Daffodil international University

Department of Software Engineering

Batch– 40th

Section- F2

Team Members

Shabiba Jahan Moni (0242310005341095)

Rakibul Hasan Zihad (0242310005341100)

Toky Yasir (0242310005341363)

Nuha Banu (0242310005341142)

Course Name: Software Engineering Design Capstone Project

Course Code: SE331

Project Title: AI-Based Job Post Fraud Detection System

Submitted to: Rahat Uddin Azad

Lecturer, Department of Software Engineering

Daffodil International University

AI-Based Job Post Fraud Detection System

Software Requirement Specification (SRS)

Introduction

1.1 Problem Statement

In recent years, many fraudulent job advertisements have been posted on online job portals and social media platforms. These fake job posts often include:

- Fake company information
- Unrealistic salary offers
- Requests for advance payment
- Suspicious contact details

Job seekers frequently become victims of scams. Most existing systems rely on manual reporting and verification, which is slow and inefficient. Therefore, an automated intelligent system is needed to detect fraudulent job posts**.**

1.2 Purpose

The main objectives of this software are:

- To detect fraudulent job posts using Artificial Intelligence and Machine Learning
- To classify job posts as Fraudulent or Legitimate in real-time
- To protect job seekers from online job scams
- To provide reports and analytics for system administrators

1.3 Scope

The system will:

- Analyze job post text using AI/ML techniques
- Predict fraud probability
- Flag suspicious job posts
- Provide reports and dashboards

The system will NOT:

- Guarantee any job authenticity legally
- Verify companies through legal authorities
- Manage job application processes

Design and Implementation Constraints

2.1 Programming Language / Technology Stack

- Python → Machine Learning Model (Scikit-learn, Pandas, NumPy)
- Laravel (PHP) → Backend Development
- HTML, CSS, JavaScript → Frontend Interface

2.2 Database & Servers

- MySQL → Store job posts, user data, and fraud predictions
- REST API → Communication between ML model and web system
- Local/Cloud Server → System deployment

User Classes and Characteristics

3.1 Job Seeker (General User)

- View job posts
- See fraud warnings
- Report suspicious job posts

3.2 Job Platform Administrators

- Monitor all job posts
- Review fraud detection results
- Manage users and system

3.3 System Administrator

- Analyze job post text
- system maintenance
- Manages datasets

Functional Requirements (FR)

ID	Requirement Name	Description	Users	Priority
FR-1	Job Post Submission	Admin can add new job posts	Admin	High
FR-2	Text Preprocessing	Clean and prepare job post text	System	High
FR-3	Fraud Detection	Classify job post as Fraud/Legit using ML	System	High
FR-4	Fraud Alert	Show warning for suspicious posts	User	High
FR-5	Report Generation	Generate fraud analysis reports	Admin	Medium
FR-6	User Feedback	Users can report suspicious posts	User	Medium

Non-Functional Requirements

5.1 Performance

- Fraud detection result must be generated within 5 seconds
- System should support multiple users simultaneously

5.2 Security

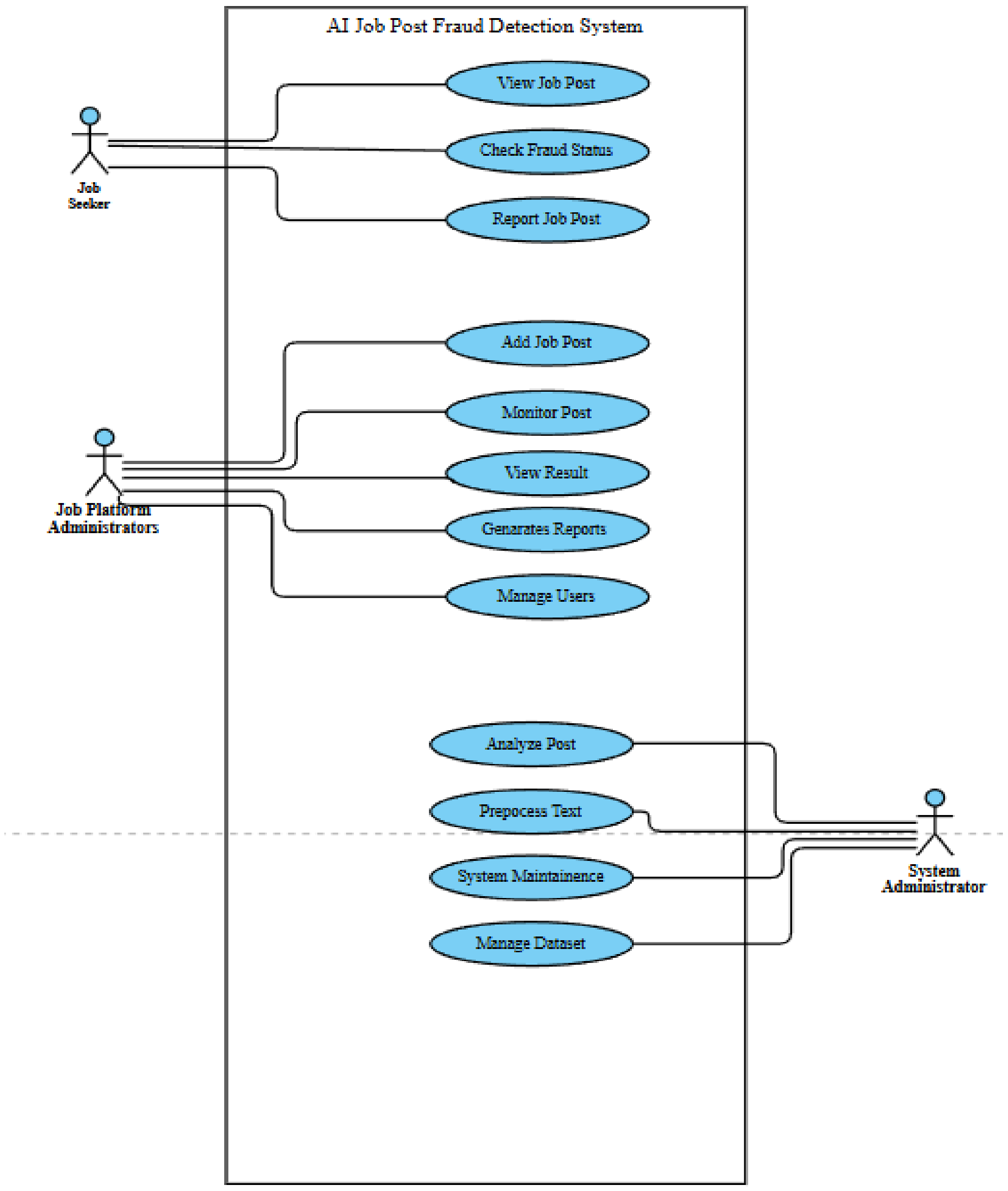
- User data must be securely stored
- Role-based admin access required
- Prevent SQL Injection and XSS attacks

5.3 Usability

- Simple and user-friendly interface
- Easy to use for non-technical users

System Design Diagrams

Use Case Diagram



Use Case Descriptions

UC-1: Add Job Post

Field	Details
Description	This use case describes the process by which an Admin adds a new job post into the system for fraud analysis.
Actors	Admin
Preconditions	Admin is logged into the system. Admin has permission to add job posts. System is running properly.
Postconditions	Job post is stored in the database. Job post is sent to fraud detection module.
Trigger	Admin selects “Add Job Post”.

UC-2: Manage Dataset

Field	Details
Description	This use case describes how the Admin manages the dataset used for training and testing the fraud detection model.
Actors	Admin
Preconditions	Admin logged in. Dataset available.
Postconditions	Dataset updated, added, or removed successfully.
Trigger	Admin selects “Manage Dataset”.

UC-3: View Job Posts

Field	Details
Description	This use case describes how a Job Seeker views available job posts and their fraud status.
Actors	Job Seeker
Preconditions	User has access to the system. Job posts exist in database.
Postconditions	Job posts displayed to user. Fraud warning shown if post is suspicious.
Trigger	User selects “View Job Posts”.

UC-4: Check Fraud Status

Field	Details
Description	This use case describes how a user checks whether a job post is fraudulent or legitimate.
Actors	Job Seeker
Preconditions	Job post already analyzed by system.
Postconditions	Fraud result displayed. Fraud alert shown if necessary.
Trigger	User opens job post details.

UC-5: Report Job Post

Field	Details
Description	This use case describes how a user reports a job post to the Admin.
Actors	Job Seeker
Preconditions	User logged in (optional). Job post exists.
Postconditions	Report stored in database. Admin notified.
Trigger	User clicks “Report” button.

UC-6: Generate Reports

Field	Details
Description	This use case describes how Admin generates analysis reports from system data.
Actors	Admin
Preconditions	Fraud detection results available. Admin logged in.
Postconditions	Fraud report generated. Report displayed or downloaded.
Trigger	Admin selects “Generate Report”.

UC-7: Manage Users

Field	Details
Description	This use case describes how Admin manages system users (add, update, delete).
Actors	Admin
Preconditions	Admin logged in. Admin has user management permission.
Postconditions	User information updated in database.
Trigger	Admin selects “Manage Users”.

UC-8: Text Preprocessing

Field	Details
Description	This use case describes how the system cleans and prepares job post text for machine learning analysis.
Actors	System
Preconditions	Job post text available.
Postconditions	Cleaned and tokenized text ready for feature extraction.
Trigger	Fraud detection process starts.

UC-9: Monitor Job Posts

Field	**Details**
Description	This use case describes how the Admin monitors all submitted job posts and checks their fraud status and activity.
Actors	Admin
Preconditions	Admin is logged into the system. Job posts exist in the database.
Postconditions	Admin can view job post list with fraud status and take necessary actions.
Trigger	Admin selects “Monitor Job Posts”.

UC-10: View Fraud Detection Result

Field	**Details**
Description	This use case describes how the Admin views the fraud detection results generated by the system.
Actors	Admin
Preconditions	Fraud detection process completed. Results stored in database. Admin logged in.
Postconditions	Fraud result displayed including fraud probability and classification.
Trigger	Admin selects “View Result”.

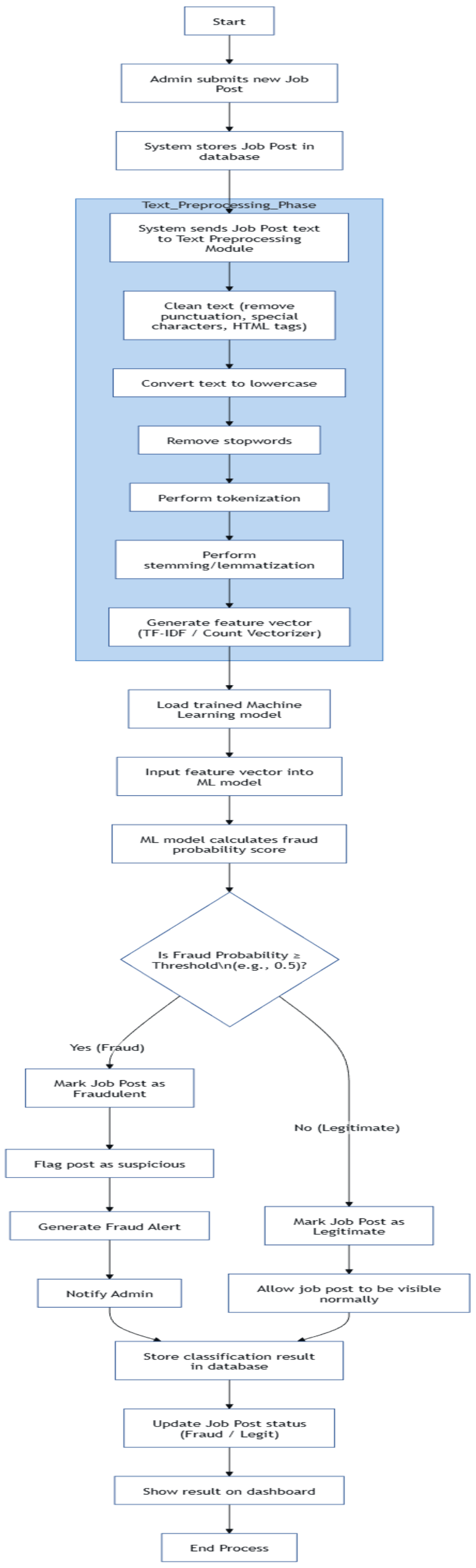
UC-11: Analyze Job Post

Field	Details
Description	This use case describes how the system analyzes job post content using machine learning techniques to determine fraud probability.
Actors	AI Detection System
Preconditions	Job post text available. Feature extraction completed. ML model loaded.
Postconditions	Fraud probability calculated and sent for classification.
Trigger	Fraud detection process initiated.

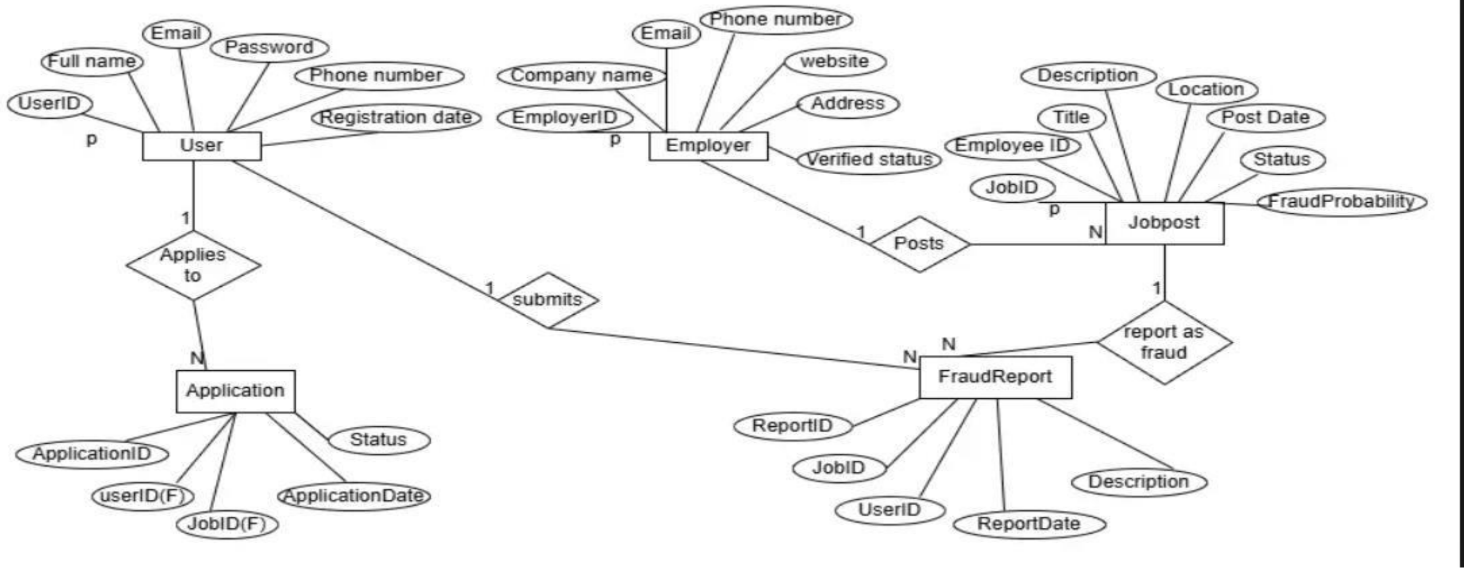
UC-12: System Maintenance

Field	Details
Description	This use case describes how the Admin maintains the system including updating model, fixing errors, and ensuring system performance.
Actors	Admin
Preconditions	Admin logged in with maintenance privileges. System operational.
Postconditions	System updated, bugs fixed, and performance optimized.
Trigger	Admin selects “System Maintenance”.

Activity Diagram


Start
Admin submits new Job Post
System stores Job Post in database
Text_Preprocessing_Phase
System sends Job Post text to Text Preprocessing Module
Clean text (remove punctuation, special characters, HTML tags)
Convert text to lowercase
Remove stopwords
Perform tokenization
Perform stemming/lemmatization
Generate feature vector (TF-IDF / Count Vectorizer)
Load trained Machine Learning model
Input feature vector into ML model
ML model calculates fraud probability score
Is Fraud Probability ≥ Threshold\n(e.g., 0.5)?
Yes (Fraud)
Mark Job Post as Fraudulent
Flag post as suspicious
Generate Fraud Alert
Notify Admin
No (Legitimate)
Mark Job Post as Legitimate
Allow job post to be visible normally
Store classification result in database
Update Job Post status (Fraud / Legit)
Show result on dashboard
End Process

ER Diagram



Team Member Responsibility Matrix

Member Name	ID	Primary Responsibilities (Database, UI, API)
Rakibul Hasan Zihad	0242310005341100	Machine Learning model development, Fraud detection logic, Text preprocessing, Model training & evaluation.
Shabiba Jahan Moni	0242310005341095	Backend API development using Laravel, Authentication system, Business logic implementation, System integration.
Nuha Banu	0242310005341142	Frontend UI/UX design using HTML, CSS, JavaScript, Dashboard development, User interaction & validation.
Toky Yasir	0242310005341363	Database design (MySQL), ER Diagram, Dataset management, System testing, Bug fixing & documentation support.