Academic Monitoring & Specialization Selection System

A
Project Report
submitted in partial fulfillment of the
requirements for the award of the degree of

BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE & ENGINEERING

by

Name	Roll No.
Dhuruv Kumar	**500107769**
Khushi Chauhan	**500102244**
Shruti Srivastava	**500105401**

under the guidance of
Dr. Deepak Sharma
Asst. Professor (SG)
System Cluster, SoCS

School of Computer Science
University of Petroleum & Energy Studies
Bidholi, Via Prem Nagar, Dehradun, Uttarakhand

November – 2024

CANDIDATE'S DECLARATION

We hereby certify that the project work entitled **"Academic Monitoring & Specialization Selection System"** in partial fulfilment of the requirements for the award of the Degree of BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE AND ENGINEERING with specialization in BIG DATA and submitted to the Department of Systemics, School of Computer Science, University of Petroleum & Energy Studies, Dehradun, is an authentic record of our work carried out during a period from **AUGUST**, **2024** to **NOVEMBER**, **2024** under the supervision of **Mr. Deepak Sharma (Asst. Professor (SG) System Cluster, SoCS)**.

The matter presented in this project has not been submitted by us for the award of any other degree of this or any other University.

Dhuruv Kumar
500107769
Khushi Chauhan
500102244
Shruti Srivastava
500105401

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Date: ______________2024

Dr. Deepak Sharma
Asst. Professor (SG)
System Cluster, SoCS
Project Guide

ACKNOWLEDGEMENT

We wish to express our deep gratitude to our guide D**r. Deepak Sharma**, for all the advice, encouragement and constant support he has given us throughout our project work. This work would not have been possible without his support and valuable suggestions.

We sincerely thank our respected **Dr. Virender Kadyan, Cluster Head of Data Science Cluster** for his great support in doing our project in **Academic Monitoring and Specialization Selection System.**

We are also grateful to Dean SoCS UPES for giving us the necessary facilities to carry out our project work successfully. We also thank our Course Coordinator, Dr. Prabhat Ranjan Singh and our Activity Coordinator Dr. Sachin Chaudhary for providing timely support and information during the completion of this project.

We would like to thank all our **friends** for their help and constructive criticism during our project work. Finally, we have no words to express our sincere gratitude to our **parents** who have shown us this world and for every support they have given us.

Name	**Dhuruv Kumar**	**Khushi Chauhan**	**Shruti Srivastava**
Roll No.	**500107769**	**500102244**	**500105401**

ABSTRACT

The **Academic Monitoring & Specialization Selection System** is a comprehensive data-driven platform aimed at improving academic support and career guidance for students in higher education institutions. In today's competitive educational landscape, students face increasing pressure to make informed decisions about their academic specializations, while educators seek efficient methods to support diverse student needs. This system addresses these challenges by employing machine learning techniques to analyze student data, thereby delivering both individualized recommendations and group-based support.

This project encompasses two main modules: a **classification module** and a **clustering module**. The classification module utilizes the ID3 decision tree algorithm to evaluate students' academic histories and personal preferences, enabling it to recommend specialization paths that align with each student's strengths and interests. By offering tailored guidance, this module empowers students to make informed choices about their academic focus, helping them build a foundation for future career success.

The clustering module organizes students based on their performance metrics, creating distinct groups that allow educators to deliver targeted interventions and academic support. This approach enables institutions to efficiently allocate resources and focus attention on specific student clusters, facilitating better academic outcomes and supporting each student's unique learning journey.

For data management, the system relies on MongoDB for robust, flexible storage, while CSV files are used for clustering and preprocessing to ensure streamlined data handling. The user experience is designed to be intuitive, with a menu-driven interface that simplifies navigation for both students and administrators, enhancing usability across different user roles.

The accompanying Software Requirements Specification (SRS) comprehensively documents the system's functional requirements, technical specifications, and operational guidelines. Key aspects include performance monitoring, scalability, user documentation, training needs, and backup protocols, ensuring the system's resilience and adaptability in a dynamic educational environment.

By combining machine learning with user-centered design, the Academic Monitoring & Specialization Selection System provides an essential tool for advancing educational support, fostering informed decision-making, and ultimately enhancing student success.

Table of Contents

List of Figures

1. INTRODUCTION

1.1 PURPOSE OF THE PROJECT

The purpose of the "Academic Monitoring & Specialization Selection System" project is to develop, design and implement a dual-purpose academic support system that enhances both student guidance and educational management.

The purpose of this project is twofold:

- **Specialization Selection for Students**: Many students struggle with choosing a specialization that aligns with their academic strengths and career aspirations. This often leads to dissatisfaction and reduced performance in their chosen fields.
- **Performance-based Grouping for Educators**: Educators face difficulties in providing targeted support due to the varied academic performance levels of students. Managing these diverse needs in a large class setting can be challenging, leading to inefficient resource allocation and missed opportunities for student improvement.

This project aims to solve these problems by implementing advanced algorithms for specialization recommendation and performance-based student grouping. The motivation for this project stems from the growing need for data-driven decision-making in education and the potential to significantly improve student outcomes and institutional efficiency. By leveraging data-driven techniques such as classification and clustering algorithms, this project seeks to improve educational outcomes and streamline academic management.

1.2 MOTIVATION

As third-year students, we found ourselves at a crossroads when it came to selecting a specialization. The decision was both confusing and overwhelming, as we lacked clarity about which area of study truly aligned with our interests and strengths. This confusion, coupled with limited guidance, often led to a mismatch between our chosen specialization and our actual inclinations or career goals. This challenge inspired us to create a system that could guide students in making informed decisions based on their academic performance and interests. By identifying patterns in students' grades and preferences, we aim to recommend specializations that best suit their skills and aspirations.

Additionally, another key motivation for this project was to enhance resource allocation in educational settings. Students vary significantly in their learning pace and requirements. For instance, fast learners may find repetitive content redundant, while slow learners may feel overburdened with advanced material. By grouping students based on their academic performance, this system ensures tailored resource distribution. Fast learners can focus on advanced content, fostering deeper understanding, while slower learners can receive the foundational support they need. This approach not only optimizes learning outcomes but also promotes an inclusive and efficient educational environment. Through this project, we aim to address these challenges and simplify the academic journey for students.

1.3 OBJECTIVE

The objective of the "Academic Monitoring and Specialization Selection System" project is to create a dual-purpose platform that enhances educational guidance for students and optimizes academic management for educators. This system aims to address two primary challenges in higher education: assisting students in selecting specializations that align with their strengths and aspirations, and enabling educators to effectively group students based on performance metrics. Key Objectives:

Specialization Selection: By employing advanced algorithms like Decision Trees, the system analyzes students' academic records, preferences, and career goals to provide personalized specialization recommendations. This feature aims to improve student satisfaction and success rates by aligning their studies with their individual strengths.

Performance-based Grouping: Utilizing K-Medoids clustering algorithms, the system categorizes students based on performance data such as grades and attendance. This allows educators to identify varying performance levels within a class, facilitating targeted interventions and resource allocation.

1.4 TARGET BENEFICIARY

The target beneficiaries of the "Academic Monitoring and Specialization Selection System" project include:

Students: The primary beneficiaries are students who will benefit from personalized recommendations for specialization choices based on their academic records, preferences, and career aspirations. This tailored guidance aims to enhance their academic journey and improve overall satisfaction and success rates in their chosen fields.

Educators: Teachers and academic advisors will gain valuable insights into student performance through the system's performance-based grouping feature. This capability allows educators to efficiently identify varying levels of student achievement, enabling them to provide targeted support and resources where needed, thus enhancing the learning experience for all students.

Educational Institutions: Universities and colleges are also significant beneficiaries as they can utilize the system to improve academic planning and performance management processes. By leveraging data-driven decision-making, institutions can enhance educational quality, streamline operations, and ultimately improve student outcomes.

Academic Administrators: Administrators will gain insights into student performance and trends, which can inform policy decisions and resource allocation. This information will help them better address the diverse needs of the student body and ensure that institutional resources are used effectively to improve educational quality.

1.5 PROJECT SCOPE

The scope of the "Academic Monitoring and Specialization Selection System" project encompasses the development and implementation of a dual-purpose platform aimed at enhancing student guidance and educational management within higher education institutions.

Areas of Application

- **Specialization Guidance**: The system uses a **Decision Tree classification algorithm** to analyse students' academic records, personal preferences, and career goals. It then recommends the most suitable specialization paths, helping students align their aspirations with their academic strengths.
- **Performance-based Grouping**: A **K-Medoids clustering algorithm** is implemented to group students based on performance data such as test scores, attendance, and feedback. This feature enables educators to identify different performance levels and provide customized support accordingly.

Functional Components

- **Specialization Guidance:** Utilizing a Decision Tree classification algorithm, the system analyzes students' academic records, personal preferences, and career goals to recommend suitable specialization paths. This feature aims to align students' aspirations with their academic strengths, thereby improving their chances of success in their chosen fields.
- **Performance-based Grouping:** The project implements a K-Medoids clustering algorithm to categorize students based on performance metrics such as grades, attendance, and feedback. This grouping enables educators to identify varying performance levels, facilitating targeted interventions and resource allocation.

Key Deliverables

- **Data Storage:** Implementation of a MongoDB database to store academic history, preferences, performance data, and grouping information.
- **Algorithm Development:** Design and implementation of algorithms for specialization recommendations and performance-based grouping.
- **User Interface:** A user-friendly interface that allows both students and educators to interact with the system easily.
- **Testing and Refinement:** Continuous testing to refine algorithms and the user interface based on user feedback and performance metrics.

1.6 PERT CHART LEGEND

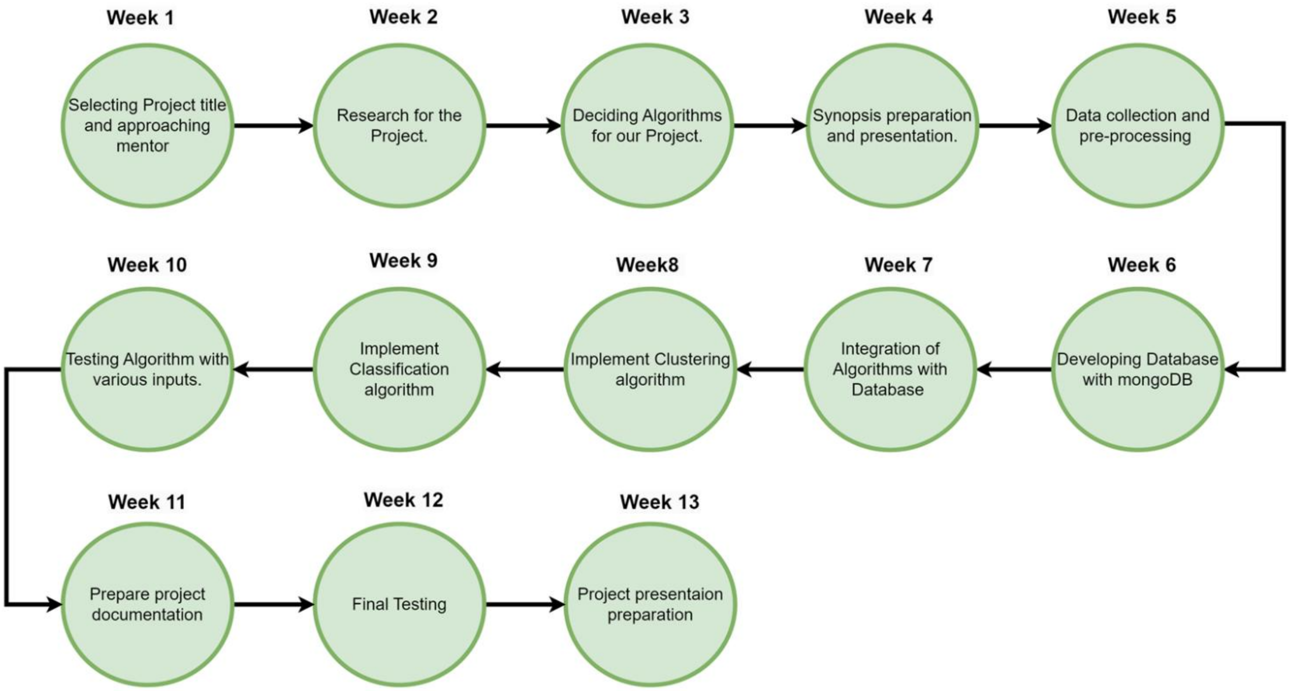

Figure 1-1 PERT CHART

2. PROJECT DESCRIPTION

2.1 ALGORITHM

2.1.1 Decision Tree Algorithm for Specialization Selection

Purpose: To recommend suitable specializations for students based on their academic records, preferences, and career aspirations.

This project employs an Iterative Dichotomiser (ID3) Decision Tree Algorithm to assist students in selecting the most suitable specialization based on their academic performance, preferences, and other relevant factors. The decision tree classifies students using numerical and categorical data, ensuring an accurate and well-informed recommendation.

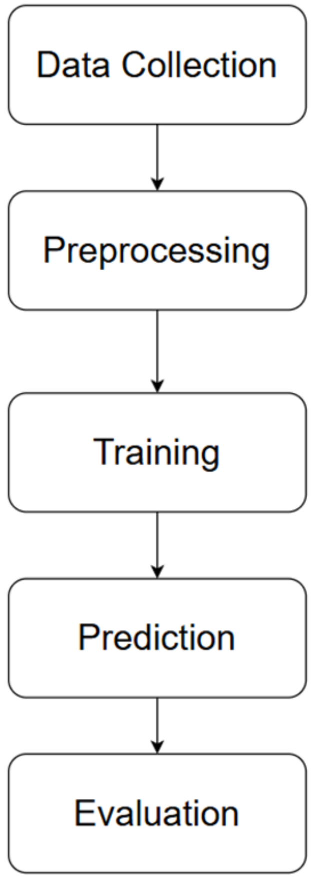

Figure 1-2 Steps in Decision Tree

Steps in the Algorithm:

1. **Data Collection:**

   The dataset is composed of various student attributes:

   Numerical Attributes: CGPA, grades in core subjects like CPL, DSA, Python, etc.
   Categorical Attributes: Preferences in specializations such as Machine Learning, Cybersecurity, Networking, etc.
   Target Attribute: Specialization choice, which serves as the class label.

2. **Preprocessing:**

   Data preprocessing ensures clean and structured data for better accuracy.

   Data Cleaning: Missing values are handled by either removing incomplete entries or imputing them using statistical methods (mean, median, or mode).
   Normalization: Numerical features (e.g., grades) are scaled to a common range (e.g., 0 to 1) to prevent attributes with larger ranges from dominating the model.

Categorical Encoding: Convert categorical preferences into numerical form using one-hot encoding.

3. **Entropy Calculation:**

Entropy measures the disorder or impurity of the dataset. We use the entropy formula to determine the uncertainty in the dataset.
Formula:

$$Entropy(S) = -\sum_{i=0}^{n} p_i \log_2(p_i)$$

Here, pi represents the probability of each class in the dataset S.

4. **Information Gain:**

For each attribute, calculate the information gain, which indicates the reduction in entropy. For each attribute, we calculate the information gained to identify which attribute best splits the data.
Formula: For each attribute (A):

$$\text{Information Gain(A)} = \text{Entropy(S)} - \sum_{v \in Value(A)} \frac{|S_v|}{|S|} \text{Entropy}(S_v)$$

Here is the subset of S where attribute A has value. The attribute with the highest information gain is selected for the splitting

5. **Decision Tree Construction:**

Choose the attribute that provides the highest information gain as the decision node for the current node in the tree. The algorithm recursively **Selects the best attribute** (with the highest information gain) as the decision node.
Subtrees are created for each value of the chosen attribute, continuing until all subsets are pure or no further attributes are left or all data points in a subset belong to the same class.

6. **Stopping Criteria:**

The recursion stops when:

- All instances in a node belong to the same class.
- No attributes remain to split further.
- A predefined maximum depth is reached to prevent overfitting.

7. **Pruning (Optional):**

   After constructing the tree, pruning is performed to remove branches that contribute little to predictive accuracy.
   Methods such as reduced error pruning or cost-complexity pruning are used.

Prediction Process:

Once the decision tree is trained, it classifies new students by traversing the tree based on their input attributes, ultimately predicting the most suitable specialization.

Evaluation:

The model's performance is evaluated using various metrics, such as accuracy, to ensure its effectiveness in recommending specializations.

2.1.2 K-Medoids Clustering Algorithm for Academic Monitoring

Purpose: K-Medoids is a robust clustering algorithm that minimizes the impact of outliers by selecting medoids (representative data points) as cluster centers.
The K-Medoids algorithm is employed to group students based on their performance metrics, such as grades, CGPA, and attendance. By clustering students, we can identify patterns and categorize them into distinct groups, aiding in better specialization guidance.

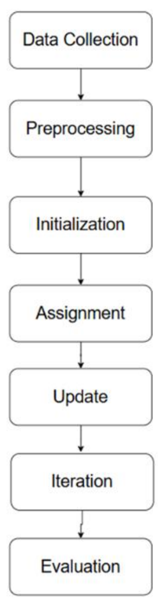

Figure 1-3 Steps in K-Medoids Clustering Algorithm

Steps of the Algorithm:

1. Data Collection:

The data required for clustering consists of:

Numerical data:

- Grades: The academic performance of students in various subjects.
- CGPA: The cumulative grade point average, representing overall academic performance.
- Attendance: The percentage of classes attended by students.

Categorical data:

- Specialization interests: This includes interests in areas like Machine Learning, Cybersecurity, Data Science, etc. Each student's interest can be represented as a vector or one-hot encoded attributes.

2. Data Preprocessing:

Data preprocessing ensures clean and structured data for better accuracy.

Handling missing values: If any student's performance data (such as grades or attendance) is missing, it may need to be imputed or removed.

Normalization/Standardization: Since K-Medoids uses distance metrics, the data must be normalized or standardized. This is crucial when dealing with attributes like grades, CGPA, and attendance, as they may have different scales (e.g., CGPA might range from 0 to 10, while grades could be percentage-based). A common technique is min-max scaling or z-score normalization to scale numerical data.

Encoding categorical variables: For specialization interests (a categorical attribute), we may use techniques like one-hot encoding or label encoding to convert them into numerical representations.

Outlier detection and removal: Outliers can distort the distance metrics, so it's important to identify and address them if necessary.

3. Input:

A dataset containing student performance metrics:

- **Numerical attributes**: Grades, CGPA, attendance percentage, etc.
- **Categorical attributes**: Interests in specific specializations such as Machine Learning, Cybersecurity, Data Science, etc.

The desired number of clusters k, which needs to be specified upfront.

4. Initialization:

The algorithm begins by randomly selecting k data points as the initial medoids. These medoids serve as the central representatives of each cluster.
The selection of these medoids is critical as they determine the starting point for the clustering process**.**

5. Assignment Step:

For each student x_i in the dataset Calculate the distance from to each medoid using the Euclidean distance:

$$d(x_i, m_j) = \sqrt{\sum_{p=1}^{n} (x_{ip} - m_{jp})^2}$$

Once the distances are calculated, each student is assigned to the cluster of the closest medoid, i.e., the student is associated with the medoid that minimizes the distance.

6. Update Step:

After assigning students to clusters, the algorithm moves to the update step.
For each medoid m_j, the algorithm considers all non-medoid students x_i within the cluster.
It evaluates the cost of choosing each student x_i as a new medoid. The cost is the sum of distances from all students in the cluster to x_i.
If the total cost of using x_i as the new medoid is lower than the current cost of the medoid m_j, the medoid is updated to x_i.
This step ensures that the medoid is the most representative point for the cluster.

7. Convergence Check:

The Assignment and Update steps are repeated until:

- The medoids stabilize (i.e., no further changes occur).
- Alternatively, a maximum number of iterations is reached to avoid infinite loops.

8. Output Clusters:

Once convergence is achieved, the algorithm outputs the final clusters. These clusters represent groups of students with similar academic profiles and interests.

Each cluster is defined by its **medoid**, which is the most representative student of that cluster. Students are assigned to the cluster corresponding to the closest medoid, and each cluster can be analyzed further to provide insights into student behavior and preferences.

2.2 DATA/ DATA STRUCTURE

The project uses data categorized into:

Student Data:

- Academic History: Includes numerical attributes such as GPA, grades in completed courses (e.g., CPL, DSA, Python).
- Personal Preferences: Categorical attributes representing student interests in various specializations (e.g., Machine Learning, Cyber Security).
- Demographic Information: Data such as age, major, and year of study.

Performance Data: Contains quantitative metrics like test scores, attendance records, and interest scales that reflect student performance.

Specialization Data: A structured list of available specializations along with their requirements, which aids in the specialization selection process.

Grouping Data: Performance metrics used for clustering students into groups based on their academic performance levels.

Data Sources

- Primary Data: Collected through forms filled out by students during the enrollment process.
- Secondary Data: Synthetic data generated by the project team for testing purposes to ensure robustness in algorithm performance.

Sampling Technique

To ensure a fair representation of the student population:

First, students are divided into broad categories such as undergraduate and graduate levels, and further into academic years like freshmen, sophomores, juniors, and seniors. Within each category, a list of all students is compiled. A random number generator is then used to select a predetermined number or percentage of students from each list. Finally, the selected students from all categories are combined to form the final sample. This approach ensures a balanced representation of students from different academic levels while maintaining a straightforward and random selection process within each group.

Statistical Methods

Descriptive Statistics: Used to summarize trends in the data, such as **average GPA and common preferences** among students. These summaries help you understand patterns in the data without diving into the details of individual students. This broad view is valuable for educators and administrators to recognize general academic strengths, areas of interest, and trends in student performance.

Normalization: Applied to scale different metrics (e.g., grades, interest levels) to a consistent range for comparability. To ensure that no single metric (like test scores) dominates or skews the results. In our Program we have:

- The interest level is scaled from **1 to 5**.
- Grades are converted into equivalent numerical values.
- For students who provided "NC" for CGPA, the project substitutes **5** in place of "NC."

Outlier Detection: Outliers are data points that are significantly different from the rest of the dataset, such as a student with an exceptionally high or low GPA compared to others. These outliers can distort the results of algorithms if not handled properly.

To handle outliner Statistical techniques are employed to identify and handle outliers that could skew results in classification or clustering. Identify these extreme values and handle them by either removing them or adjusting their influence (e.g., capping the extremes).

This ensures that outliers don't skew the results or lead to incorrect conclusions in classification or clustering. For example, a student with an unusually low-test score might unduly influence clustering, leading to poor grouping decisions.

Data Structures

Database: The project employs a MongoDB database to store various types of data efficiently. This NoSQL database allows for flexible data modeling and easy retrieval of complex data structures.

Data Models:

- Student Model: Represents individual student records with fields for academic history, preferences, and demographic information.

- Performance Model: Captures performance metrics associated with each student, facilitating analysis for clustering.

- Specialization Model: Details available specializations, including prerequisites and related career paths.

Data Handling Structures:

- Arrays or lists to store collections of student records, performance metrics, and specialization options.

- Dictionaries or hash maps may be used to map student IDs to their respective records for quick access and updates.

2.3 SWOT ANALYSIS

Strengths

- Data-Driven Decision Making: The project leverages advanced algorithms for classification and clustering, enabling personalized recommendations that help students make informed choices regarding their academic paths.
- Personalized Learning Recommendations: By analyzing individual academic histories and preferences, the system provides tailored specialization suggestions, which can enhance student engagement and success rates.
- Enhanced Educator Support: The performance-based grouping feature allows educators to easily identify students who need additional help and recognize high achievers, facilitating targeted interventions and efficient resource allocation.
- Automated Performance-Based Grouping: The use of clustering algorithms simplifies the process of grouping students based on performance metrics, saving educators time and ensuring that students receive appropriate support tailored to their needs.

Weaknesses

- Dependence on Accurate Data Input: The system's effectiveness is highly reliant on the accuracy and completeness of input data; inaccuracies can lead to misleading recommendations and groupings.
- Complex Implementation: Integrating classification and clustering algorithms into existing educational systems can pose technical challenges, requiring significant expertise and resources.
- Initial Resistance from Stakeholders: There may be resistance from educators or administrative staff accustomed to traditional methods, which could hinder full adoption of the new system.

Opportunities

- Growing Demand for Personalized Education: As educational institutions increasingly focus on personalized learning experiences, there is a significant opportunity to expand the system's adoption across various educational settings.
- Potential Partnerships with Institutions and EdTech Companies: Collaborating with educational organizations and technology firms can enhance system functionality and facilitate broader implementation.
- Expansion into Other Educational Areas: The methodologies employed in this project can be adapted for use in other educational contexts, such as vocational training or online learning platforms.

Threats

- Technological Advancements: Rapid changes in technology could outpace the system's development, necessitating ongoing updates and adaptations to remain relevant.
- Competition from Other Solutions: The emergence of alternative educational technologies offering similar functionalities may pose a threat to the project's market position.
- Data Privacy Concerns: Increasing scrutiny over data privacy and security could impact the system's acceptance among institutions that prioritize student data protection.

2.4 PROJECT FEATURES

The **Academic Monitoring and Specialization Selection System** is a comprehensive solution designed to address the challenges faced by students and educators in academic decision-making. The key features of the system include:

1. User Authentication and Role-Based Access

- Secure login and registration for students, educators, and administrators.
- Role-based access provides tailored menus based on user type:
  - **Student Menu**: Options for specialization recommendation.
  - **Educator Menu**: Options for performance-based clustering.
  - **Admin Menu**: Comprehensive administrative controls, including account management and advanced functionalities.

2. Specialization Recommendation

- Uses a **decision tree classification algorithm** to analyze student academic performance and interests.
- Provides personalized specialization recommendations that align with student strengths and career aspirations.
- Displays results in an easy-to-understand format, helping students make informed decisions.

3. Performance-Based Grouping

- Implements the **K-Medoids clustering algorithm** to group students based on their academic performance.
  - Clustering facilitates efficient resource allocation:
  - High-performing students are given advanced content.
- Students needing support receive tailored assistance.
- Displays clustering results in a tabular format for better visualization.

4. Comprehensive Admin Controls

Admins can:

- Create educator accounts.
- View, edit, and delete student and educator records.
- Execute both classification and clustering functionalities.
- Manage the overall database and ensure data integrity.

5. Data Management

- Stores academic records, performance metrics, and user details in a secure **MongoDB database**.
- Efficient CSV file handling for importing and exporting data, especially for clustering purposes.

6. User-Friendly Interface

- A menu-driven command-line interface ensures seamless navigation and ease of use.
- Provides clear feedback and displays results for user-selected options.

7. Algorithmic Efficiency

- Ensures accurate and reliable outputs with minimum computational cost through optimized algorithms.
- Supports multi-iteration clustering for enhanced accuracy.

8. Scalability and Extendibility

- Designed to handle large datasets and multiple user interactions simultaneously.
- Easily extendable to integrate additional features or data sources in the future.

This system bridges the gap between student aspirations and academic opportunities while streamlining educator responsibilities, creating a cohesive and efficient academic ecosystem.

2.5 USER CLASSES AND CHARACTERISTICS

The **Academic Monitoring and Specialization Selection System** serves multiple user classes, each with distinct roles and requirements. The system is designed to cater to the following user groups:

1. Students

Description: Students are the primary users of the system. They provide their academic history and preferences through forms to receive personalized recommendations for specialization paths.
Characteristics:

- **Demographics**: Typically, undergraduate or graduate students.
- **Technical Proficiency**: Varies from beginner to intermediate; familiar with basic user interfaces.
- **Goals**:
  - To Obtain accurate guidance for choosing specializations aligned with their strengths and career aspirations.
  - View and analyze their academic performance metrics.
- **Tasks**:
  - Register and log in to the system.
  - Input academic and interesting data for analysis.
  - Access specialization recommendations.

2. Educators

Description: Faculty members or academic advisors responsible for monitoring student performance and providing guidance. Utilize the system for grouping students based on their academic performance.
Characteristics:

- **Demographics**: Faculty members with a focus on mentoring and resource allocation.
- **Technical Proficiency**: Intermediate; comfortable with educational tools and data-driven platforms.
- **Goals**:
  - Efficiently group students for tailored teaching strategies.
  - Use clustering results to identify students who need additional support or advanced challenges.
- **Tasks**:
  - Log in to access educator functionalities.
  - Perform clustering to group students by performance.
  - Review clustering results to make data-driven educational decisions.

3. Administrators

Description: System administrators are responsible for managing users, maintaining data integrity, and ensuring smooth operation of the platform.
Characteristics:

- **Demographics**: IT professionals or designated staff members with advanced technical skills.
- **Technical Proficiency**: Advanced; knowledgeable in database management, system configuration, and user role management.
- **Goals**:
  - Manage accounts for students and educators.
  - Oversee and maintain the integrity and security of the system.
  - Ensure the system is functioning efficiently and meeting user needs.
- **Tasks**:
  - Create and manage user accounts.
  - Access and edit student and educator data.
  - Perform classification and clustering for analysis.
  - Manage database interactions and ensure data accuracy.
  - Monitor system performance and address technical issues

3. DESIGN

3.1 DESIGN AND IMPLEMENTATION CONSTRAINTS

Clustering Component

Hardware Boundary Conditions

- **Timing Requirements**: The clustering algorithm should process and group student data within a maximum response time of 10 milliseconds to ensure a responsive user experience.
- **Memory Requirements: The** application must efficiently manage up to 10,000 student records, requiring a minimum of 4 GB of RAM during peak operations.

Interfaces to Other Applications

- **Database Management System**: MongoDB for storing and retrieving student data used in clustering.

Specific Technologies and Tools to Be Used

- **Programming Language**: Java, utilizing libraries such as OpenCSV for CSV data handling and collections for managing student records.
- **Development Tools**:
  - IDE: IntelliJ IDEA or Eclipse.

Parallel Operations

- The clustering process should be capable of executing in parallel for multiple student datasets, using Java's multi-threading capabilities to enhance processing efficiency.

Language Requirements

- Primary language: Java for implementation; English for the user interface.

Security Considerations

- Implement user authentication to ensure secure access to clustering functionalities.

- Use data encryption for sensitive information to protect student data during transmission and storage.

Design Conventions or Programming Standards

- Follow Java naming conventions and modular design practices for maintainable and readable code.
- Include Javadoc comments for documentation of methods and classes.

Classification Component

Hardware Boundary Conditions

- **Timing Requirements:** The classification algorithm should generate specialization recommendations within a maximum of 5 seconds to maintain user engagement.
- **Memory Requirements**: The application should efficiently handle and analyse up to 10,000 student profiles, requiring around 4 GB of RAM for optimal performance.

Interfaces to Other Applications

- **Database Management System**: MongoDB for storing academic records and user profiles relevant to classification tasks.
- **Integration with External APIs**: Possible integration with academic systems or platforms that provide additional data for improving classification accuracy.

Specific Technologies and Tools to Be Used

- **Programming Language**: Java, employing libraries for machine learning (e.g., Weka or Smile) to implement the decision tree algorithm.
- **Development Tools**:
  - IDE: IntelliJ IDEA or Eclipse.

Parallel Operations

- The classification process should be capable of handling multiple user requests simultaneously, utilizing multi-threading in Java for efficient processing.

Language Requirements

- Primary language: Java; English for the user interface.

Security Considerations

- Ensure user authentication for secure access to classification functionalities.
- Encrypt sensitive user data to safeguard personal and academic information during both transmission and storage.

Design Conventions or Programming Standards

- Adhere to Java coding standards, including meaningful variable naming and modular code structure for ease of maintenance.
- Implement Javadoc comments for public methods and classes for clear documentation.

Data accuracy

- Training Data Quality: The effectiveness of the Decision Tree depends on accurate and representative training data. Biases or inaccuracies can lead to poor specialization recommendations.
- Student Data Accuracy: Ensuring the accuracy of student data (academic history, preferences, demographics) is vital. Errors or missing information can adversely affect classification results

3.2 DESIGN DIAGRAMS

3.2.1 FLOW CHART

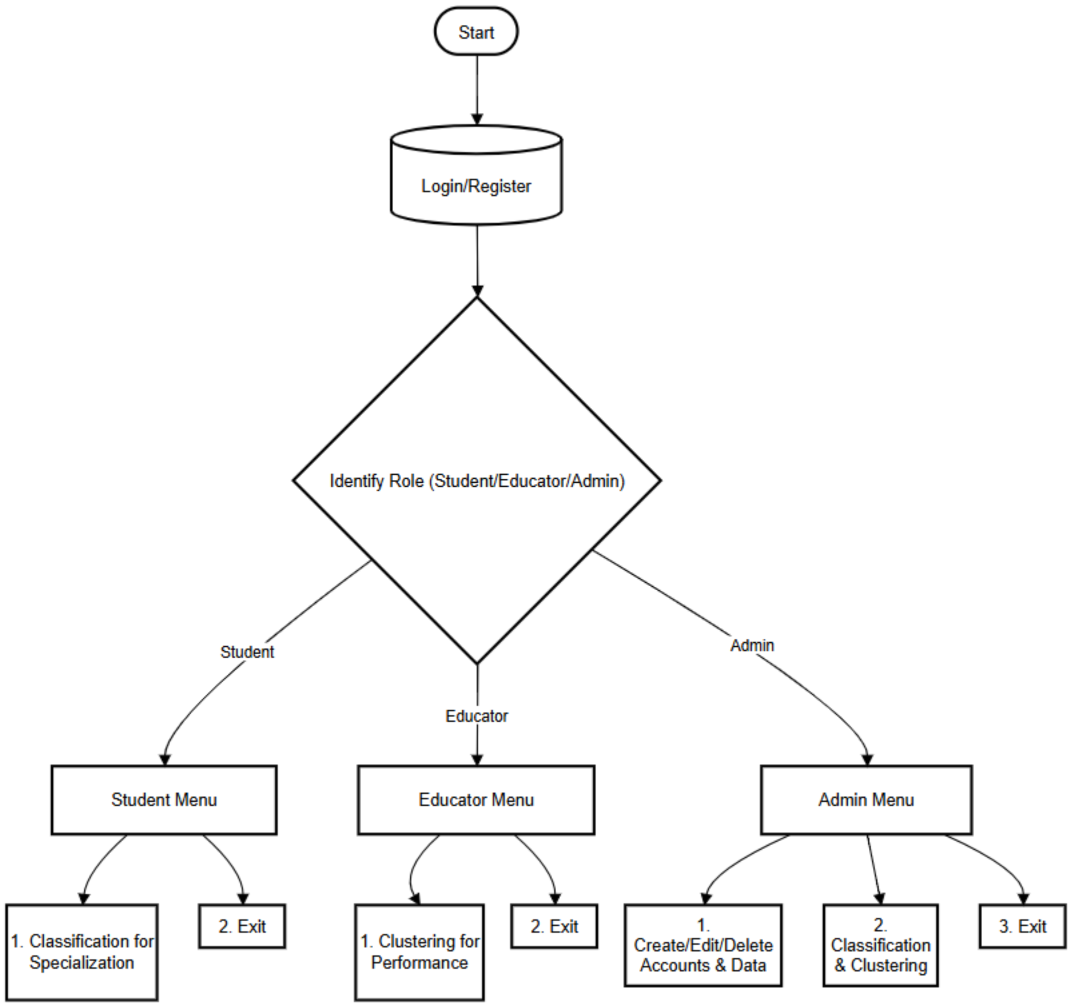

Figure 3-1 Flow Chart

3.2.2 USE CASE MODEL FOR REQUIREMENT ANALYSIS

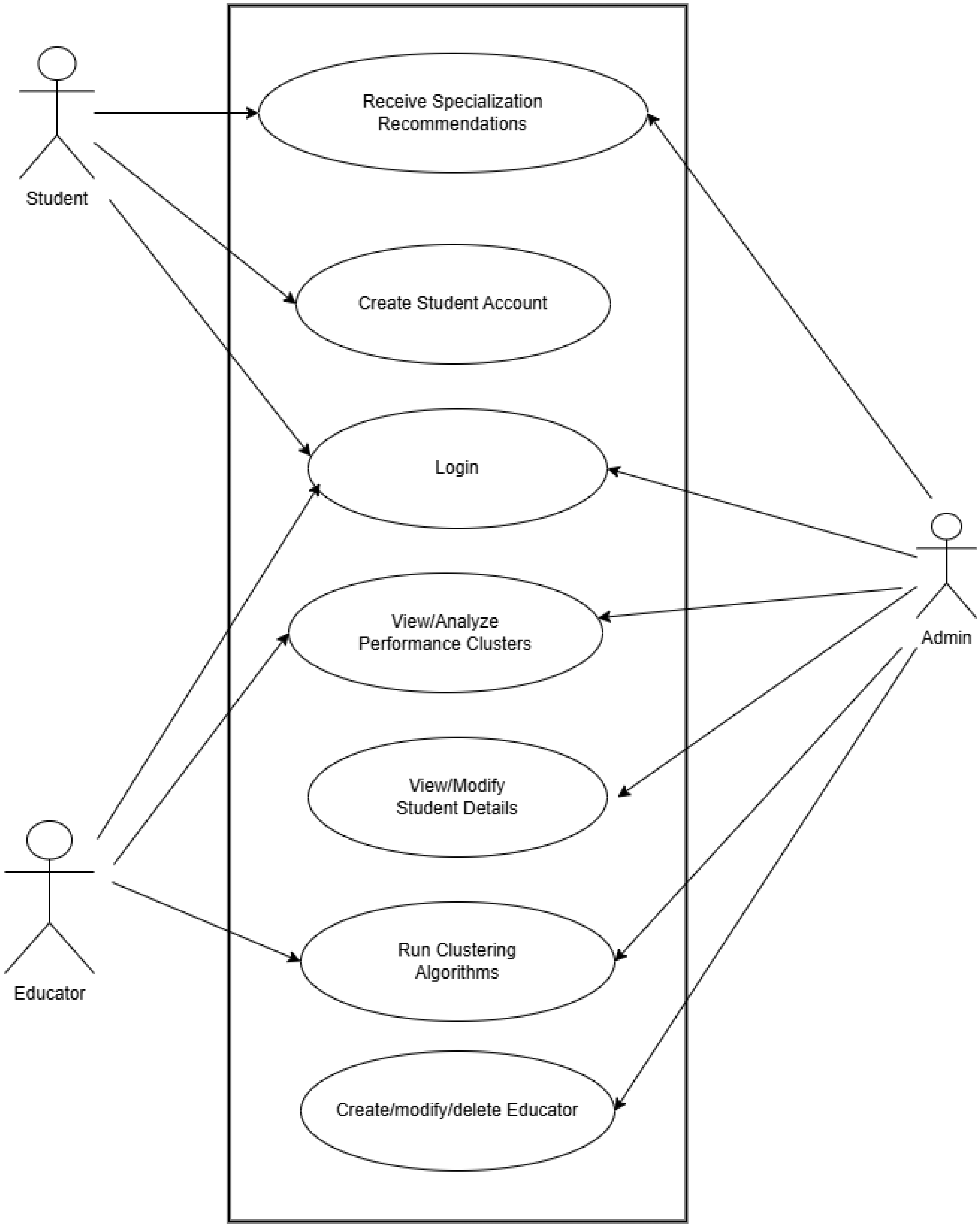

Figure 3-2 Use Case Diagram

3.2.3 CLASS DIAGRAM

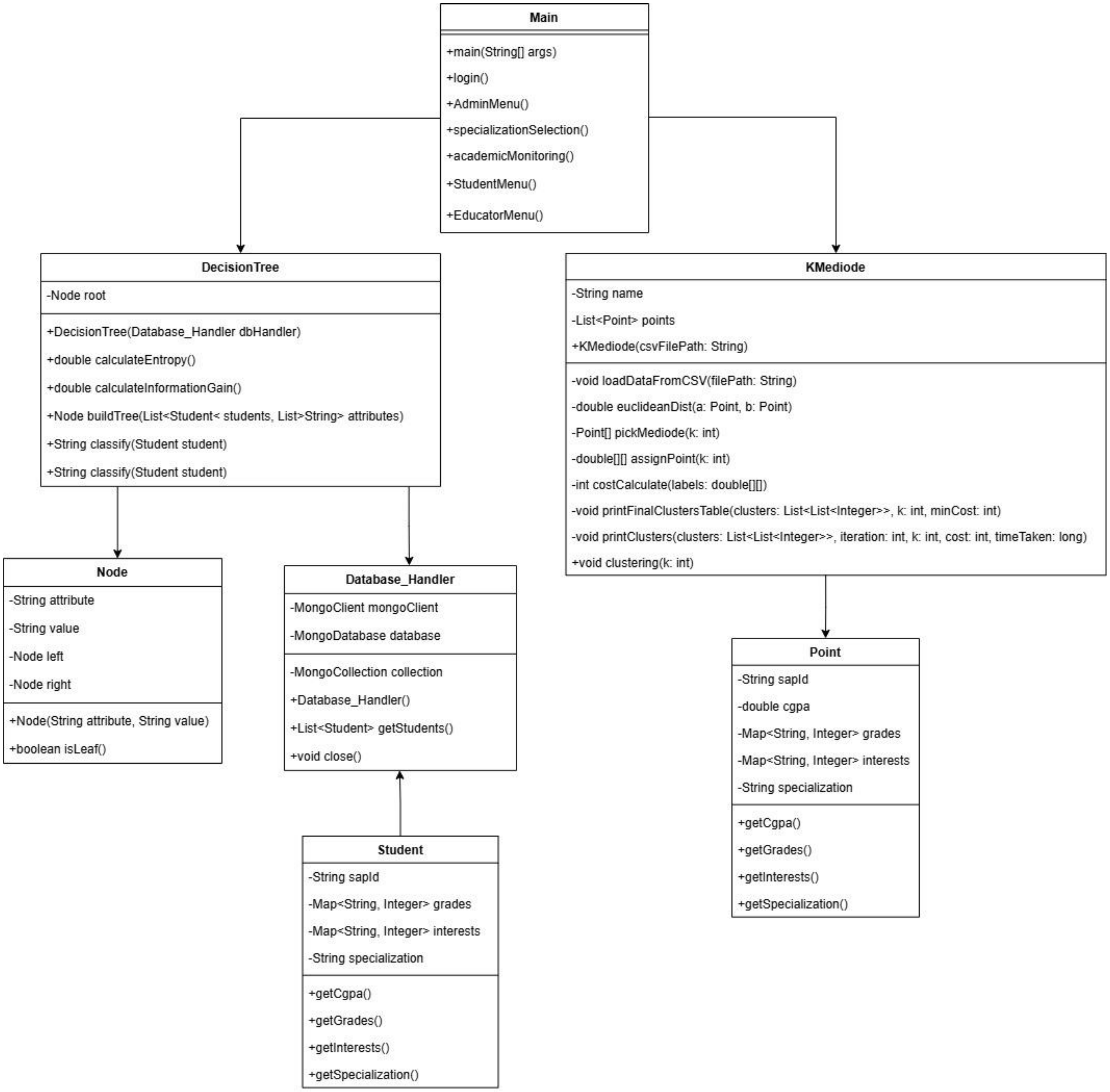

Figure 3-3 Class Diagram

3.2.4 DATA FLOW DIAGRAM

Level - 0 data flow diagram:

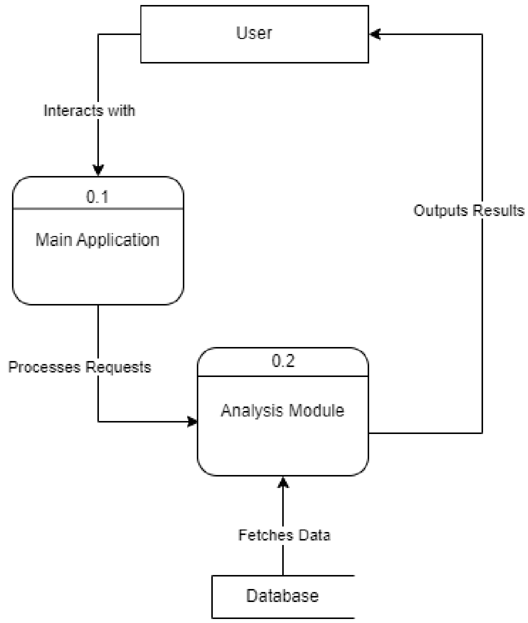

Figure 3-4 Level -0 Data Flow Diagram

Level - 1 data flow diagram:

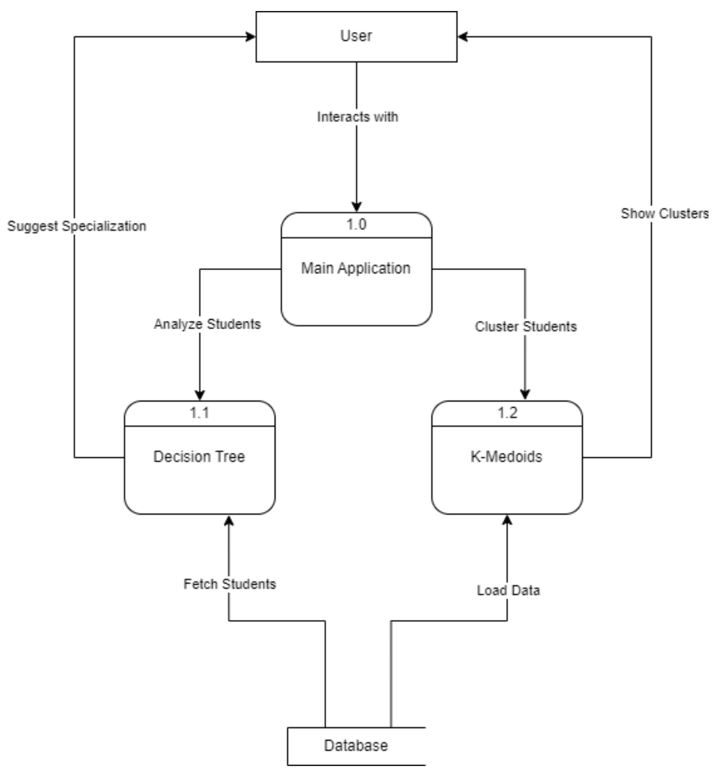

Figure 3-5 Level -1 Data Flow Diagram

Level - 2 data flow diagram:

Level – 2.1 data flow diagram of Specialization recommender:

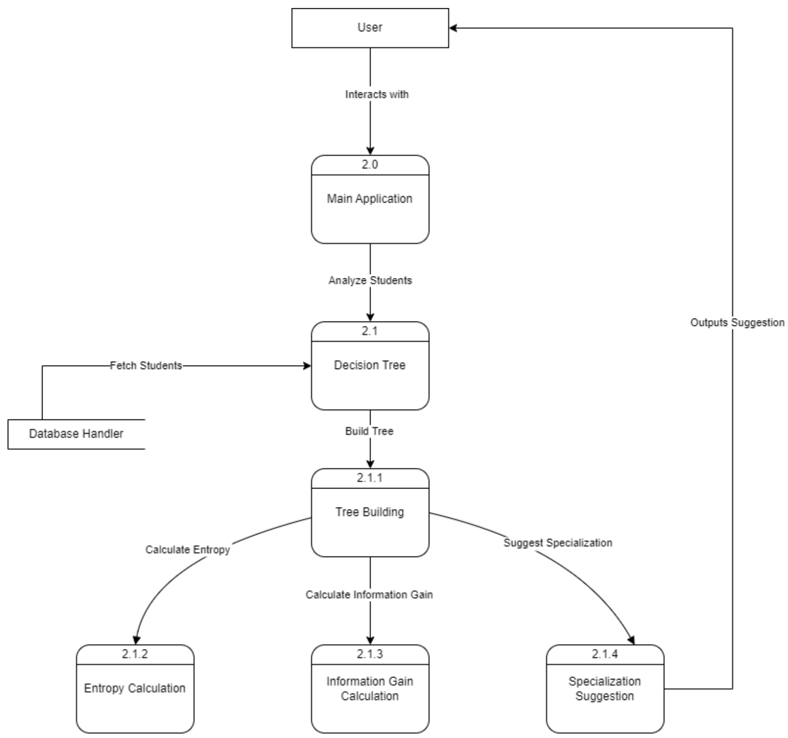

Figure 3-6 Level – 2.1 data flow diagram of Specialization recommender

Level – 2.2 data flow diagram of Academic monitoring:

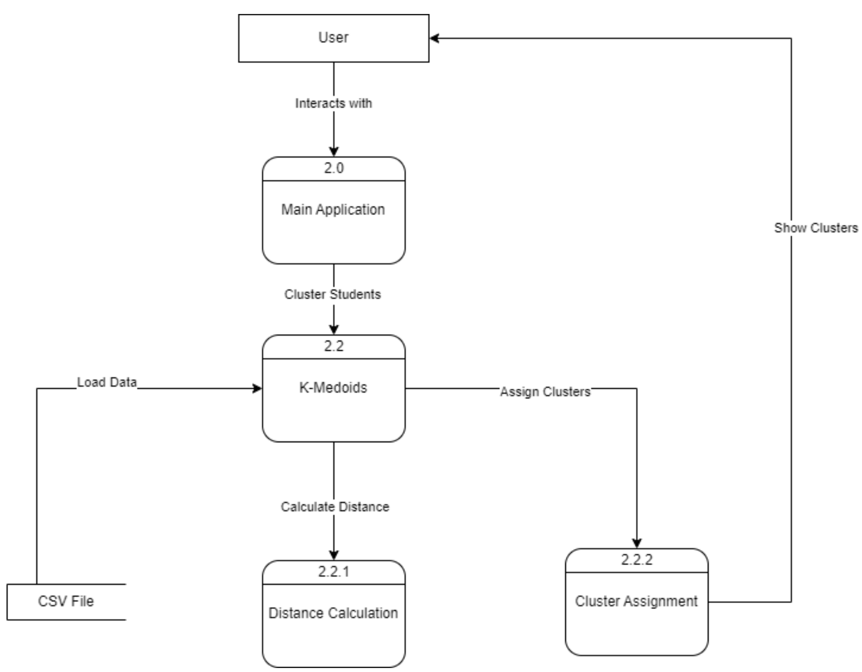

Figure 3-7 Level – 2.2 data flow diagram of Academic Monitoring

3.2.5 ACTIVITY DIAGRAM

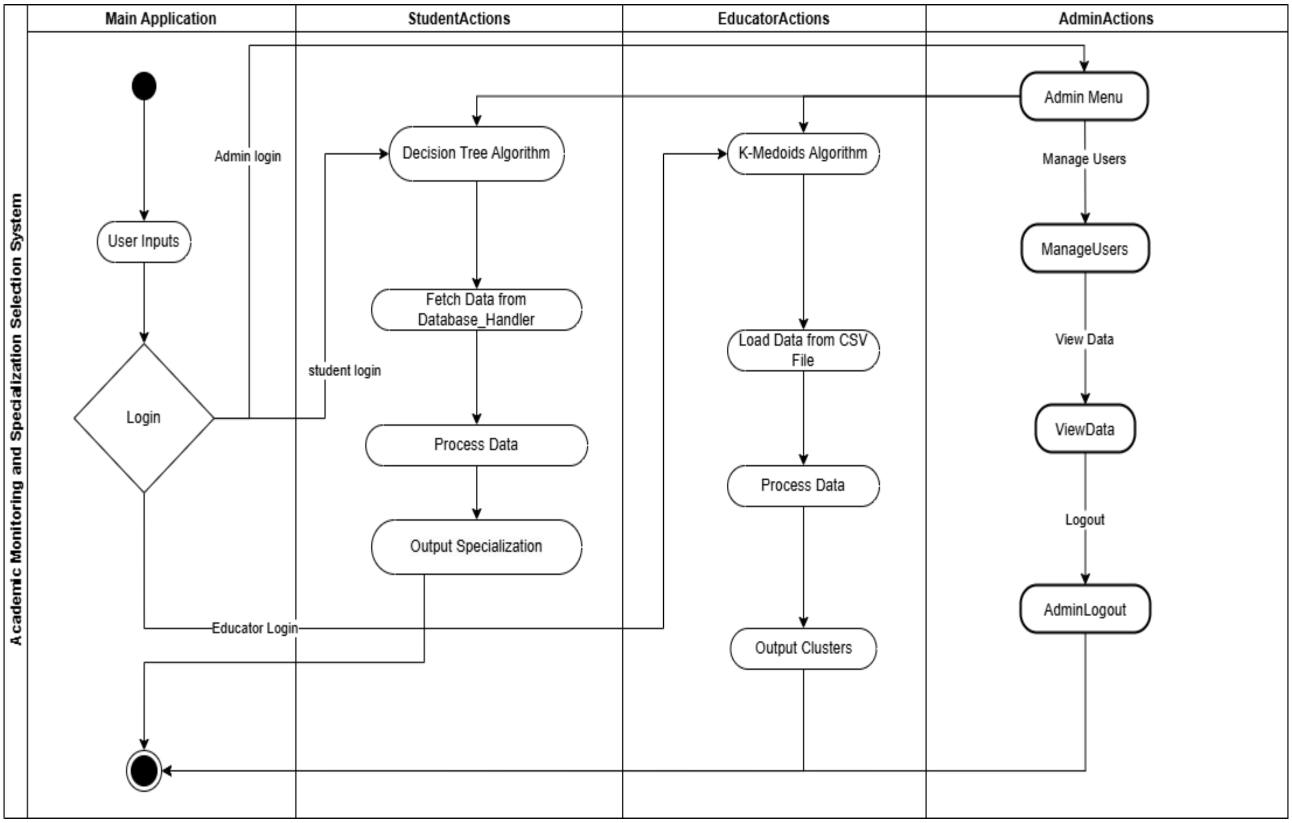

Figure 3-8 Activity Diagram

3.2.6 STATE DIAGRAM

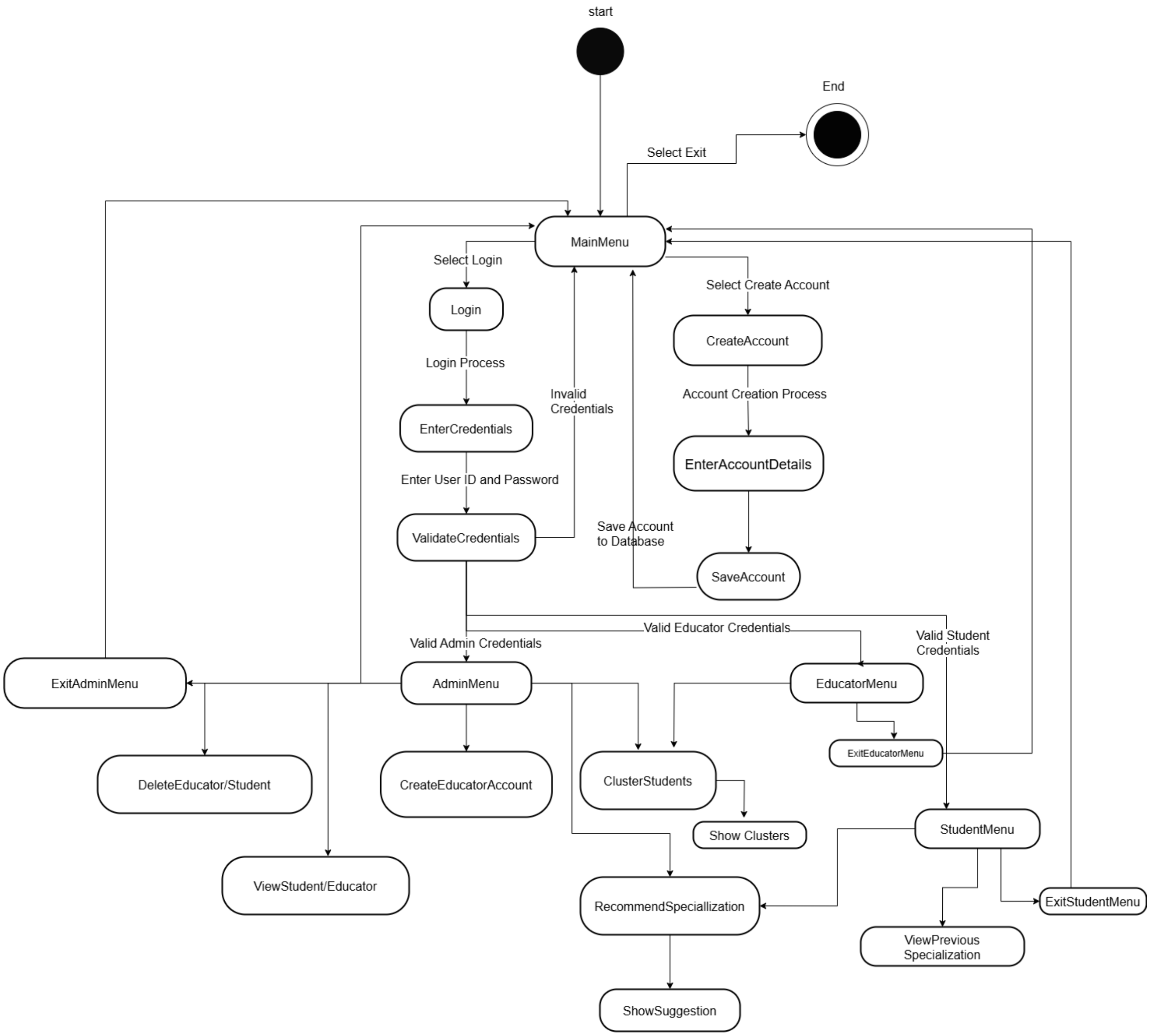

Figure 3-9 State Diagram

3.2.7 SEQUENCE DIAGRAM

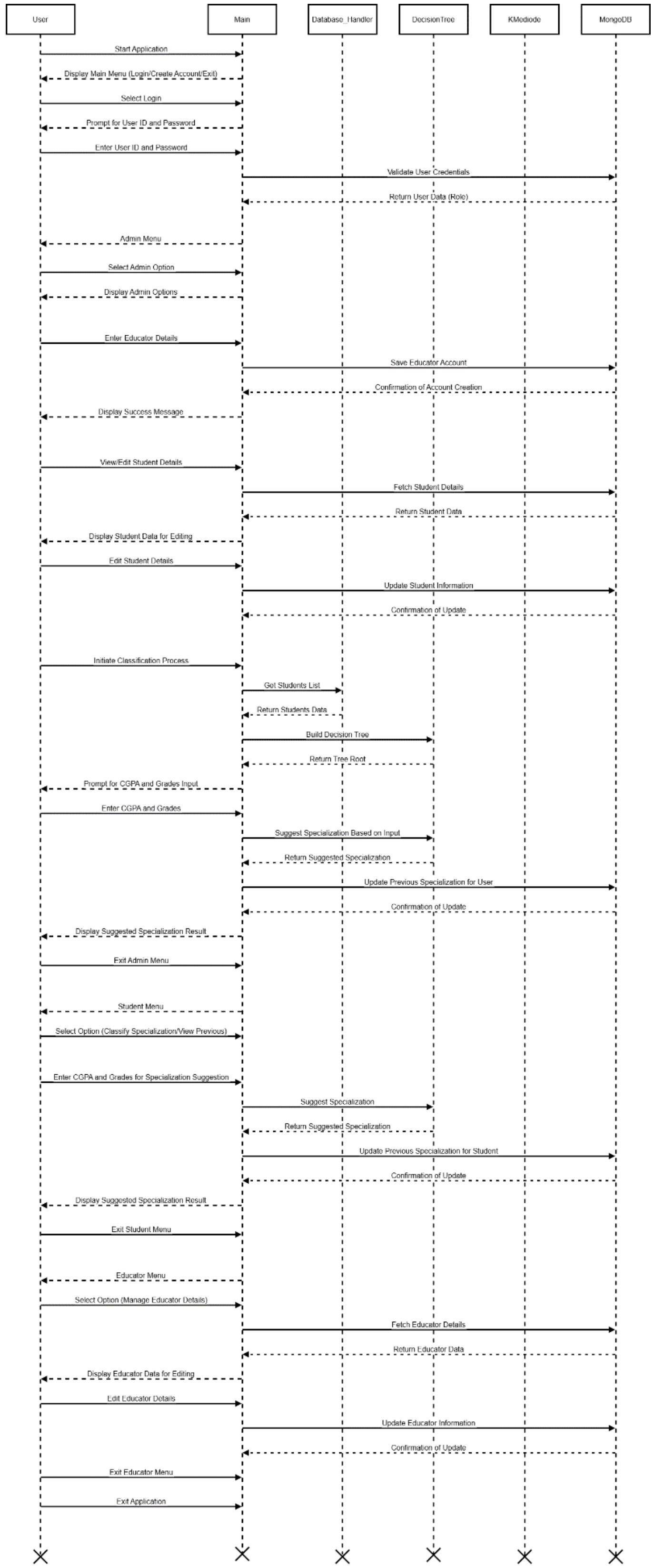

Figure 3-10 Sequence Diagram

3.2.8 ENTITY RELATION DIAGRAM

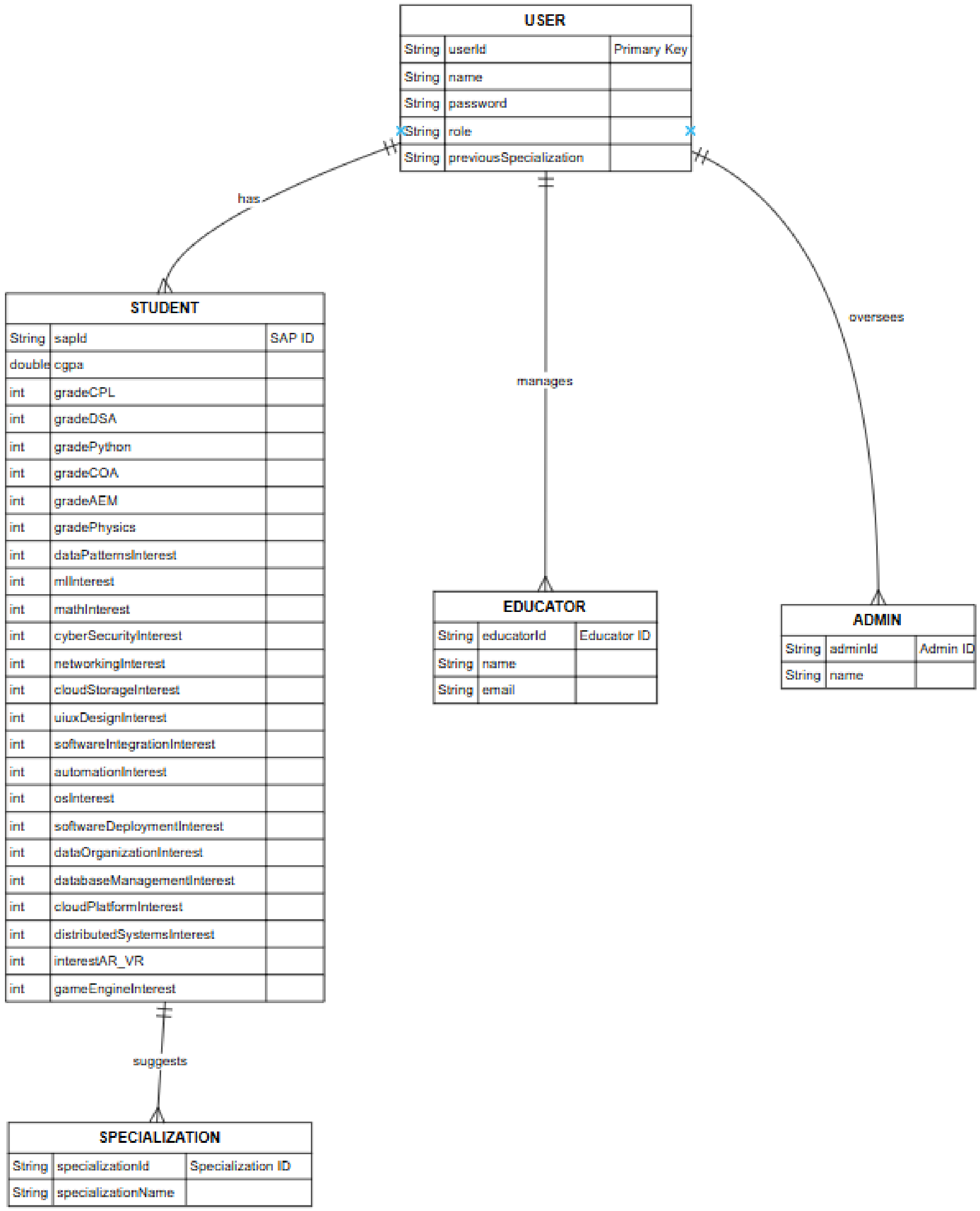

Figure 3-11 Entity relation diagram

4. SYSTEM REQUIREMENTS

4.1 HARDWARE REQUIREMENT

Minimum Requirements

- Processor: Intel Core i3 or equivalent
- RAM: 4 GB
- Storage: 500 GB HDD or SSD
- Network: Broadband internet connection for database access (MongoDB)

- Graphics: Integrated graphics (for basic UI rendering)

Recommended Requirements

- Processor: Intel Core i5 or equivalent
- RAM: 8 GB or more
- Storage: 1 TB SSD for faster data access and application performance
- Network: Stable broadband internet connection with low latency
- Graphics: Dedicated graphics card (optional, for enhanced UI performance)

Additional Considerations

- Operating System: Windows 10/11, macOS, or a compatible Linux distribution.
- Development Tools: Java Development Kit (JDK) installed for running Java applications, along with an IDE like IntelliJ IDEA or Eclipse.
- Database Server: MongoDB server installed locally or accessible via a cloud service.

4.2 USER INTERFACE

Below attached how the user interface of the project looks like

1. Running the Project First asks user to Login/Register

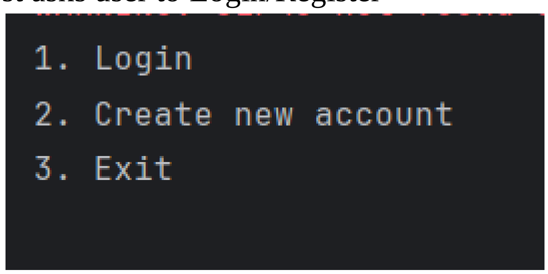

Figure 4-1 Login /Register Menu

2. Now the new user creates account and logs in (also shows error message, if ID is invalid)

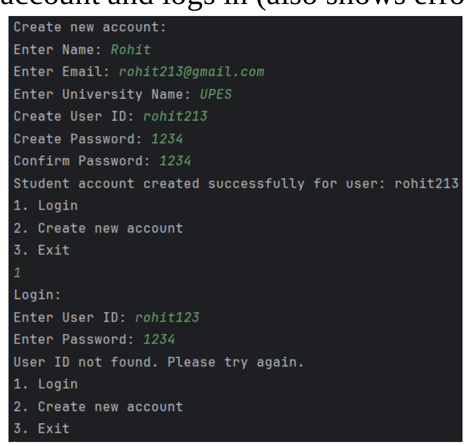

Figure 4-2 User creates account

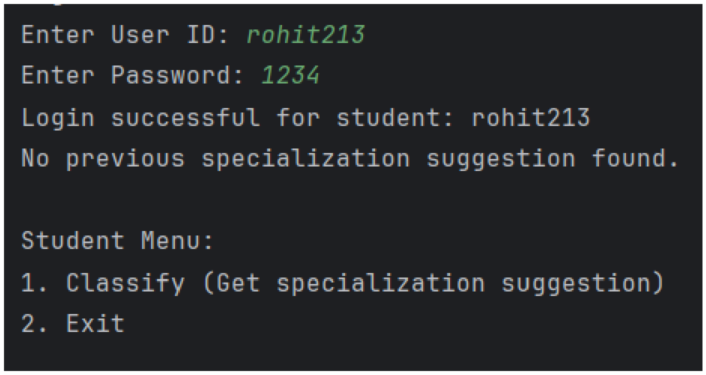

Figure 4-3 User's account

4.3 SOFTWARE INTERFACE

The software interface of the Academic Monitoring System is designed to facilitate user interactions through a menu-driven approach. The interface caters to three main types of users: Admins, Students, and Educators. Below is a detailed description of the various components and functionalities of the interface.

1. Main Menu

Purpose: Serves as the entry point for all users.

Options:

- Login: Directs users to the login screen.
- Create a New Account: Allows new users (students) to register.
- Exit: Closes the application.

2. Login Screen

Functionality: Users can enter their credentials to access the system.

Fields:

- User ID: Input field for entering the user ID.
- Password: Input field for entering the password.

Actions:

- Submit: Validates credentials against the database.
- If successful, it directs users to their respective menus based on their role (Admin, Student, Educator).
- If unsuccessful, displays appropriate error messages.

3. Account Creation Screen (For Students)

Purpose: Allows students to create a new account.

Fields:

- Name: Input field for entering the student's name.
- Email: Input field for entering the student's email address.
- Password: Input field for setting a password.
- Confirm Password: Input field for confirming the password.

Actions:

- Create Account: Saves account details to the database and confirms registration.

- Returns to the main menu upon completion.

4. Admin Menu
Purpose: Provides administrative functionalities for managing users and accounts.
Options:

- Create Educator Account: Prompts admin to enter details for creating a new educator account.
- View/Edit Student Details: Displays a list of students with options to edit their information.
- View/Edit Educator Details: Displays a list of educators with options to edit their information.
- Delete User: Prompts admin for a user ID to delete from the system.
- Classify Specialization: Initiates the classification process for student specialization suggestions.
- Cluster Students: Initiates clustering based on student performance data.
- Exit Admin Menu: Returns to the main menu.

5. Student Menu
Purpose: Provides functionalities specific to students after login.
Options:

- Classify Specialization: Prompts students for input data (CGPA, grades) and displays suggested specialization based on their inputs.
- View Previous Specialization Suggestion: Displays any previously suggested specialization for the student.
- Exit Student Menu: Returns to the main menu.

6. Educator Menu
Purpose: Provides functionalities specific to educators after login.
Options:

- Manage Educator Details: Displays educator information with options to edit or update details.
- Exit Educator Menu: Returns to the main menu.

Confirmation and Error Messages
After successful actions (e.g., login, account creation):

- "Login successful! Welcome, [Username]."
- "Account created successfully! You can now log in."

For errors:

- "Invalid choice! Please try again."
- "Credentials are incorrect! Please try again."
- "An unexpected error occurred: [Error Message]."

4.4 DATABASE INTERFACE

The database interface for the Academic Monitoring System is designed to facilitate interaction with the MongoDB database. This interface handles user authentication, account creation, data retrieval, and updates for students and educators.

1. Database Connection Setup

MongoDB Client Initialization: The MongoDB client is initialized using the connection string to connect to the local MongoDB instance.

The database used is named userDatabase, and the collection for user data is called users.

2. User Authentication

Login Method: This method validates user credentials against the MongoDB collection.

It checks if the user ID exists and verifies the password.

Depending on the role (Admin, Student, Educator), it directs users to their respective menus.

3. Account Creation

This method allows new students to create an account by entering their details. It saves the new user's information (name, email, password, role) into the user's collection.

4. User Management (Admin Functions)

- Admins can create accounts for educators by entering their details.
- This method saves educator information into the same user's collection with an appropriate role.
- View/Edit User Details: Admins can retrieve and modify details of both students and educators.
- This includes fetching user data based on user ID and updating it as necessary.
- Delete User: Admins can delete users from the system by specifying their user ID.

5. Specialization Classification

This method retrieves student data for classification purposes.

It uses a decision tree algorithm to suggest specializations based on student inputs (CGPA, grades).

6. Clustering Students

This method allows admins to perform clustering operations on student performance data using K-Means or similar algorithms.

7. Data Retrieval and Updates

Update Previous Specialization: After classification, this method updates the student's record with their suggested specialization in the database.

4.5 PROTOCOLS

The protocols for the Academic Monitoring System outline the communication standards and procedures for interactions between users and the system, as well as between different components of the system itself. These protocols ensure that data is handled securely and efficiently while providing a seamless user experience.

1. User Authentication Protocol

Purpose: To verify user identity before granting access to the system.

Process:

- Users enter their credentials (User ID and Password).
- The system validates these credentials against stored data in the MongoDB database.
- If credentials are valid, the user is granted access based on their role (Admin, Student, Educator).
- If invalid, an error message is displayed, prompting the user to retry.

2. Account Creation Protocol

Purpose: To allow new users (students) to register for an account.

Process:

- Users provide the required information (Name, Email, Password).
- The system checks for existing accounts with the same User ID or Email.
- If no conflicts are found, the new account is created and stored in the MongoDB database.
- A confirmation message is displayed to the user.

3. Admin Management Protocol

Purpose: To enable Admin users to manage accounts and data.

Process:

- Admin can create educator accounts by providing necessary details.
- Admin can view, edit, or delete student and educator accounts.
- Admin can initiate classification for specialization suggestions based on student data.
- All changes are logged in the database to maintain an audit trail.

4. Data Retrieval Protocol

Purpose: To fetch user data from the database as needed.

Process:

- When a user logs in, their details are retrieved from the MongoDB database using their User ID.
- For admin functions, relevant user details are fetched based on specified criteria (e.g., all students or specific educator details).

5. Specialization Classification Protocol

Purpose: To suggest specializations for students based on their academic performance and interests.

Process:

- Students input their academic details (CGPA, grades) and interest levels.
- The system processes this information using a decision tree algorithm to classify suitable specializations.

- Suggestions are displayed to the student, and previous suggestions are updated in their profile.

6. Clustering Protocol
Purpose: To group students based on performance metrics for analysis purposes.
Process:

- Admin selects the number of clusters (between 1 and 10).
- The clustering algorithm processes student performance data from a CSV file.
- Results are displayed to the admin for further action or analysis.

7. Error Handling Protocol
Purpose: To manage unexpected errors gracefully during user interactions.
Process:

- The system captures exceptions during operations (e.g., input mismatches or database errors).
- Informative error messages are displayed to guide users on how to correct their inputs or understand issues.

5. IMPLEMENTATION

5.1 PROPOSED SYSTEM

The proposed system, the Academic Monitoring System, is designed to enhance the management of student and educator accounts in an educational environment. It utilizes a menu-driven interface to facilitate user interactions, allowing students to register, log in, and receive specialization suggestions based on their academic performance and interests. Educators, on the other hand, have restricted access and cannot create their accounts directly; instead, this functionality is reserved for administrators. The system is built using Java and integrates with a MongoDB database for efficient data storage and retrieval. Key features include user authentication, account management, specialization classification using decision trees, and clustering of students based on performance metrics. Admins can create educator accounts, manage user details, and oversee the classification process. The system emphasizes usability, security, and robust data handling, providing a comprehensive solution for academic institutions to monitor and guide students effectively.

5.2 MODULES

The Academic Monitoring System is structured into several key modules, each serving a specific purpose to facilitate user interactions and data management effectively. Below are the proposed modules:

1. User Authentication Module
Purpose: This module handles user login and account creation.
Functions:

- Validate user credentials against the MongoDB database.
- Allow students to create new accounts.
- Implement error handling for invalid inputs.

2. Admin Management Module

Purpose: Provides administrative functionalities for managing users.

Functions:

- Create educator accounts.
- View, edit, or delete student and educator details.
- Classify specializations for students based on their academic performance.
- Cluster students based on performance metrics.

3. Student Management Module

Purpose: Allows students to manage their profiles and access academic resources.

Functions:

- Classify specialization suggestions based on input data (CGPA, grades, interests).
- View previous specialization suggestions.

4. Educator Management Module

Purpose: Enables educators to manage their details and access relevant functionalities.

Functions:

Edit personal information as needed.

5. Data Processing Module

Purpose: Handles the processing of data for classification and clustering.

Functions:

- Implement decision tree algorithms for specialization classification.
- Execute clustering algorithms (e.g., K-Means) on student performance data.

6. Database Interface Module

Purpose: Manages interactions with the MongoDB database.

Functions:

- Perform CRUD (Create, Read, Update, Delete) operations on user accounts and academic data.
- Ensure secure connections to the database and handle exceptions.

5.2.1 CLASSIFICATION USING DECISION TREE

This module employs the ID3 algorithm to classify students into suitable specializations based on their academic performance and preferences.

Purpose:

To assist students in identifying the most appropriate specialization.

Functions:

- Analyze student data such as CGPA, grades, and interests.

- Generate specialization recommendations using a decision tree classifier.
- Present classification results in an interpretable format.

5.2.2 CLUSTERING USING K MEDOID

This module uses K-Medoids clustering to group students based on performance metrics for academic analysis.

Purpose:
To identify performance-based clusters of students for better academic management and support.

Functions:

- Group students into clusters based on academic data (e.g., test scores, assignment performance).
- Provide insights into student performance trends.
- Allow administrators to analyze the clusters for targeted interventions.

6. OUTPUT SCREENS

Educator's Menu

```
1. Login
2. Create new account
3. Exit
1
Login:
Enter User ID: ds123
Enter Password: 1234
Login successful for educator: ds123

Educator Menu:
1. Cluster (Group students based on performance)
2. Exit
1
```

Figure 6-1 Educator's menu

```
=====================================================================================
** Welcome to Academic Monitoring **
=====================================================================================

Enter the number of clusters (must be between 1 and 10): 5
Random Medoids Selected: 500082772 500108706 500102243 500083620 500106041
```

Figure 6-2 Educator choose option 1

```
========================================
Final Clusters After 5 Iterations (Table Format)
========================================
| Cluster 1           | Cluster 2           | Cluster 3           | Cluster 4           | Cluster 5           |
|---------------------|---------------------|---------------------|---------------------|---------------------|
| 500108020           | 500105401           | 500093950           | 500107761           | 500110490           |
| 500107565           | 500107156           | 500083620           | 500110607           | 500108261           |
| 500109754           | 500106951           | 50010326            |                     | 500109927           |
| 500108707           | 500109497           |                     |                     | 500101726           |
| 500105642           | 500107049           |                     |                     | 500101897           |
| 500107193           | 500096292           |                     |                     | 500101970           |
| 500111697           | 500082524           |                     |                     | 500102243           |
| 500109330           | 500107769           |                     |                     | 500105545           |
| 500105700           |                     |                     |                     | 500110794           |
| 500105682           |                     |                     |                     | 500109627           |
| 500093995           |                     |                     |                     | 500105016           |
| 500083540           |                     |                     |                     | 500108706           |
| 500106041           |                     |                     |                     | 500107615           |
| 500108348           |                     |                     |                     | 500107148           |
| 500108601           |                     |                     |                     | 500107366           |
| 500102244           |                     |                     |                     | 500082772           |
|                     |                     |                     |                     | 500109805           |
|                     |                     |                     |                     | 500107098           |
|                     |                     |                     |                     | 500108342           |
|---------------------|---------------------|---------------------|---------------------|---------------------|
==========================================================================================================================
                                                                            The minimum cost achieved in clustering is: 223
==========================================================================================================================
```

Figure 6-3 This is the visualization of the clusters.

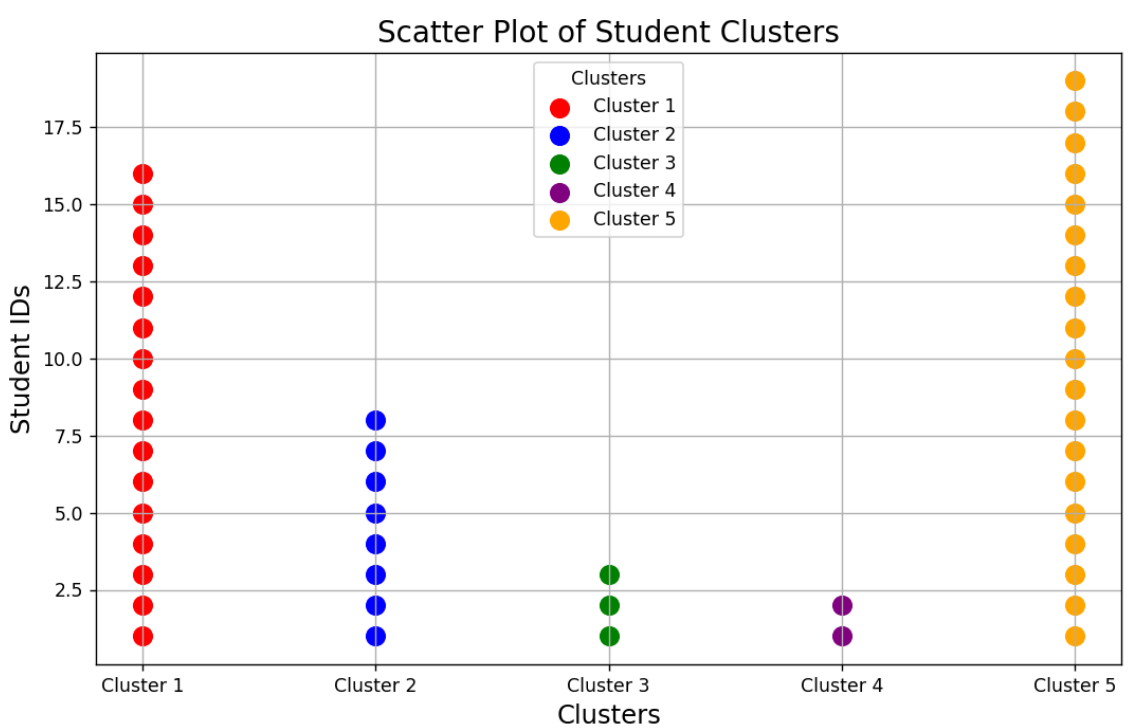

Figure 6-4 Scatter Plot of the clusters.

Student's Menu

The student if already registered will log in and will get recommendation based on his/her inputs

```
1. Login
2. Create new account
3. Exit
1
Login:
Enter User ID: stud1
Enter Password: 1234
Login successful for student: stud1
Your previous specialization suggestion was: BIG DATA

Student Menu:
1. Classify (Get specialization suggestion)
2. Exit
1
```

Figure 6-5 Students Menu

```
==========================================================================================
                    ** Welcome to Specialization Predictor **
==========================================================================================


    Enter SAP ID: 5001050001

    Enter your overall 1st year CGPA: 7

    Enter Grade for C Programming Language (CPL) (O, A+, A, B+, B, C+, C, D, F): B

    Enter Grade for Data Structures and Algorithms (DSA) (O, A+, A, B+, B, C+, C, D, F): a

    Enter Grade for Python Programming (O, A+, A, B+, B, C+, C, D, F): b

    Enter Grade for Computer Organization and Architecture (COA) (O, A+, A, B+, B, C+, C, D, F): a

    Enter Grade for Automata and Engineering Mathematics (AEM) (O, A+, A, B+, B, C+, C, D, F): b

    Enter Grade for Physics (O, A+, A, B+, B, C+, C, D, F): a
```

```
Do you enjoy working with data to uncover patterns or trends? (1-5) : 1

Are you interested in learning how machines can be trained to recognize objects or predict outcomes? (1-5) : 1

Do you find solving complex mathematical problems and using algorithms exciting? (1-5) : 1

Are you curious about how to protect data and systems from cyber threats? (1-5) : 1

Do you enjoy understanding how computer networks work and ensuring their reliability? (1-5) : 1

Are you interested in learning how data is stored and accessed securely in the cloud? (1-5) : 1

Do you enjoy designing and building user interfaces for websites or applications? (1-5) : 1

Are you interested in learning how different parts of a software system (front-end and back-end) communicate? (1-5) : 2

Are you interested in automating tasks and making processes more efficient? (1-5) : 2

Do you enjoy working with operating systems and exploring how software runs on them? (1-5) : 2

Are you curious about integrating and deploying software systems in real-time environments? (1-5) : 2

Do you enjoy working with large sets of data and organizing them efficiently? (1-5) : 2

Are you interested in understanding how data is stored, queried, and managed in databases? (1-5) : 3

Do you like exploring cloud platforms and understanding how they store and process data? (1-5) : 3

Are you interested in learning how distributed systems process big data across multiple machines? (1-5) : 3

Are you interested in creating visual designs, animations, AR, VR or working with game mechanics? (1-5) :
Error: Invalid number input. Please enter a number between 1 and 5.

Are you interested in creating visual designs, animations, AR, VR or working with game mechanics? (1-5) : 3
```

Figure 6-6 Specialization Predictor

```
Specialization suggestion: Cloud Computing and Virtualisation Technology
The specialization has been saved as the previous specialization suggestion.

Student Menu:
1. Classify (Get specialization suggestion)
2. Exit
```

Figure 6-7 Specialization Suggestion

The next time when student logs in and checks the recommendation it will show the previous recommendation as well.

The Original data’s visualization:

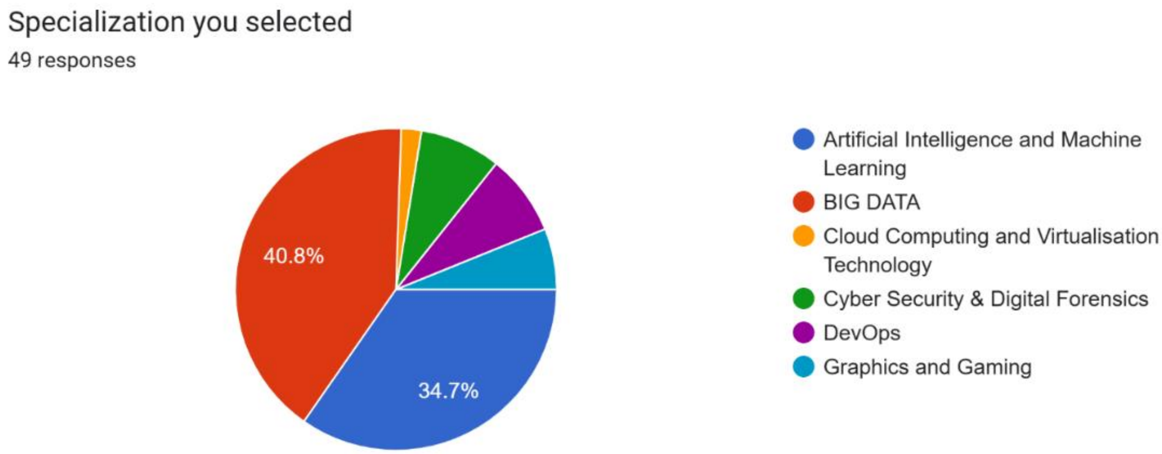

Figure 6-8 Specialization Selected by students

The synthetic data visualization:

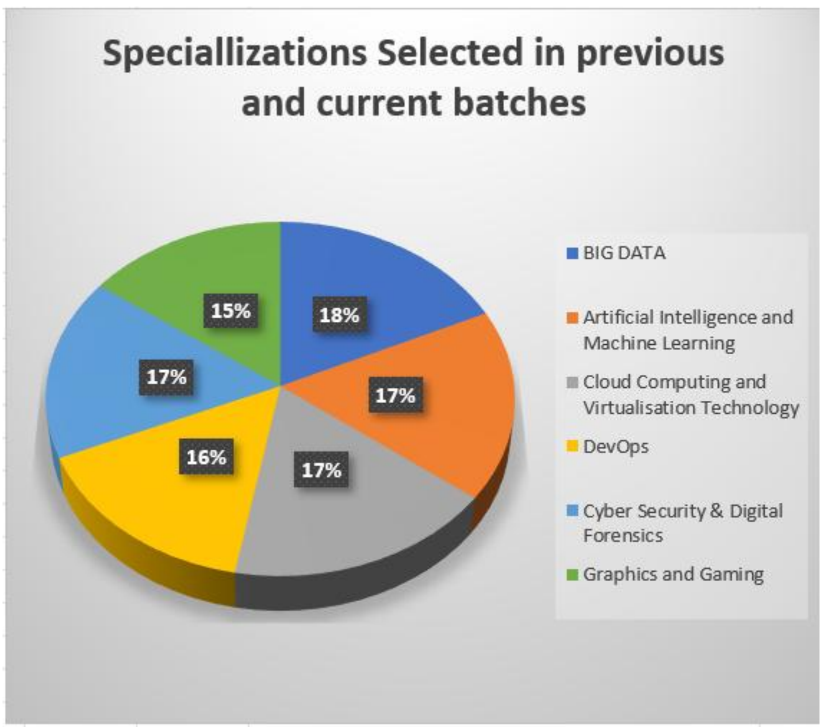

Figure 6-9 Specialization Selected by students in previous and current batches

Admin's Menu

The admin's menu is reserved with particular ID and password.

```
1. Login
2. Create new account
3. Exit
1
Login:
Enter User ID: admin
Enter Password: admin123
Admin login successful.

Admin Menu:
1. Create educator account
2. See/Edit student details
3. See/Edit educator details
4. Delete educator or student
5. Classify (Student specialization suggestion)
6. Cluster (Group students based on performance)
7. Exit
```

Figure 6-10 Admin's Menu

Admin can also delete the student or educator account

```
Admin Menu:
1. Create educator account
2. See/Edit student details
3. See/Edit educator details
4. Delete educator or student
5. Classify (Student specialization suggestion)
6. Cluster (Group students based on performance)
7. Exit
4
Delete user by:
1. User ID
2. Name
3. Exit
Choose an option: ronik
Invalid input. Please enter a number.
Delete user by:
1. User ID
2. Name
3. Exit
Choose an option: 2
Enter Name to delete: ronik
User with ID ronik123 deleted successfully.
```

```
3
Search for educator by:
1. User ID
2. Name
3. Exit
Choose an option: 2
Enter educator Name: ronik
Educator details: {"_id": {"$oid": "6737565c0bf762271248c231"}, "name": "ronik", "email": "ronik@gmail.com", "university": "UPES", "userId": "ronik123", "password": "1234", "r
Search for educator by:
1. User ID
2. Name
3. Exit
```

Figure 6-11 Admin choose option form the menu

```
Admin Menu:
1. Create educator account
2. See/Edit student details
3. See/Edit educator details
4. Delete educator or student
5. Classify (Student specialization suggestion)
6. Cluster (Group students based on performance)
7. Exit
7
Exiting admin menu.
1. Login
2. Create new account
3. Exit
3
Exiting system...
```

Figure 6-12 Admin exits the from the menu

Here the Program ends.

7. LIMITATION AND FUTURE ENHANCEMENTS

Limitations of the System

1. **Dependence on Accurate Data**
    - The system's effectiveness heavily depends on the accuracy and completeness of input data.
    - Inaccurate or incomplete data can result in misleading specialization recommendations and performance groupings.
2. **Complex Implementation**
    - Integrating classification (ID3) and clustering (K-Medoids) algorithms into existing educational systems is technically challenging.
    - This process requires significant expertise, time, and resources, which may delay deployment.
3. **User Resistance**
    - Educators and administrators may resist adopting the system due to their familiarity with traditional academic management methods.

    - Resistance from key stakeholders could limit the system’s full utilization.
4. **Limited Scope for Personalization**
    - Although the system aims to provide personalized recommendations, it may not capture all unique student circumstances (e.g., personal challenges, extracurricular interests).
    - This limitation could affect its effectiveness for certain users.
5. **Data Privacy Concerns**
    - Handling sensitive student information poses risks related to privacy and data security.
    - Compliance with data protection regulations and ensuring user trust are critical challenges.

Future Enhancements

1. **Enhanced User Interface**
    - Develop a more intuitive and user-friendly interface.
    - Simplify navigation to improve the overall user experience for both students and educators.
2. **Integration of Machine Learning**
    - Implement advanced machine learning techniques to enhance the accuracy of specialization recommendations and clustering outcomes.
    - Enable continuous learning from user interactions to refine system performance over time.
3. **Mobile Application Development**
    - Create a mobile version of the system for increased accessibility.
    - Allow students and educators to interact with the platform on-the-go, ensuring convenience and flexibility.
4. **Feedback Mechanism**
    - Introduce a system where users can provide feedback on recommendations.
    - Use the feedback to refine algorithms and improve the system’s accuracy and relevance.
5. **Broader Data Sources**
    - Incorporate additional data sources, such as industry trends, alumni success rates, and job market analyses.
    - Enhance specialization recommendations to align with current and future job market demands.
6. **Collaboration Features**
    - Add functionalities for students to collaborate on projects or form study groups based on clustering results.
    - Foster peer support and improve learning outcomes through enhanced collaboration.
7. **Regular Updates and Maintenance**
    - Establish a routine for updating algorithms and system features.

- Stay aligned with emerging educational trends and user needs to ensure continued system relevance and effectiveness.

CONCLUSION

The "Academic Monitoring & Specialization Selection System" project represents a significant advancement in educational support through its dual-purpose functionality for students and educators. By leveraging advanced algorithms for specialization recommendations and performance-based grouping, the system aims to enhance student guidance and streamline academic management. The user-friendly interface facilitates easy navigation for students and educators alike, ensuring that personalized recommendations and performance insights are readily accessible. Despite its strengths, such as data-driven decision-making and enhanced educator support, the project faces limitations, including reliance on accurate data input and potential resistance from stakeholders accustomed to traditional methods. Future enhancements could focus on improving the user interface, integrating machine learning for better recommendations, and expanding functionalities to include mobile access and collaborative features. Overall, this system has the potential to significantly improve educational outcomes by providing tailored support to students and enabling educators to effectively manage diverse classroom needs. As educational institutions increasingly prioritize personalized learning experiences, the implementation of this system could lead to more informed decision-making, better resource allocation, and ultimately, greater student success.

REFERENCES

[1] Singh, Samrat, and Vikesh Kumar. "Performance analysis of engineering students for recruitment using classification data mining techniques." *International Journal of Science, Engineering and Computer Technology* 3, no. 2 (2013): 31.

[2] Le Quy, Tai, Gunnar Friege, and Eirini Ntoutsi. "A review of clustering models in educational data science toward fairness-aware learning." *Educational data science: Essentials, approaches, and tendencies: Proactive education based on empirical big data evidence* (2023): 43-94.

[3] Bobâlcă, Claudia, Oana Ţugulea, and Cosmina Bradu. "How are the students selecting their bachelor specialization? A qualitative approach." *Procedia economics and Finance* 15 (2014): 894-902.

[4] Kurniawan, Tri Basuki, and Indah Hidayanti. "Classification Algorithms to Determine Students' Specialization in a Higher Education Institution." *Journal of Data Science* 2023 (2023).

Appendix A: Glossary

Academic Monitoring: The process of tracking and analysing students' academic performance to provide insights and support for their educational journey.

Clustering: A data mining technique that groups a set of objects in such a way that objects in the same group are more similar to each other than to those in other groups. Used for identifying patterns in student performance.

Classification: A machine learning approach that assigns labels to data points based on their attributes, aiming to predict the categorical outcome of a data point.

Decision Tree: A flowchart-like structure used in machine learning for making decisions based on the values of input features, commonly used for classification tasks.

Database Management System (DBMS): Software that interacts with the user, applications, and the database itself to capture and analyse data. In this project, MongoDB is utilized for storing classification data.

MongoDB: A NoSQL database that stores data in a flexible, JSON-like format, allowing for scalability and efficient data retrieval.

CSV (Comma-Separated Values): A simple file format used to store tabular data in plain text, where each line represents a data record and each record consists of fields separated by commas. Used in this project for clustering data storage.

User Authentication: The process of verifying the identity of a user attempting to access the system, ensuring that only authorized individuals can use the application's functionalities.

Menu-Driven Program: An application interface that allows users to interact with the system by selecting options from a list or menu, facilitating easy navigation.

Role-Based Access Control (RBAC): A security approach that restricts system access to authorized users based on their roles, ensuring that users can only perform actions permitted for their role.

Data Protection Regulations: Legal requirements that govern the processing of personal data, ensuring individuals' privacy and rights. Examples include the General Data Protection Regulation (GDPR).

Usability: A measure of how easy and efficient it is for users to interact with the system, emphasizing user satisfaction and the effectiveness of the interface.

Robustness: The ability of the system to handle errors and unexpected inputs gracefully without crashing or producing incorrect results.

Performance Benchmarking: The process of measuring the performance of the system under specific conditions to ensure it meets the required performance standards.

Interoperability: The capability of the system to integrate and work with other external systems or services, enhancing its functionality and data exchange.

Appendix B: Analysis Model

- **Use Case Diagrams**:

  Show interactions between users (administrators and students) and the system.

  Highlight functionalities like viewing recommendations, inputting student data, and generating clustering results.

  Define functional requirements based on user needs.

- **Class Diagrams**:

  Illustrate the structure of the system with different classes, attributes, and methods.

  Show relationships among classes (associations, generalizations, dependencies).

  Serve as a blueprint for future code implementation.

- **Data Flow Diagrams (DFDs)**:

  Visualize how data moves through the system.

  Outline inputs, outputs, data stores, and processes.

  Present in multiple levels:

  Level 0: Context diagram showing the system as a whole.

  Level 1: Breakdown of major processes like data input, classification, and clustering.

- **Activity Diagrams**:

  Represent the sequence of activities or flow of control during operations.

  Show steps involved in processes, such as generating a specialization recommendation based on student data.

  Help visualize how key functions are performed.

- **State Diagrams**:

Depict the different states an object can be in and transitions between these states.

Outline the lifecycle of important objects, like a student's status changing from "Registered" to "Recommended."

Provide a comprehensive view of object behaviour throughout the system.

- **Sequence Diagrams**:

Clarify interactions within the system by showing the order of operations during specific use cases.

Illustrate how different components collaborate to complete tasks, such as generating recommendations.

Ensure correct sequencing of interactions between objects.

- **Entity-Relationship (ER) Diagram**:

Represents the data model for the system, showing the entities involved and their relationships.

Key entities may include:

Student: Contains attributes like Name, SAP ID, GPA, and Interests.

Specialization: Contains details about available specializations and requirements.

Performance: Stores performance metrics such as test scores and interest levels.

Clustering: Represents clusters formed based on academic performance metrics.

Shows relationships between entities, such as:

A **Student** can have multiple **Performance** records.

A **Specialization** can be recommended to multiple **Students** based on their profiles.

Appendix C: Issues List

The following outlines the current issues identified:

Data Privacy Concerns: There are ongoing discussions regarding compliance with local data protection regulations, particularly concerning the handling and storage of sensitive student information. Further analysis is required to align the system's practices with these regulations.

Integration Challenges: There may be potential integration issues with external educational platforms or APIs that could affect data accuracy and interoperability. A detailed assessment of required APIs and their functionalities is pending.

User Access Control: The role-based access control implementation requires further clarification to ensure that user permissions are appropriately assigned and managed. A comprehensive review of user roles and permissions is needed.

Testing Strategy: A detailed testing strategy that encompasses unit testing, integration testing, and user acceptance testing is yet to be defined. This strategy is crucial for ensuring the system's reliability and usability.