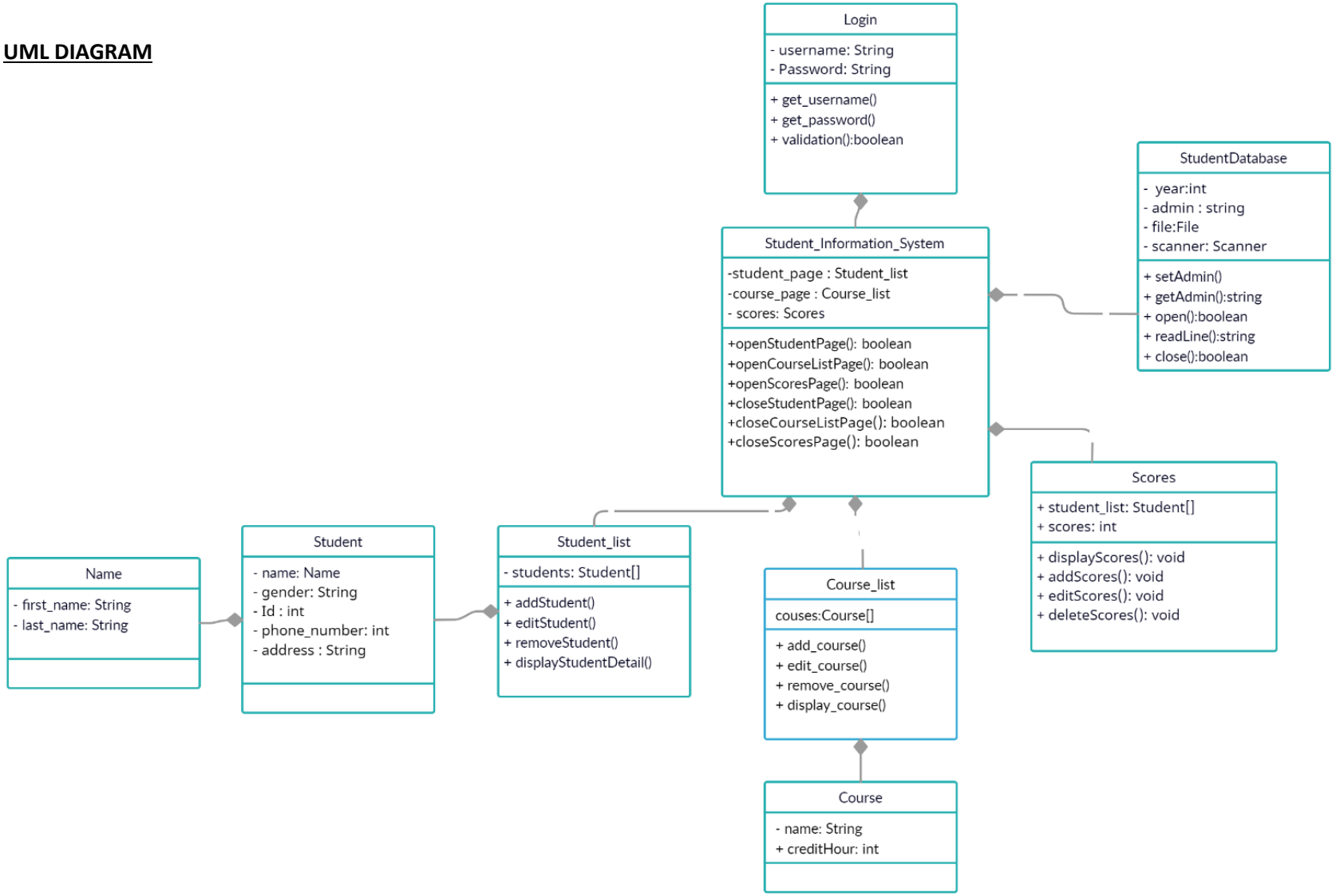
UML DIAGRAM AND NARRATION: STUDENT INFORMATION SYSTEM

MAY SEMESTER of 2021

Name of student:	Fatin Nur Afiqah Binti Ibrahim
Student ID:	20001565
Name of student:	Ong Wei Ling
Student ID:	19001359
Program:	Computer Engineering
Lecturer:	Dr.Mohd Nordin Zakaria

UML DIAGRAM


Login
- username: String
- Password: String
+ get_username()
+ get_password()
+ validation():boolean
StudentDatabase
- year:int
- admin : string
- file:File
- scanner: Scanner
+ setAdmin()
+ getAdmin():string
+ open():boolean
+ readLine():string
+ close():boolean
Student_Information_System
-student_page : Student_list
-course_page : Course_list
- scores: Scores
+openStudentPage(): boolean
+openCourseListPage(): boolean
+openScoresPage(): boolean
+closeStudentPage(): boolean
+closeCourseListPage(): boolean
+closeScoresPage(): boolean
Scores
+ student_list: Student[]
+ scores: int
+ displayScores(): void
+ addScores(): void
+ editScores(): void
+ deleteScores(): void
Name
- first_name: String
- last_name: String
Student
- name: Name
- gender: String
- Id : int
- phone_number: int
- address : String
Student_list
- students: Student[]
+ addStudent()
+ editStudent()
+ removeStudent()
+ displayStudentDetail()
Course_list
couses:Course[]
+ add_course()
+ edit_course()
+ remove_course()
+ display_course()
Course
- name: String
+ creditHour: int

Design

When someone wants to access the Student Information System, the system will pop out a login page first which required username and password from the user. If the username and the password entered correctly, then the user able to access the system. There will be three main parts in this Student Information System which are “Student”, “Course” and “Score”. User able to perform a series of operations and the data entered or edited is saved into a file and stored in a database called StudentDatabase.

Login Class

In this class, there are two attributes which is the username and the password. This class has three operations which is the get_username() to get the the username input from the user, the get_password() function to get the password input from the user and validation() function to validate either the username and the password enter by the user is correct or not. If the user entered the correct username and the correct password then user will have the access to go the next class which is Student_Information_System class to continue with other operation from the class itself.

Student_Information_System

This class has the attributes of student_page which the datatype is Student_list from the class of Student_list. Next, this class also has the attributes of course_page which the datatype is Course_list from the class of Course_list. The last attributes for this class are scores witch the datatype is Score from the class Score. From this class, the user can choose either to open or close Student page by using the openStudentPage() method and closeStudentPage() method respectively. Next, the user can also choose either to open or close course page by using the openCourseListPage() method and closeCourseListPage() method respectively. Next option that user have is either to open or close scores page by using the openScoresPage() method and closeScoresPage() method respectively. Lastly, from all these choice the user can only choose one operation at a time .The user cannot open several pages at the same time.

Student_list class

This class has the attributes of the students which are the array of student from class Student. In this class, the user will have several options to do which are to add a student detail on the system by using the addStudent() method. Then, the user can also edit the student details in the system by using the editStudent() method. Next, the user can also delete the student details he or she wants to delete by using the removeStudent() method. Lastly, the displayStudentDetail() will automatically display all latest student details in the system.

Student Class

This class has variables that contains the details of an applicant. This class is only used to declare the student details attributes which are name, gender, id, phone number and address.

Name Class

This class has variable that contains the full name of a student which are first_name and last_name.

Scores Class

This class consists of 2 attributes and 5 operations. This class was designed for better organisation for the scores of the students. The variable which called as student_list that contains Student object inside it. Then, the attribute of scores represents the scores recorded by the users.

The operations for this class are displayScores(), addScores(), editScores() and deleteScores(). For the function of displayScores() is to display the table of the scores obtained by the students, addScores() is to allows the users to add the students score into the system for records, editScores() is to edit the scores which previously appended.

Course_list class

This class has the attributes of courses which are the list of courses from Course class. In this class, the user will have several options to do which are to add a course on the system by using the add_course() method. Then, the user can also edit the course details in the system by using the edit_course() method. Next, the user can also delete the course details he or she wants to delete by using the remove_course() method. Lastly, the display_course() will automatically display all latest call details in the system.

Course class

This class is used to declare the course detail which are the course name and the credit hour for each course

StudentDatabase Class

This class consists of 4 attributes and 5 operations. The attributes are year, admin, file and scanner. The year represent the year of the record so that it will be more organised. Then, the admin represents the owner of the system could be the staff, and this attribute is set as private which means it could not be access outside from this class. Besides, there is a list of files which can be only accessed by the admin. The scanner attributes which is the Object of Scanner is to scan the file.

The operations contain in this class re setAdmin(), getAdmin(), open(), readLine() and close(). The setAdmin() operation is to set the admin then getAdmin() is to get the information of the admin who wants to access the database. Then the file will open for read and close while finish.

.