**Luxor University**

**Faculty of Computers & Information**

**ــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــ**

**STUDENT PERFORMANCE PREDICTION**

Software Development and Professional Practice

Final Project

**UNDER SUPERVISION**

Dr. Bassem Abdelatty Eldeeb

Eng. Abdelrahim Alsadiq

**SUBMITTED BY**

* Abdelrahman Awadallah Elneel
* Mahmoud Elrayes
* Salma Ibrahim Nasri
* Youssef Gamal
* Abdelrahman Hussien

Contents

[CHAPTER 1 SYSTEM OVERVIEW 1](#_Toc198422772)

[1.1 MOTIVATION 2](#_Toc198422773)

[1.2 PROBLEM STATEMENT 2](#_Toc198422774)

[1.3 OBJECTIVES AND SCOPE 2](#_Toc198422775)

[1.4 TARGET AUDIENCE AND EXPECTED 4](#_Toc198422776)

[CHAPTER 2 RELATED WORKS 5](#_Toc198422777)

[2.1 EXISTING SYSTEMS 6](#_Toc198422778)

[2.2 COMPARISON & LIMITATIONS 7](#_Toc198422779)

[2.3 JUSTIFICATION FOR THE PROPOSED SYSTEM 7](#_Toc198422780)

[CHAPTER 3 SYSTEM REQUIREMENTS & TECHNIQUE 8](#_Toc198422781)

[3.1 Functional Requirements 9](#_Toc198422782)

[3.2 Non-Functional Requirements 10](#_Toc198422783)

[3.2 Techniques and Tools 11](#_Toc198422784)

[CHAPTER 4 PROPOSED SYSTEM & METHODOLOGY 13](#_Toc198422785)

[4.1 OVERVIEW 14](#_Toc198422786)

[4.2 SYSTEM USE CASE DIAGRAM 14](#_Toc198422787)

[4.3 USE CASE DESCRIPTION (USE CASE SCENARIO) 16](#_Toc198422788)

[CHAPTER 5 Activity Diagram 19](#_Toc198422789)

[DATA FLOW DIAGRAM 21](#_Toc198422790)

[CONTEXT DIAGRAM 22](#_Toc198422791)

[Logical data flow 23](#_Toc198422792)

[logical data flow 24](#_Toc198422793)

[CHAPTER 6 System architecture 25](#_Toc198422794)

[Entity Relationship Diagram 27](#_Toc198422795)

# CHAPTER 1 SYSTEM OVERVIEW

## 1.1 MOTIVATION

Student academic performance prediction is a critical concern for educational institutions. Many students face academic challenges due to various factors including study habits, sleep patterns, extracurricular activities, social interactions, physical activity levels, and stress management. Traditional academic monitoring systems often identify struggling students too late, after they've already failed courses or fallen significantly behind.

Our Student Performance Prediction System offers a proactive approach to addressing these challenges. By analyzing key lifestyle and behavioral factors, this system helps universities identify at-risk students early, enabling timely interventions and personalized support strategies. This approach not only improves student retention and academic success rates but also enhances the overall educational experience.

The system empowers academic advisors, professors, and student affairs departments to make data-driven decisions regarding resource allocation and targeted support programs. By understanding the complex interplay of factors affecting academic performance, institutions can develop more effective strategies to foster student success.

## 1.2 PROBLEM STATEMENT

University students juggle multiple responsibilities and lifestyle factors that significantly impact their academic performance. Traditional assessment methods primarily focus on grades and attendance, which don't always capture the underlying factors influencing student success or failure.

Factors such as study hours, sleep patterns, extracurricular involvement, social activities, physical exercise, and stress levels all play crucial roles in determining academic outcomes, yet these are rarely monitored or analyzed systematically. Without understanding these factors, universities struggle to provide timely and appropriate support to students who need it most.

This project aims to develop a prediction model that analyzes these various lifestyle factors to forecast student performance, specifically GPA. By leveraging machine learning techniques, our system identifies students who may be at risk of academic underperformance before their grades suffer, enabling proactive intervention strategies.

## 1.3 OBJECTIVES AND SCOPE

**Project Objectives**

• Early Identification of At-Risk Students: Develop a predictive system that analyzes student lifestyle data—including study habits, sleep patterns, extracurricular involvement, social activities, physical exercise, and stress levels—to identify students who may struggle academically before their performance deteriorates.

• Provide Actionable Recommendations: Generate specific, practical recommendations for students based on their lifestyle patterns to help improve their academic performance.

• Create Role-Based Access: Implement a secure, role-based system that allows administrators, professors, and students to access appropriate levels of information and functionality.

• Deploy a User-Friendly Web Interface: Create an intuitive Flask-based web application that makes the prediction system accessible to all stakeholders.

• Ensure Data Privacy and Security: Implement robust data protection measures to safeguard sensitive student information in compliance with educational data regulations.

Project Scope Included Features • Comprehensive Data Analysis: Utilize student lifestyle data encompassing study hours, sleep patterns, extracurricular activities, social interactions, physical activity, and stress levels to provide a holistic view of factors affecting performance.

• Predictive Modeling: Implement machine learning models to predict student GPA based on lifestyle factors and generate a performance level classification (Excellent, Good, Average, Below Average).

• Personalized Recommendations: Provide tailored suggestions for students based on their specific lifestyle patterns and predicted performance.

• Admin Dashboard: Create a comprehensive interface for administrators to manage users, students, and view prediction history.

• User Management System: Implement a role-based access control system distinguishing between administrators, professors, and students.

• Prediction History: Maintain a searchable and filterable history of past predictions for analysis and follow-up.

Excluded Features (Out of Scope) • Real-Time Monitoring: The system will not continuously track student activities but rather perform analysis based on data input at specific points in time.

• Automated Intervention Implementation: While the system provides recommendations, it does not automatically implement intervention strategies.

• Integration with Learning Management Systems: The current scope does not include direct integration with existing LMS platforms.

## 1.4 TARGET AUDIENCE AND EXPECTED

IMPACT Target Audience • Students: To gain insights into how their lifestyle choices impact academic performance and receive personalized recommendations for improvement.

**• Academic Advisors and Professors:**

To identify at-risk students early and provide targeted support and guidance.

**• University Administrators:**

To make data-driven decisions about resource allocation and support program development.

**• Student:**

To make the student aware about his performance and take early steps to fix his life style

**Expected Impact**

• Improved Academic Performance: By helping students understand the relationship between lifestyle choices and academic outcomes.

• Reduced Dropout Rates: Through early identification of at-risk students and timely intervention.

• Enhanced Student Well-being: By promoting balanced lifestyle choices that support both academic success and overall wellness.

• Data-Driven Decision Making: Enabling universities to allocate resources more effectively based on identified patterns and needs.

• Increased Student Satisfaction: Through personalized support and guidance tailored to individual needs and circumstances.

# CHAPTER 2 RELATED WORKS

## 2.1 EXISTING SYSTEMS

* **model1**:

This study proposes a model that utilizes machine learning algorithms to predict undergraduate students' final exam grades based on their midterm exam results.

The researchers employed various algorithms, including Random Forests, Nearest Neighbor, Support Vector Machines, Logistic Regression, Naive Bayes, and k-Nearest Neighbor.

The model achieved a classification accuracy ranging between 50% and 55%, indicating that midterm grades are a significant predictor of final exam performance.

Notably, the Random Forest, Nearest Neighbor, and Support Vector Machines algorithms demonstrated the highest accuracy, while the k-Nearest Neighbor algorithm had the lowest.

The study underscores the potential of machine learning in early identification of students who may need additional support, allowing for timely interventions to enhance academic success.

* **Authors(model2)**: Yannick Meier, Jie Xu, Onur Atan, Mihaela van der Schaar

Summary: This study aimed to predict students' final grades early in a course to enable timely interventions.

Utilizing data from approximately 700 undergraduate students over seven years, the proposed algorithm achieved a prediction accuracy of 76% for 85% of the students by the fourth week of the course

* **model3**:

Predicting School Children Academic Performance Using Machine Learning Techniques": Conducted by Radwan Qasrawi and colleagues, this study applied machine learning algorithms to predict academic performance among middle and high school students in the West Bank.

The research found that logistic regression and Naive Bayes models achieved the highest accuracy levels (94.3% and 94%, respectively) in classifying and predicting student academic performance

## 2.2 COMPARISON & LIMITATIONS

* **model1** :

A larger dataset can enhance model training by providing more diverse examples, potentially improving generalizability and robustness. It might also allow your model to capture a wider range of variability in student performance, leading to more reliable predictions.

* **model2**:

A larger dataset can enhance model training by providing more examples, It might also allow the model to capture a wider range of variability in student performance, leading to more reliable predictions.

* **model3**:

(مقاطعة) is an Israeli model

## 2.3 JUSTIFICATION FOR THE PROPOSED SYSTEM

Our proposed system is designed to overcome these challenges by addressing several key issues:

* **Integrating Multiple Data Sources**:

We are not limiting ourselves to academic records alone. By combining academic performance data with social engagement metrics and psychological factors, our system aims to capture a more complete picture of a student’s performance.

* **User-Friendly Interface:**

Finally, we provide an interactive dashboard. This tool not only displays up-to-date predictions but also allows educators to quickly identify students who may need early intervention, thereby supporting timely and effective educational strategies.

# 

# 

# CHAPTER 3 SYSTEM REQUIREMENTS & TECHNIQUE

## 3.1 Functional Requirements

**1. User Authentication and Authorization**

* **Requirement ID**: FR-01
* **Description**:
  + The system shall support role-based authentication with three distinct user roles: admin, professor, and student.
  + The system shall authenticate users through email address and password credentials.
  + The system shall enforce appropriate access restrictions based on user roles.
  + In case of authentication failure, the system shall provide appropriate error messages.

**2. Student Performance Prediction**

* **Requirement ID**: FR-02
* **Description**:
  + The system shall predict student GPA based on input variables including study hours, sleep hours, social hours, extracurricular hours, physical activity hours, and stress level.
  + The system shall use a machine learning model to generate predictions with appropriate accuracy.
  + The system shall present prediction results in a clear, understandable format showing predicted GPA and performance level.
  + The system shall generate personalized recommendations based on the input data and prediction results.

**3. User Management**

* **Requirement ID**: FR-03
* **Description**:
  + The system shall allow administrators to create, edit, and delete user accounts.
  + The system shall enable administrators to assign appropriate roles to users (admin, professor, student).
  + The system shall maintain relationships between user accounts and student records for students.
  + The system shall prevent the deletion of the last admin account to ensure system accessibility.

**4. Student Management**

* **Requirement ID**: FR-04
* **Description**:
  + The system shall allow administrators to create, edit, and delete student records.
  + The system shall maintain student information including student ID, name, and major.
  + The system shall automatically create student records when student user accounts are created.
  + The system shall handle the relationship between student records and their prediction history.

**5. Prediction History and Reporting**

* **Requirement ID**: FR-05
* **Description**:
  + The system shall maintain a history of all predictions made for each student.
  + The system shall provide filtering capabilities for prediction history by performance level, date range, and student ID.
  + The system shall allow professors and administrators to view prediction histories across all students.
  + The system shall limit students to viewing only their own prediction history.

## 3.2 Non-Functional Requirements

**User Authentication and Authorization**

* **NFR-01.1**: The system shall process authentication requests and return a response within 2 seconds under normal network conditions.
* **NFR-01.2**: The system shall secure all password data using appropriate hashing algorithms (e.g., Werkzeug's generate\_password\_hash).
* **NFR-01.3**: The system shall implement role-based access control to ensure users can only access appropriate functionality.

**Student Performance Prediction**

* **NFR-02.1**: The prediction model shall generate results within 3 seconds of form submission.
* **NFR-02.2**: The prediction algorithm shall return a GPA score between 0 and 4.0 with appropriate precision.
* **NFR-02.3**: The system shall generate relevant recommendations based on user input that are actionable and helpful.

**Web Interface**

* **NFR-03.1**: The web interface shall be responsive and compatible with major browsers.
* **NFR-03.2**: The system shall display appropriate flash messages for success and error conditions.
* **NFR-03.3**: The interface shall implement appropriate form validation to prevent invalid data entry.
* **NFR-03.4**: The system shall incorporate Bootstrap styling for a consistent, professional appearance.

**Data Management**

* **NFR-04.1**: The system shall maintain referential integrity when deleting records (e.g., deleting predictions when students are deleted).
* **NFR-04.2**: The system shall use appropriate database relationships to maintain data consistency.
* **NFR-04.3**: The system shall implement database transactions to prevent partial operations.

**Security**

* **NFR-05.1**: The system shall implement login\_required decorators to prevent unauthorized access to protected routes.
* **NFR-05.2**: The system shall use Flask's session management for maintaining secure user sessions.
* **NFR-05.3**: The system shall implement proper input validation to prevent common security vulnerabilities.

## 3.2 Techniques and Tools

**Tools**

**Programming Languages & Frameworks**

* **Python**: Core programming language for backend development.
* **Flask**: Web framework for the application backend.
* **SQLAlchemy**: ORM for database interactions.

**Machine Learning**

* **xgboost** or custom ML implementation for student performance prediction.

**Techniques**

**Software Development Methodologies**

* **Model-View-Controller (MVC)**: Architectural pattern separating data model, user interface, and control logic.

**Security Practices**

* **Password Hashing**: Using secure password storage techniques.
* **Role-Based Access Control**: Restricting functionality based on user roles.
* **Form Validation**: Preventing invalid or malicious input.

**Database Design**

* **Relational Database**: Using relationships between Users, Students, and Predictions tables.
* **Transactions**: Ensuring data consistency during complex operations.

**User Experience**

* **Flash Messages**: Providing immediate feedback on user actions.
* **Responsive Design**: Ensuring usability across different devices.
* **Modal Dialogs**: For confirmation of important actions.

# CHAPTER 4 PROPOSED SYSTEM & METHODOLOGY

## 4.1 OVERVIEW

This chapter outlines the **Student Performance Prediction System**, detailing its design, structure, and functionality. The system leverages machine learning to predict students at risk of poor academic performance, assisting educators in early intervention.

**System Overview**

* Predicts student performance using academic, behavioral, psychological, and socioeconomic factors.
* Utilizes machine learning models such as **Random Forest, XGBoost, and Neural Networks**.

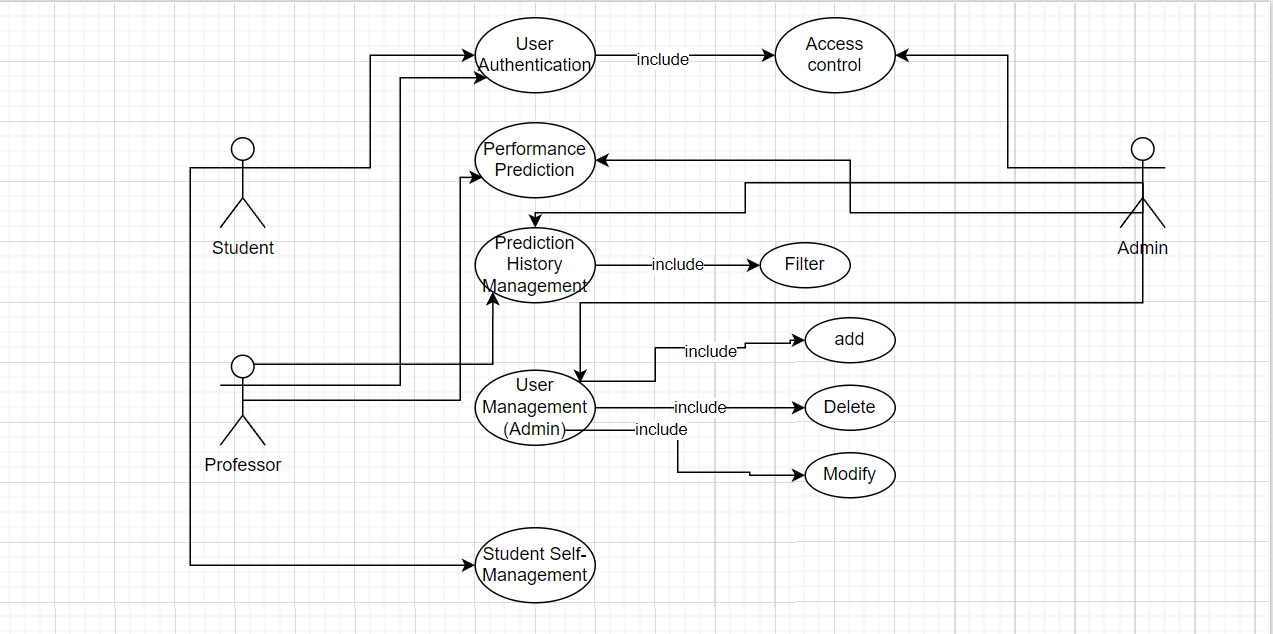
**System Architecture**

* **Data Collection:** Gathers academic records, attendance, and survey data.
* **Preprocessing:** Handles missing values, feature selection, and normalization.
* **Model Training & Evaluation:** Uses an **80-20 split**, hyperparameter tuning, and evaluation metrics (R-Squared).
* **User Interface:** Provides dashboards and early warning alerts for educators.

**System Functionalities**

* **Performance Prediction Dashboard**
* **Early Warning Alerts**
* **Recommendation System for academic support**

## 4.2 SYSTEM USE CASE DIAGRAM



The **Use Case Diagram** illustrates the interactions between different users (actors) and the **Student Performance Prediction System**. The key components are:

**Actors:**

1. **Academic Advisor** – Uses the system to analyze student performance and generate reports.
2. **Professor** – Uses the system to predict student performance and update student data.
3. **University Administrator** – Fetches and updates student data in the system.
4. **Student Performance Prediction System** – The central system that connects all functionalities.
5. **Student predict his own GPA**.

**Use Cases & Relationships:**

1. **Generate Performance Reports** (Academic Advisor)
   * Extends **Export Report to PDF**, allowing the advisor to save reports.
2. **Analyze Student Performance** (Academic Advisor)
   * Depends on **Predict Student Performance**, as performance prediction is required for analysis.
3. **Predict Student Performance** (Professor)
   * Includes **Fetch Student Data** and **Update Student Data** to ensure the latest student data is used.
4. **Update Student Data** (Professor & University Administrator)
   * Includes **Fetch Student Data** to retrieve necessary records before updating.
5. **Fetch Student Data** (University Administrator)
   * Used as a foundational step for other use cases.

## 4.3 USE CASE DESCRIPTION (USE CASE SCENARIO)

|  |  |
| --- | --- |
| ID | UCS\_1 |
| Actors | - Primary: User (Student/Professor/Admin) - Secondary: Authentication System |
| Preconditions | - User has registered account  - System is operational |
| Main Flow | 1. User accesses login page 2. System displays credentials form  3. User enters username/password  4. System validates credentials  5. System grants access to authorized resources |

|  |  |
| --- | --- |
| ID | UCS\_2 |
| Actors | - Primary: Authenticated User  - Secondary: Authorization System |
| Preconditions | - User is successfully authenticated  - Role permissions are configured |
| Main Flow | 1. User requests resource  2. System checks user permissions  3. System verifies access rights  4. System grants/denies access  5. System logs access attempt |

|  |  |
| --- | --- |
| ID | UCS\_4 |
| Actors | - Primary: Admin  - Secondary: User Database |
| Preconditions | - Admin has elevated privileges  - User accounts exist |
| Main Flow | 1. Admin accesses user management  2. System displays user list  3. Admin selects user  4. Admin chooses delete action  5. System confirms deletion  6. System removes user record |

|  |  |
| --- | --- |
| ID | UCS\_3 |
| Actors | - Primary: Admin  - Secondary: Reporting System |
| Preconditions | - Admin is authenticated  - Prediction history exists |
| Main Flow | 1. Admin accesses history module  2. System displays all records  3. Admin applies filters  4. System returns filtered results  5. Admin exports/views data |

|  |  |
| --- | --- |
| ID | UCS\_6 |
| Actors | - Primary: Professor  - Secondary: Prediction System |
| Preconditions | - Professor is authenticated  - Student data exists  - Model is trained |
| Main Flow | 1. Professor selects prediction module  2. System displays input form  3. Professor enters student data  4. System processes request  5. System displays predictions  6. Professor reviews results |

|  |  |
| --- | --- |
| ID | UCS\_5 |
| Actors | - Primary: Student  - Secondary: Profile System |
| Preconditions | - Student is authenticated  - Profile exists |
| Main Flow | 1. Student accesses profile  2. System displays current info  3. Student edits fields  4. System validates input  5. Student submits changes  6. System add new records |

# CHAPTER 5 Activity Diagram

A diagram of a flowchart

AI-generated content may be incorrect.

# DATA FLOW DIAGRAM

# CONTEXT DIAGRAM

A diagram of a software system

AI-generated content may be incorrect.

# Logical data flow

A diagram of a data flow

AI-generated content may be incorrect.

# logical data flow

A diagram of a software company

AI-generated content may be incorrect.

# CHAPTER 6 System architecture

A diagram of a software project

AI-generated content may be incorrect.

# Entity Relationship Diagram

**ERD**

A diagram of a company

AI-generated content may be incorrect.