

ITU Student Record Management System

Ömer Faruk Zeybek

Department of Artificial Intelligence and Data

Istanbul Technical University

Istanbul, Turkey

Email: zeybeko22@itu.edu.tr

June 10, 2024

Abstract—ITU Student Record Management System is a console-based application created to efficiently handle student records within Istanbul Technical University. The system offers functionalities including adding, viewing, searching, updating, and deleting student records. Additionally, it offers statistical analysis and visualization tools for a deeper understanding of the data.

This project utilizes various Python libraries such as Pandas, Seaborn, Matplotlib, CSV, Faker, and Random to data processing, analysis, and visualization tasks. The system’s capabilities extend to generating fake data for testing purposes, ensuring robustness and reliability.

The core features of the system include the ability to add new student records with validation checks, view existing records sorted by different criteria, search for specific students based on various parameters, update records as needed, and delete records securely. Moreover, the system offers statistical analysis tools to derive meaningful insights from the data, and visualization capabilities to present these insights in a clear and informative manner for the users.

This report provides a detailed overview of the ITU Student Record Management System, its design, implementation, functionalities, and the technologies utilized. It also discusses how to improve this project, future of the project.

Index Terms—python, data analysis, console-based application, python libraries

I. INTRODUCTION

The ITU Student Record Management System is designed to manage student records efficiently. This project involves adding, viewing, searching, updating, deleting student records, and performing statistical analysis and visualization.

II. METHODS

The project utilizes several Python libraries for data processing and visualization, including Pandas, Seaborn, Matplotlib, CSV, Faker, and Random.

A. Libraries

- **Pandas:** Used for data processing and analysis.
- **Seaborn and Matplotlib:** Used for data visualization.
- **CSV:** Used for data storage and manipulation with CSV files.
- **Faker:** Used to generate fake data.
- **Random:** Used for generating random data.

B. Functions

- **Add Student** - Adds a new student record. There are 9 fields in the CSV file:
 - Roll No
 - Name Surname
 - Age
 - Gender
 - Students Id
 - Grade
 - Email
 - Phone
 - Major

Each field must adhere to specific rules. User inputs are validated using validation functions, and the added student data is displayed.

- **View Student** - Displays existing student records. Data is sorted as in the database: by roll number, by gender, by major, by grade, in alphabetical order, by gender in a specific major, and by grades in a specific major.
- **Search Student** - Searches for students based on specific criteria. You can search for students by their name surname, ID, and roll number.
- **Update Student** - Updates an existing student record by entering the student’s ID.

CONTENTS

I	Introduction	1
II	Methods	1
II-A	Libraries	1
II-B	Functions	1
II-C	FakeData.py	2
III	Discussion	2
III-A	Strengths	2
III-B	Challenges Encountered	2
III-C	Future Work	2
III-D	Functionality Testing Results, Data Validation Checks, Data Visualization and Statistical Analysis	2
IV	Conclusion	3
IV-A	Project Achievements	3
V	Recommendations	3

- **Delete Student** - Deletes a student record by entering the student's ID.
- **Visualization** - Visualizes student data. It offers various options and creates a chart based on user selection (although it may be meaningless). The generated chart is saved where your ProjectCode.py file is located.
- **Student Statistics** - Performs statistical analysis on student records. It offers various options and saves the analysis results to a text file.

C. FakeData.py

This Python file generates fake data . It configures the faker library for Turkish and generates fake names, student IDs, and phone numbers. Student IDs, roll numbers, and email addresses are unique. By default, it generates 300 fake data. It can be changed by changing num_students variable

III. DISCUSSION

The ITU Student Record Management System provides a robust and flexible system for managing student records using Python.

A. Strengths

- **Comprehensive Functionality:** The system covers all basic functionalities required for student record management, including adding, viewing, searching, updating, and deleting records.
- **Use of Python Libraries:** Leveraging powerful Python libraries such as Pandas, Seaborn, and Matplotlib enables efficient data processing, analysis, and visualization. These libraries are well-supported and widely used in the data science community.
- **Data Visualization:** The integration of visualization tools allows users to gain insights from student data through charts and graphs, making it easier to identify trends and patterns.
- **Fake Data Generation:** The ability to generate fake data using the Faker library is useful for testing and validating the system without compromising real student information.

B. Challenges Encountered

- **Data Validation:** Ensuring that all user inputs are validated correctly to maintain data integrity posed some challenges. Developing robust validation functions was necessary to handle various edge cases and input errors.
- **Data Visualization:** While creating meaningful visualizations, selecting the appropriate chart types and ensuring they accurately represent the data was challenging.
- **Performance Optimization:** Managing large datasets efficiently required careful consideration of data structures and algorithms.

C. Future Work

Future enhancements to the system could include:

- **Graphical User Interface (GUI):** Developing a GUI to make the system more user-friendly.
- **Advanced Security Measures:** Implementing encryption and user authentication to enhance data security.
- **Scalability Improvements:** Optimizing the system to handle larger datasets efficiently.
- **Comprehensive Documentation:** Providing detailed user guides and support resources.

D. Functionality Testing Results, Data Validation Checks, Data Visualization and Statistical Analysis

TABLE I
FUNCTIONALITY TESTING RESULTS

Functionality	Description	Test Result
Add Student	Adds a new student record	Successful
View Student	Displays student records sorted by criteria	Successful
Search Student	Searches student records by various parameters	Successful
Update Student	Updates existing student record	Successful
Delete Student	Deletes a student record	Successful
Visualization	Generates charts based on student data	Successful
Statistics	Provides statistical analysis of student records	Successful

TABLE II
DATA VALIDATION CHECKS

Field	Validation Check
Roll No	Must be unique and numeric
Name Surname	Must contain only alphabetic characters
Age	Must be an integer between 17 and 30
Gender	Must be either 'Male' or 'Female'
Students Id	Must be unique ,numeric, length should be 9
Grade	Must be a float between 1.0 and 4.0
Email	Must end with @itu.edu.tr
Phone	Must start with 05 and length should be 11
Major	Must be one of the predefined majors

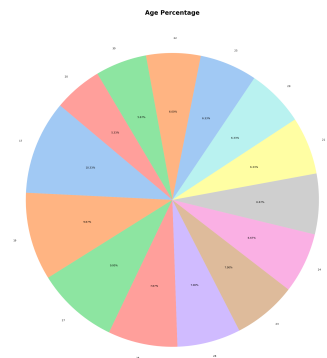


Fig. 1. Data Visualization

Statistical analysis tools were implemented to derive meaningful insights from student data. The following analyses were conducted (These analyses might change due to Faker library because it generates random fake data):

- **Average Grade:** The average grade of students was calculated to be 1.97.
- **Gender Distribution:** The gender distribution was found to be approximately equal, with a slight female predominance.
- **Age Distribution:** Most students were found to be 23 years old.

IV. CONCLUSION

The ITU Student Record Management System is a comprehensive console-based application designed to efficiently manage student records.

A. Project Achievements

- **Data Management:** The system allows users to add, view, search, update, and delete student records, providing a complete suite of functionalities for managing student data effectively.
- **Data Visualization:** With the integration of libraries like Seaborn and Matplotlib, the system offers insightful data visualization capabilities, enabling users to gain valuable insights from student records through various charts and graphs.
- **Statistical Analysis:** The inclusion of statistical analysis features empowers users to perform in-depth analysis on student data, generating valuable statistical insights and trends.
- **Data Integrity:** The system ensures data integrity by implementing validation functions for user inputs, ensuring that all data entered adheres to specified rules and formats.

V. RECOMMENDATIONS

The following recommendations are suggested for further improvement and development of the ITU Student Record Management System:

- **User Interface:**
 - Develop a graphical user interface (GUI) to enhance user experience and accessibility.
 - Ensure the interface is intuitive and easy to navigate to improve usability for non-technical users.
- **Data Analysis and Visualization:**
 - Incorporate more advanced data analysis techniques to provide deeper insights into student records.
 - Develop interactive and dynamic visualizations that allow users to explore data in real-time.
- **Data Security and Privacy:**
 - Implement user authentication and authorization to ensure that only authorized personnel can access sensitive student data.

- Introduce data masking techniques to protect private information (e.g., displaying Student ID as 1*****743).

- **Performance Improvements:**

- Optimize database operations to enhance the performance and responsiveness of the application.
- Implement efficient data processing algorithms to handle larger datasets effectively.

- **Documentation and User Guide:**

- Create a comprehensive user guide that includes detailed instructions on how to use each feature of the system.
- Develop a help section to assist users in resolving common issues.
- Provide documentation for developers.

REFERENCES

- **Pandas Documentation.** (n.d.). https://pandas.pydata.org/docs/user_guide/index.html
- **Seaborn Documentation.** (n.d.). <https://seaborn.pydata.org/tutorial/introduction.html>
- **Matplotlib Documentation.** (n.d.). https://matplotlib.org/stable/users/explain/quick_start.html
- **Faker Documentation.** (n.d.). <https://faker.readthedocs.io/en/master/>

APPENDICES

User Instructions

- **Adding a Student:** Run the script and follow the prompts to enter student details.
- **Viewing Students:** Use the 'View Student' function to display all student records in various sorted orders.
- **Searching Students:** Use the 'Search Student' function to find specific student records by name, ID, or roll number.
- **Updating Students:** Use the 'Update Student' function to modify existing student records.
- **Deleting Students:** Use the 'Delete Student' function to remove a student record from the database.
- **Generating Reports:** Use the 'Visualization' and 'Student Statistics' functions to create visual reports and statistical analyses.

Python Version

This project is developed using Python 3.11.2. It should work with a Python version ,but Python 3.11.2 recommended