GROUP PROJECT REPORT.

Group Members:

1. Abdulfatah Gobu 192181
2. Kakeeto Pius 191777
3. Fatma Omar 193306
4. Victor Mawira 193423

Project Idea: Course Management System.

We designed a Course Management System using Java to streamline university course administration. The system provides functionalities such as adding and managing courses, assigning lecturers, enrolling students, tracking grades, and generating analytics reports. It features a graphical user interface (GUI) for ease of use and integrates with a PostgreSQL database to ensure efficient data storage and retrieval. Additionally, the system includes also has visualizations using JFreeChart for enhanced analytics.

Contributions of each Group Member.

1. Victor Mawira oversaw creating the database schema, implementing the physical database design and inserting some data to the database for testing. He also made contributions towards the GUI pages that required data from the database.
2. Kakeeto Pius was in charge of the java class and interface implementations and created the base classes like Student, Course etc. He was also in charge of creating the classes that connected and retrieved data from the database.
3. Fatma Omar assisted implementing some GUI pages like the login page and also creating the documentation for the project.
4. Abdulfatah Gobu was in charge of the general overview of the project like the UML class diagram and was the main person in charge of the GUI package of the project.

THE LOGICAL DATABASE SCHEMA.

*Relation: Student*

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Constraints | Description |
| stdid | INT | PRIMARY KEY NOT NULL  SERIAL | Used to uniquely identify students in the relation. |
| fname | VARCHAR(50) |  | First name of student |
| lname | VARCHAR(50) |  | Last name of student |
| email | VARCHAR(50) | UNIQUE | The email address of the  student |
| phone | VARCHAR(50) | UNIQUE | The phone number of the  student |
| dob | VARCHAR(50) |  | The date of birth of the student |
| faculty | VARCHAR(50) |  | The faculty the student belongs to |
| gender | CHAR(1) |  | Student’s gender. |

*Relation: Lecturer.*

|  |  |  |  |
| --- | --- | --- | --- |
| **Column**  **Name** | **Data Type** | **Constraints** | **Description** |
| lecid | INT | PRIMARY KEY  NOT NULL  SERIAL | Used to uniquely identify lecturers in the relation. |
| fname | VARCHAR(50) |  | First name of lecturer |
| lname | VARCHAR(50) |  | Last name of lecturer |
| email | VARCHAR(50) | UNIQUE | The email address of the lecturer |
| phone | VARCHAR(50) | UNIQUE | The phone number of the lecturer |
| dob | VARCHAR(50) |  | The date of birth of the lecturer |
| department | VARCHAR(50) |  | The department the lecturer  belongs to |
| gender | CHAR(1) |  | Lecturer’s gender. |

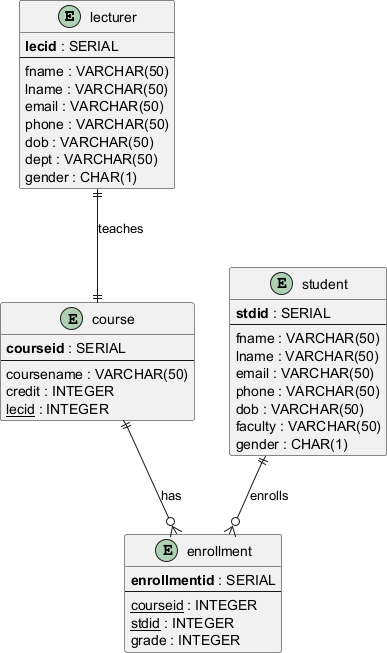
*Relation: Course*

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** | **Description** |
| courseid | INT | PRIMARY KEY  NOT NULL  SERIAL | Used to uniquely identify a course in the relation |
| coursename | VARCHAR(50) |  | The name of the course |
| credit | INT |  | The number of credits awarded  for the course. |

|  |  |  |  |
| --- | --- | --- | --- |
| lecid | INT | FOREIGN  KEY | Used identify the lecturer  assigned to teach the course. |

*Relation: Enrollment.*

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** | **Description** |
| enrollmentid | INT | PRIMARY KEY NOT NULL  SERIAL | Uniquely identifies an enrollment in the relation. |
| courseid | INT | FOREIGN KEY | Identifies the course being  enrolled to |
| stdid | INT | FOREIGN KEY | Identifies the student being  enrolled to the course |
| grade | INT |  | The mark awarded to the  student in the course. |

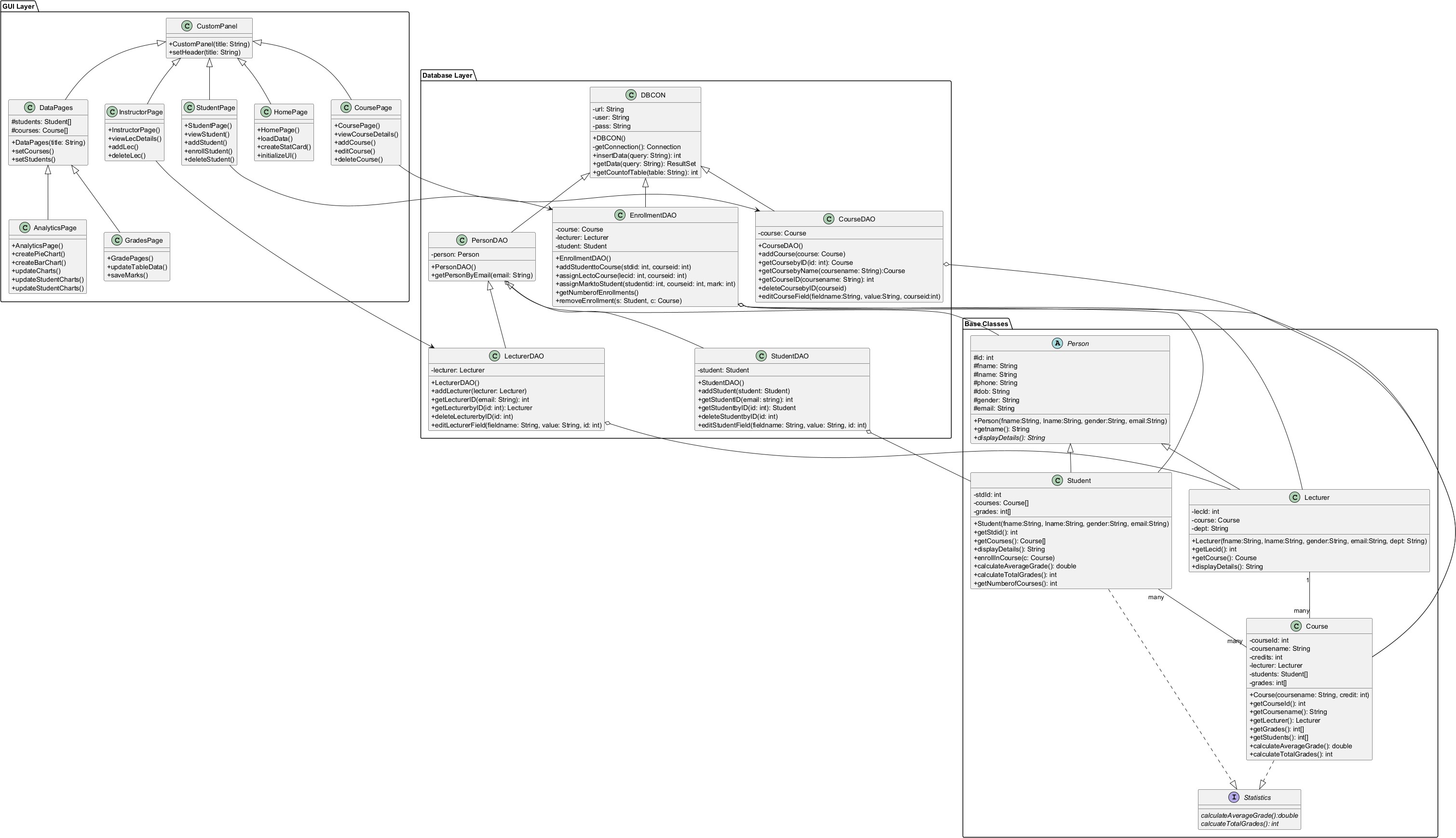


**CONCEPTUAL SCHEMA: ERD**

Challenges Faced and Solutions.

The following are some of the challenges faced during the implementation of the project.

1. We initially faced issues connecting to the database using JDBC but after watching several YouTube videos and reading some Postgres documentation, we were able to figure out how to configure the driver and connect the database to the application.
2. GUI Responsiveness. Another big challenge faced was developing the Graphical User Interface, especially the Gui pages that required retrieve data from the database to display to the user, were really slow. We tried to solve this by researching and using some of the data structures that java provides like Hash Maps, Array Lists, to store the data retrieved to prevent constantly getting data from the database which slightly improved the performance.
3. Collaborating with Git. One of the biggest challenges we faced was collaborating and sharing code using GitHub. The initial way we did it when everyone forked from one’s repository was challenging. We found an easier way where one person created the repository and then used the “invite collaborators” option of GitHub to invite the rest of the group members which automatically gave everyone push access to the main repository.



**UML CLASS DIAGRAM**

**Conclusion**

The development of our Course Management System has been a valuable learning experience, allowing us to apply key programming, database management, and GUI development skills in a real-world scenario. Through teamwork and problem-solving, we successfully designed and implemented a functional system that streamlines university course administration.

Despite facing challenges such as database connectivity issues, GUI responsiveness, and Git collaboration, we overcame these obstacles through research, trial and error, and effective communication. The integration of PostgreSQL for data management, Java for logic implementation, and JFreeChart for data visualization has resulted in a robust and user-friendly system.

This project has enhanced our technical expertise and collaboration skills, preparing us for future software development projects. While our system meets the basic requirements, there is room for future enhancements, such as improved performance optimization, additional features, and a more refined user interface.