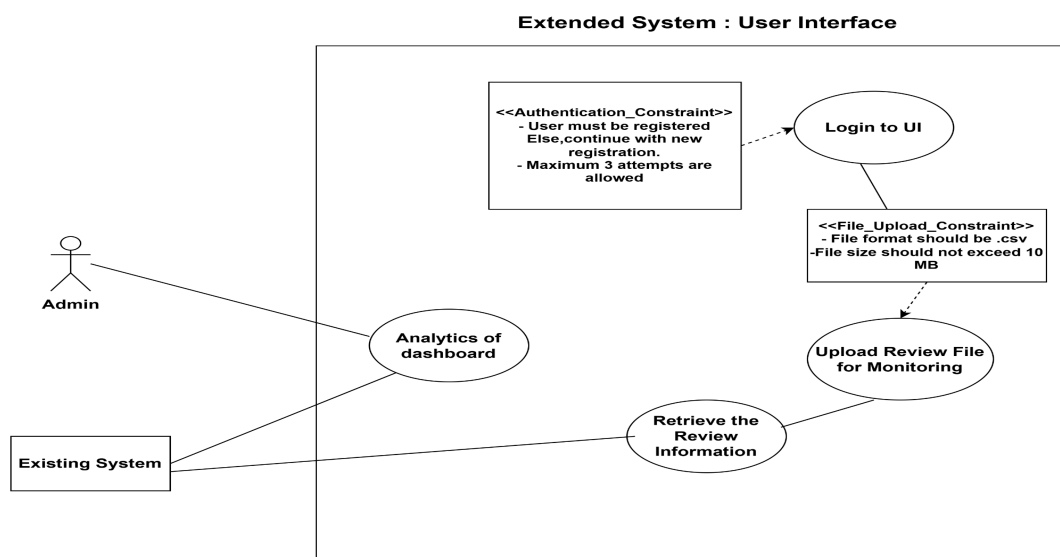
CONCERN	DESCRIPTION
Accuracy	The system should aim to minimize the false positives records
Alerting	The admin should receive the alerts when the spam is detected without any time delay.
Memory	As our system runs in local with limited memory, it can handle only small datasets

Section 2 : Update the software architecture

Task 2.1 : Solutions to the Architecture Drivers

Use case 1 : The admin login to the UI web page and upload the file for monitoring. Once uploaded, He gets the analytics of the Reviews submitted.

- **Security Solution** : We can have a pre-condition that if the user is a registered user, Then continue with the login or make a new registration. Allowing only 3 attempts for the password.
- **Fault Tolerance Solution** : If the User fails to login with the correct password in 3 attempts, Generate an Error message without failing the system.
- If the file upload is not as supported, Then generate a warning message to the user to support the file types.

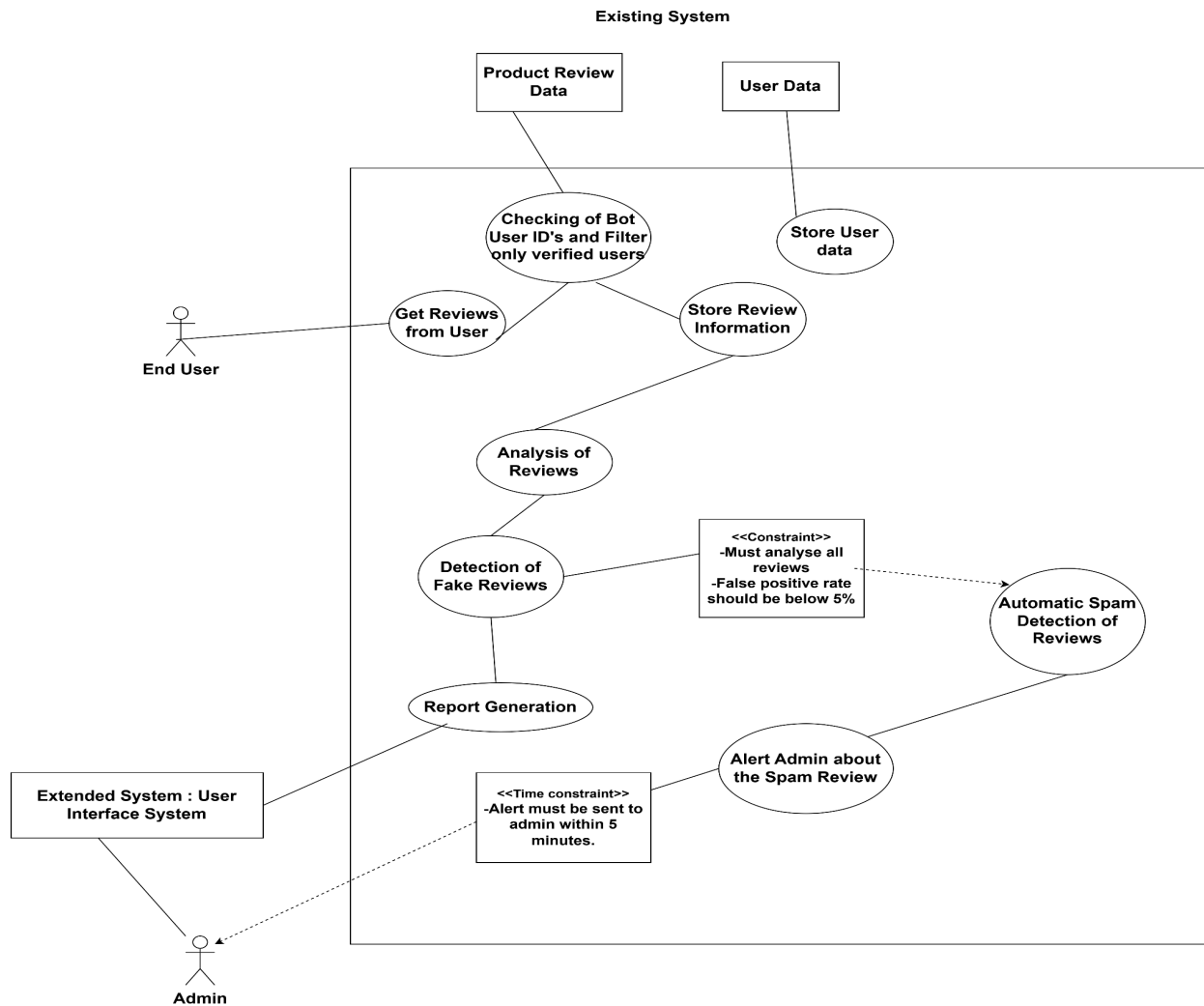


Use Case 2 : When new reviews are provided by the end-user, It automatically detect the Spam reviews and Send alerts to the Admin about the fake reviews

Constraint : Responsiveness, Reliability

Responsiveness Solution : When there is any spam present in the data, It should be detected and should send a message to the admin.

Reliability : The sending alert to the admin should be within the time constraint of 5 minutes.



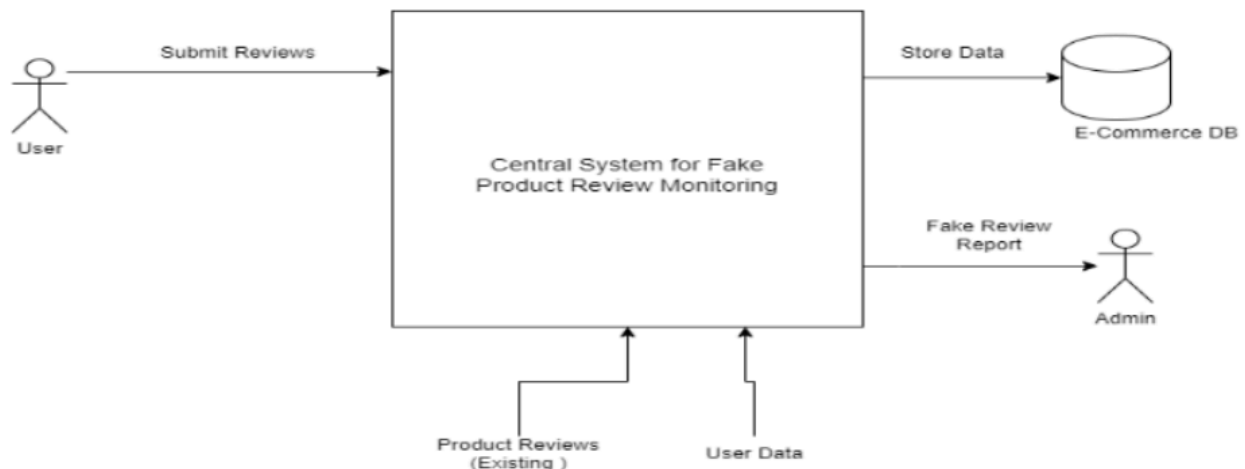
QUALITY ATTRIBUTES	SOLUTIONS
Security	Authentication and Authorisation : Only registered users can access the webpage
Performance	Using Load Balancing for processing concurrent files and large datasets. Using Efficient algorithms for less error rates
Scalability	Use Cloud services and implement data distribution across multiple servers

CONSTRAINTS	SOLUTIONS
Programming Language	We can use other languages for processing but are limited to the methods that we use. All the techniques that we use in NLP using python aren't supported by other languages. R language supports only Text2Vec, Julia. C++ supports libraries like FastText and Boost.local for handling these language processing methods.
Network Constraints	We can maintain a local system which performs the monitoring when the internet is lost. Once connection is retrieved, We can forward that information and store it on the server. This is known as Store and Forward method.
Data Constraints	Instead of failing the system completely, We can generate an error message when the wrong format of file is being uploaded by the admin.
Cost	Use object storage solutions like AWS S3, Google cloud to store large dataset. which is cost effective.
Resource Constraints	Instead of using Heavy Intensive Algorithm for detection of Fake reviews, Use light models like Logistic Regression so that within the available resources, We can perform advanced features.

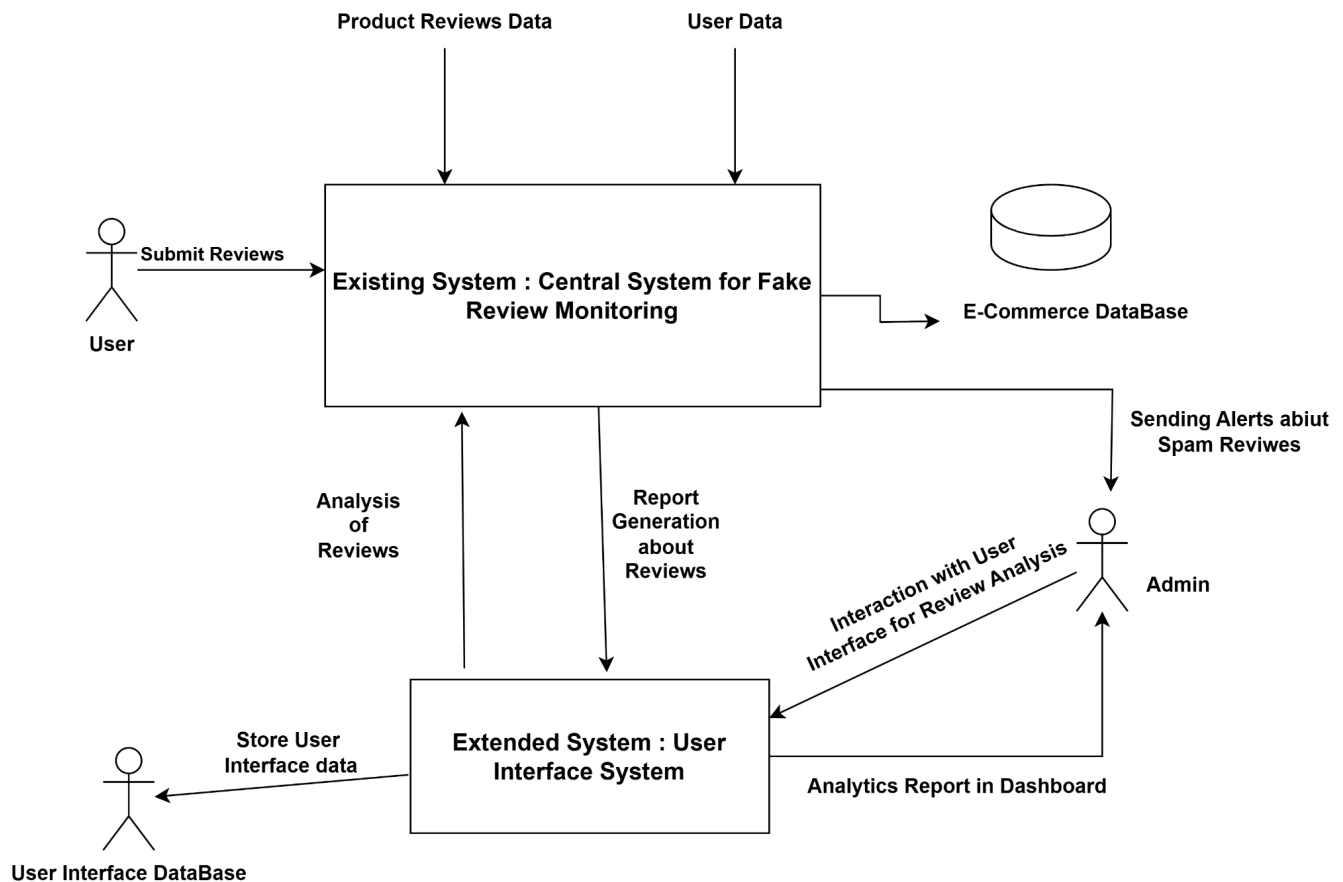
CONCERNS	SOLUTIONS
Accuracy	The model should use Optimization methods like Deep Learning for Spam detection.
Alerting	We can use a time constraint of 5 minutes, If exceeded, generate a warning message to the admin regarding the delivery failure of alert.
Memory	Instead of using the system locally, We can use cloud based memory for processing large datasets.

Task 2.2 : Context Diagram

Phase 1 : Context Diagram



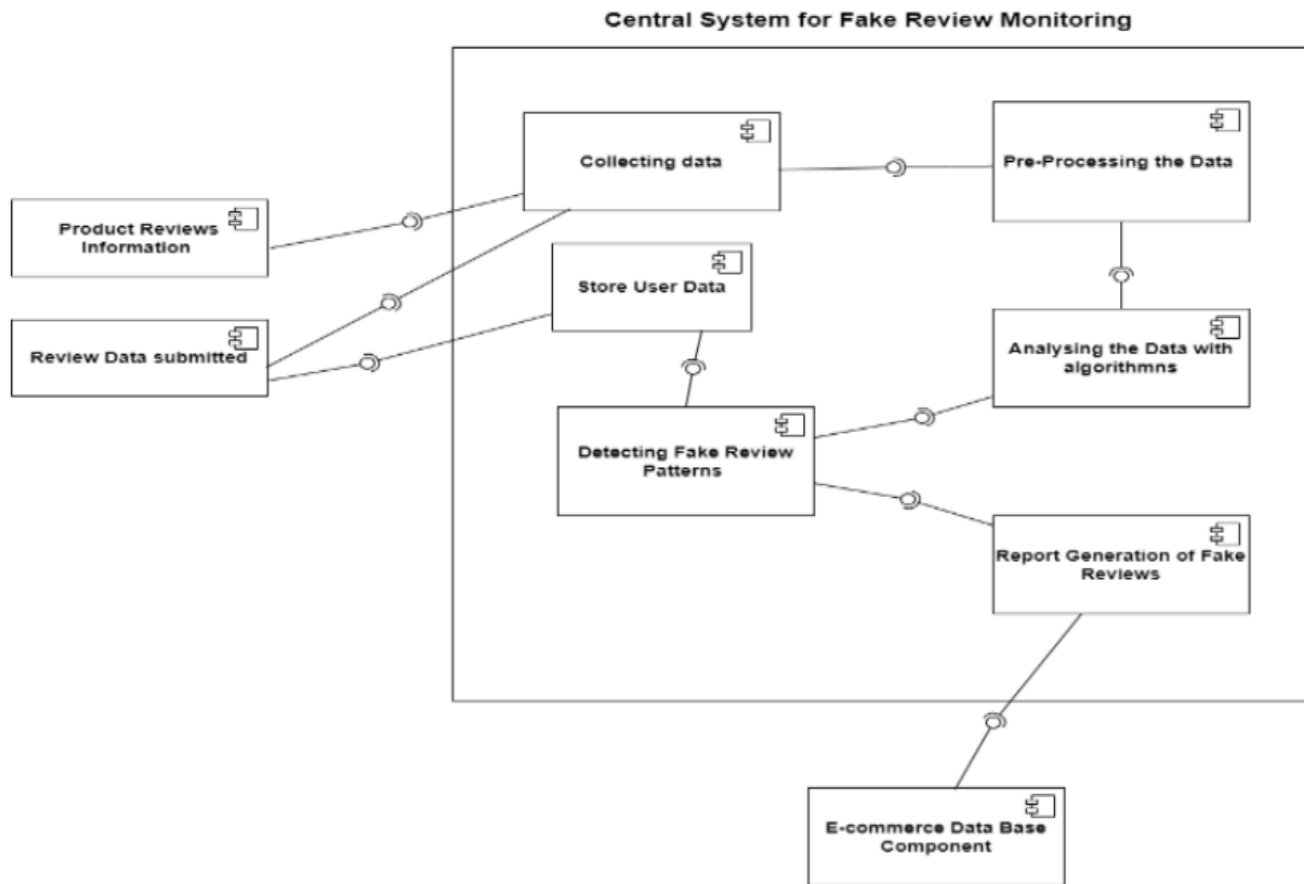
Improvements in Context Diagram



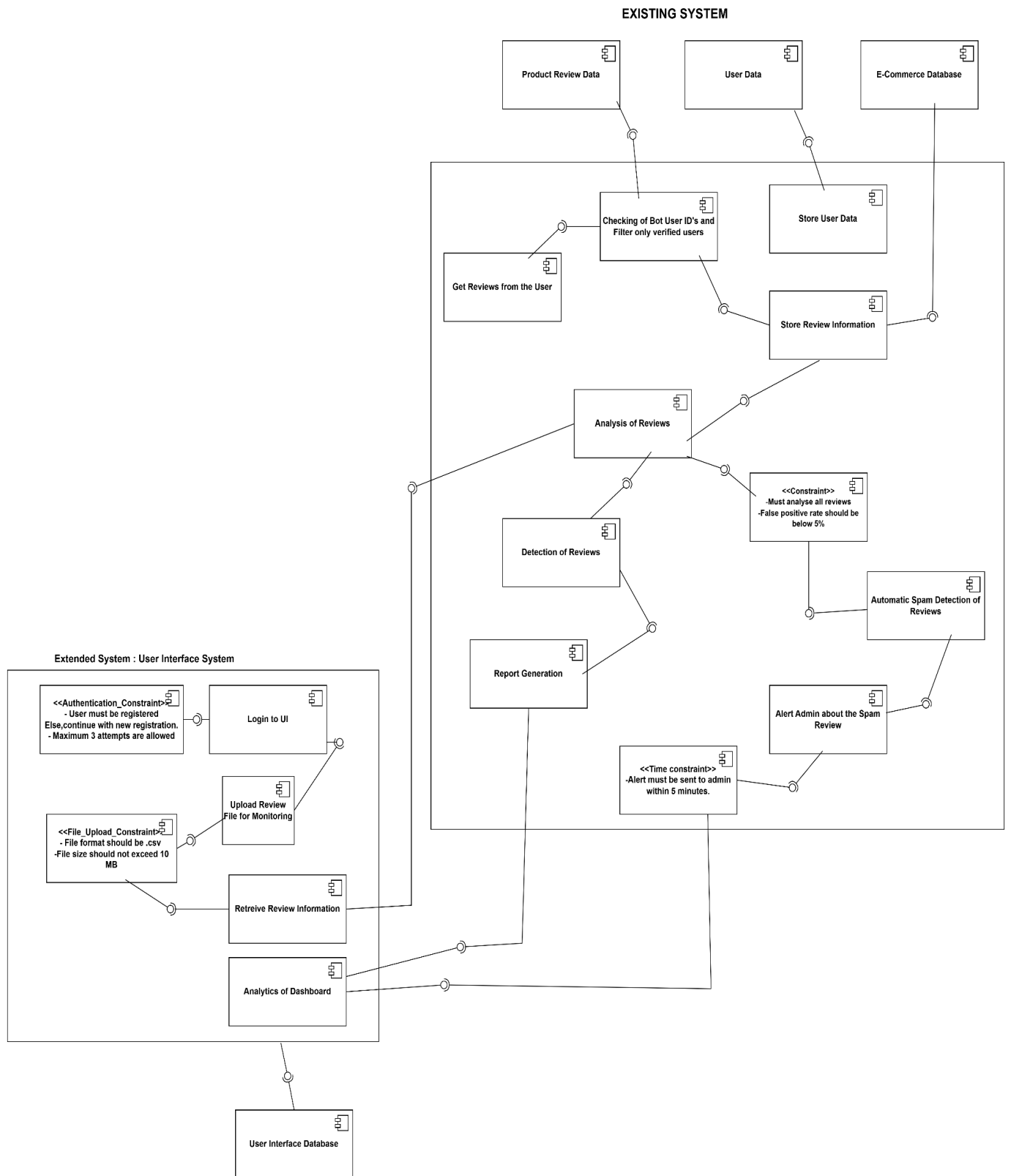
Description : In Phase 1, We have only one central system i.e Fake Review Monitoring System which accepts inputs submitted by the end-user in form of reviews .

- We use Analysis of the data and Detect the Fake reviews from the data provided. These results are stored in a report and generated to the admin who monitors the reviews.
- But in the Extended Proposed system, we have another system i.e User Interface system, Where The Admin interacts with the user interface system for analyzing his reviews.
- The UI system interacts with the existing system for analyzing reviews and generating the review report.
- Once report is generated, System uses that information to represent as analytics of Dashboard and this report is monitored by the admin
- Additional use case of sending alerts to the admin when Spam reviews are detected and are generated from the use of existing system.
- We have a User Interface Database which stores the information related to the new user interface system.

Task 2.3 : Component Diagram
Phase 1 :



Improvements in Component Diagram



Description:

- We have developed an advanced version of the system, introducing new capabilities that enhance its functionality and usability.
- One of the key additions is a user-friendly UI interface that provides an intuitive platform for admins to interact with the system.
- This interface allows for seamless operations such as uploading review data, analyzing flagged reviews, and monitoring system notifications, making it easier for users to manage and analyze feedback.
- In addition to these enhancements, we have carefully considered and integrated the constraints identified earlier, such as authentication,time and File type constraints.
- These constraints have been thoughtfully represented in the system's component diagram, ensuring that the architectural design aligns with both functional and non-functional requirements.
- This approach ensures that the system remains scalable, maintainable, and resource-efficient while addressing the core objectives of the project.

New Component	Description
Login to UI	Admin Logins to the UI page for analyzing his reviews
Upload Review File for Reviewing	This is an input file used by the admin for analyzing the spam reviews
Retrieve Review Information	The UI captures only the information required for analysis
Checking of Bot User ID's	Filters the users from bots and bot reviews will be dropped.
Spam detection	Automatic detection of Fake reviews when an end-user provide the new response
Alert to Admin	When a new response is identified as a Spam review,Sending notification to Admin about the Fake review detected.
User Interface Database	A Database for storing the information related to UI

Section 3 :REFLECTION

Task 3 : Lessons Learned

- Implementing machine learning (ML) in software architecture introduces significant changes and considerations that can enhance the system's capabilities but also add complexity
- The primary objective is to correctly classify spam messages while simultaneously allowing a user to freely interact with the application. ML models must be accurate in detecting spam without introducing latency
- Modularity allowed parts to be developed and scaled on their own to your advantage. It allows the ML components (like data preprocessing, model training, and inference) to be isolated and optimized separately.
- Using these Architecture constraints and concerns helps us to understand what all are the qualities required for the system to perform better.
- The usage of constraints before or after the components makes the system understand the importance of those conditions to make the system the better one.
- Using approaches such as transfer learning, cloud computing we can have resource optimization.
- Focusing on Scalability and Extensibility by Integrating the possibility of meeting the future requirements for data storage and adjustments for increasing data size.