**Phase 4 Python Project**

Project summary is not more than 2 pages with brief overview of all scenarios implemented.

Project Summary:

Our project is called the school management system. There are total of six scenarios implemented in our project which are: administration/login, student registration, course registration, fees calculation, input of receipt, administration, and registration table, and viewing the receipt. Administration, and registration table. The seven tables that we included in our project are: administration, course, course\_offered, feeinstallment, receipt, registration, and student. We chose these tables because it relates to our scenarios. For example, the student table relates to the student registration table. We have a main user interface which has 12 options. Each option can either register data into database, view data from database, update data, search data, or even delete data from the tables. The school management system project in python is written in python programming language. The system is designed to handle the system's storage of student information. The system's objective is to keep track of student information on campus in a secure manner. The project was created as a graphical user interface (GUI) programme using the Tkinter module, which displays the interface as a user-friendly interaction. To access the main school management system, you must enter a user login, which is protected by a rigorous encrypted password.

The school management system main user interface includes options for the administrator to manage the student information based on their valuable information. This system can be helpful for school administrators in the job market because they need an application where they can manage student information conveniently and reliably. Our project organizes student and course registration data in a unique and simple way.

Our tables have different data types which include primary keys and foreign keys. The student table consists of: (id INTEGER NOT NULL PRIMARY KEY, name VARCHAR(25), surname VARCHAR(25), email VARCHAR(25), subject VARCHAR(25), gender VARCHAR(25), fee INTEGER). The course table consists of: (course\_id VARCHAR(25) NOT NULL PRIMARY KEY, python INTEGER, java INTEGER, c INTEGER, unity INTEGER, bash INTEGER, php INTEGER). The administration table consists of: (admin\_id INTEGER NOT NULL PRIMARY KEY, initials VARCHAR(25), priority VARCHAR(25), case\_number VARCHAR(25), st\_id INTEGER) with foreign key (st\_id) references student(id). The registration table consists of (registration\_id INTEGER NOT NULL PRIMARY KEY, student\_ids INTEGER, course\_ids VARCHAR(25), year VARCHAR(25), reference VARCHAR(25)) with (student\_ids) references student(id) and (course\_ids) references course(course\_id) as foreign keys. The FEEINSTALLMENT table consists of (fee\_id INTEGER NOT NULL PRIMARY KEY, TOTAL\_FEE INTEGER, REMAIN\_FEE INTEGER, PAID\_FEE INTEGER, INSTALLMENT INTEGER, ad\_id INTEGER) and foreign key (ad\_id) references administration(admin\_id). The receipt table is receipt(receipt\_id INTEGER NOT NULL PRIMARY KEY, reference VARCHAR(25), message VARCHAR(25), year VARCHAR(25), fees\_id INTEGER) with the foreign key (fees\_id) references FEEINSTALLMENT(fee\_id). The course\_offered table is (offer\_id INTEGER NOT NULL PRIMARY KEY, semester VARCHAR(25), course\_code VARCHAR(25), year VARCHAR(25), courses\_ids VARCHAR(25)) with foreign key (courses\_ids) references course(course\_id).

During the course of all the phases, lots of changes were done. In phase 2, the scenarios were too complex and could not be done so we had to delete the parking permit table, teacher table, scholarships table, and honour roll graduation table. In phase 3, we had to remove the view table and add the course\_offered table and registration table. Now, all the scenarios are well defined and the tables are well thought out. Throughout all the phases of the project, everyone worked on the coding aspect of the projects. Each member contributed to each scenario and did some coding for the main code application. Amin did the administrator/login and fees calculation, Hamid did registration and administration scenarios. Salman did the student and course registration. Victor did the fees calculation and

Brief overview of all scenarios and screenshots:

**MAIN UI:** There are 12 options total to register student, register course, input administration/receipt/fees and other tables. There is a new refresh and shutdown button. The refresh table refreshes the system for any changes. The shutdown button shuts the system down completely with saved changes.

A picture containing text

Description automatically generated

**Scenario 1: Administration/Login:**

In this scenario, administration/login information is where the administrator inputs the username and password to login in the school management system. If the administrator forgets the password, there is a password hint to be pressed in order to view the password hint in a window popup. There is also “RECOVER PASSWORD” widget and another window pops up saying to enter email. The administrator will enter their email and password will be emailed to them with a window pop up.

Graphical user interface

Description automatically generated

**Scenario 2: Student Registration:** The student registration is for registering the student in the database system. The student data can also be deleted within the database by pressing delete. If there is incorrect entry in the student registration form, you can press the reset button to reset data. In this scenario, we enter the student id, name, surname, email, gender, course, subject, total fees calculated automatically, and course id for the course. Another window pops up to confirm the registration of the student. If the yes button is pressed, then the information is registered in the database system in the student and course tables.

Graphical user interface

Description automatically generated

**Scenario 3 Course Registration:** In this scenario, we will input the course information based on the course id from the registration of the student from scenario the student registration form. The course id must match from the student registration from the student registration as a foreign key. There is offer id, semester, section of the course, and course id. The window popup confirms that the course information is registered in the database. The purpose of this scenario is to register the student into a specific semester with a specific course code. Course registration information can also be viewed from the main user interface.

Graphical user interface

Description automatically generated

**Scenario 4 Fees Calculation:** In this situation, we input the student id of from the registered student. We input money in the account as paid fees. The remaining fee for the student can be shown because the student paid already from the inputted value from the total fees. The purpose of this scenario is to provide the fees for the courses. Some fees are the same for each course, some fees are different for some of the courses.

Graphical user interface

Description automatically generated

**Scenario 5 Input of receipt, administration, and registration table:**

Purpose of the input for the receipt, administration, and registration widgets is to provide valuable data to the school management system. Administrators require specific identification numbers, specific student ids, registration ids, and other data in order to organize big data.

Graphical user interface

Description automatically generatedAll the input information is going into the tables and in the database. The fees\_id must match the student id. For the administration input, the correct student id from the student table must be used. For the registration window input, course id and student id must be the same from previous data. When data is entered in the child windows, a window popups for confirmation of entry of data to the database.

Graphical user interface

Description automatically generated

Graphical user interface

Description automatically generated

**Scenario 6 View of receipt, administration, and registration table:**

In this scenario, the receipt, administration, and registration from the mysql database is viewed. The purpose of this scenario is to show the administrator who is logged in, to view which data is inputted already within the database without looking through the MySQL Workbench software. Data needs to be viewable within the school management system because it provides a convenient way to display data about inputted information.

All the input information is from the database is viewable in the options of: 6. View Receipt, 9. Administration Details, and 11. Registration View. These view tables are extracting information from the database in a formatted output in tree view functions.

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Text

Description automatically generated