Abstract

The fundamental problem that we tried to tackle in this code was how to implement an application for schools with object oritened programming.

UserStory

The first thing that we thought about was every user has what needs and how much accessibility should this user have and this led us to writing a user story that is written in the below section.

As an admin(school's management) , I want be able to add or remove students, teachers, and other employees and include their personal information, so that when need raises we'll be able to add/remove users to/from the system, and in addition use their personal information in an emergency.

As a teacher, I want to be able to manage students attendance for my class so that I can check and inform parents about their students absence, and have a fair judgment toward students according to their attendance to school.

As a teacher, I want to be able to manage grades so that I have a database to evaluate students progress and be able to check the class average and evaluate my own teaching methods.

As a student, I want to be able to see my schedule so I can attend to the right class at the right time.
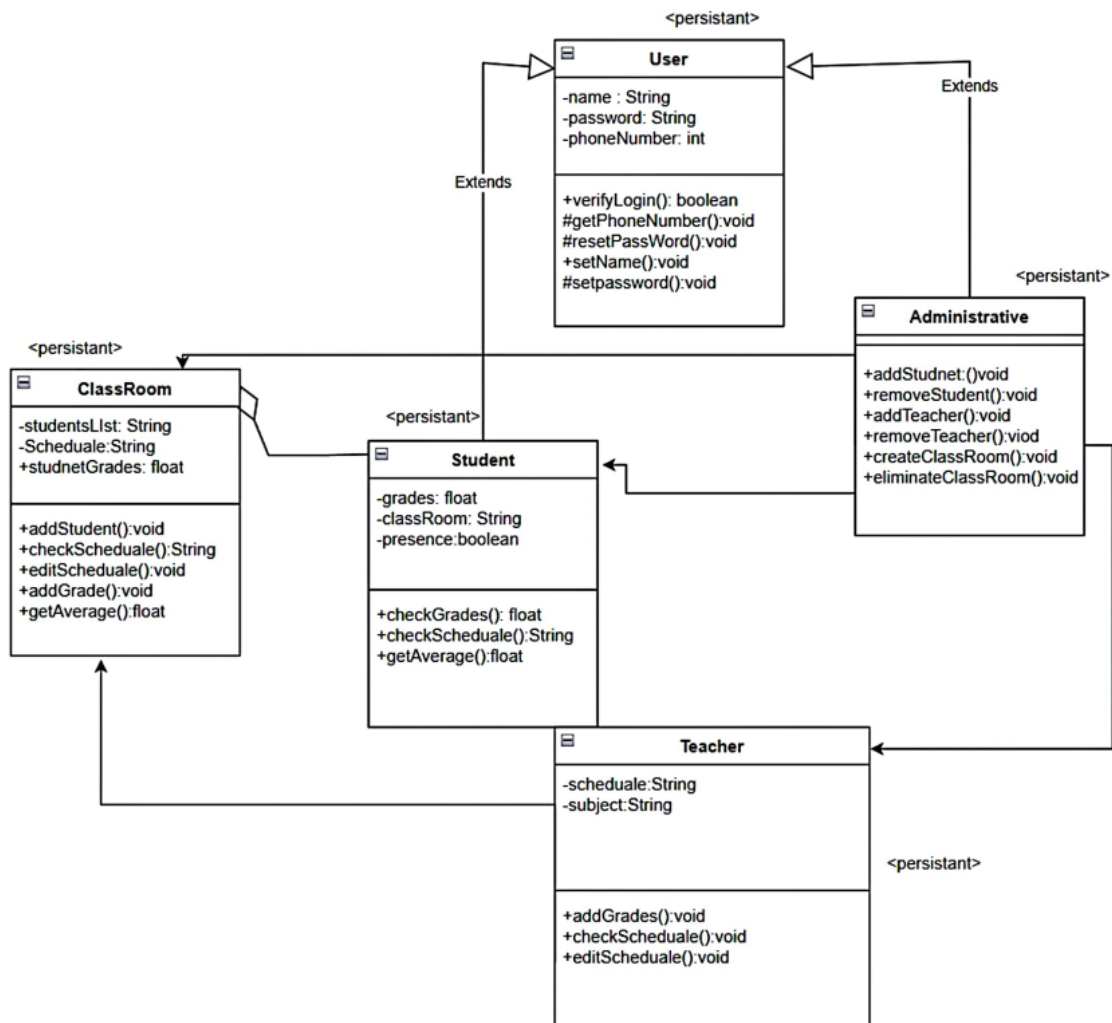
As a student, I want to see my report card so I can track my progress or falling behind.

As a manager, I want to be able to make classes, so that the teachers and students know where their classes are being held, and evaluate the teacher's success in passing on the information to the students.

The system should ensure the security and privacy of student and staff's information, so that only relevant authority can access this information.

The system should have a log in page, so that everyone has access to the methods and attributes that are necessary to them.

Class Diagram

User <persistant>
- -name : String
- -password: String
- -phoneNumber: int

- +verifyLogin(): boolean
- #getPhoneNumber():void
- #resetPassWord():void
- +setName():void
- #setpassword():void

Extends

Extends

Administrative <persistant>
- +addStudnet:()void
- +removeStudent():void
- +addTeacher():void
- +removeTeacher():viod
- +createClassRoom():void
- +eliminateClassRoom():void

ClassRoom <persistant>
- -studentsLIst: String
- -Scheduale:String
- +studnetGrades: float

- +addStudent():void
- +checkScheduale():String
- +editScheduale():void
- +addGrade():void
- +getAverage():float

Student <persistant>
- -grades: float
- -classRoom: String
- -presence:boolean

- +checkGrades(): float
- +checkScheduale():String
- +getAverage():float

Teacher <persistant>
- -scheduale:String
- -subject:String

- +addGrades():void
- +checkScheduale():void
- +editScheduale():void

Sequence Diagram