**Fundamentals of Computing II - LAB**

**CSCE 1102- Project**

**Milestone 1 - Design Report**

**Team#: 2**

**Members’ Info:**

| **Member Name** | **ID** | **Email** |
| --- | --- | --- |
| **Omar Marey** | **900245561** | **omarmarey@aucegypt.edu** |
| **Zeyad Elmoghazy** | **900241661** | **zelmoghazy@aucegypt.edu** |
| **Andrew Khalifa** | **900243415** | **andrew\_khalifa@aucegypt.edu** |

**GitHub link:** [**https://github.com/OMarey1/University-Course-Registration-System.git**](https://github.com/OMarey1/University-Course-Registration-System.git)

**Main classes**

| **Class Name /Short description** | **Data members** | **Member Functions** |
| --- | --- | --- |
| User  A base class for all users | name: string  username: string  password: string  role: enum UserRole | User(...)  getName(): string  getUsername(): string  getPassword(): string |
| Admin  This class is to generate admin objects |  | createUser(...): bool  deleteUser(...): bool  createCourse(...): bool  deleteCourse(): bool  listUsers(): vector<User\*>  saveUsersData(): bool  validatePassword(...): bool |
| Student  This class is to generate student objects | studentID: int  registeredCourses: vector<Course\*> | registerCourse(Course\* course): bool  dropCourse(Course\* course): bool  getCourses(): vector<Course\*> |
| Instructor  This class is to generate instructors objects | instructorID: int  assignedCourses: vector<Course\*> | addCourse(Course\* course):bool  removeCourse(Course\* course):bool  getCourses() vector<Course\*> |
| Course  This class is to generate course objects | courseID: string  courseName: string  creditHours: unsigned int  instructor: Instructor\*  capacity: unsigned int  scheduleTime: string  department: string  enrolledStudents: vector<Student\*>  waitingList: queue<Student\*> | enrollStudent(Student\* student): bool  dropStudent(Student\* student): bool  checkScheduleConflict(string otherSchedule): bool |
| Banner  The management class | users: map<string, User\*>  courses: map<string, Course\*> loggedInUser: User\*  isAdmin: bool | login(...): bool  logout(): void  addUser(User\* newUser): bool  deleteUser(...): bool  listUsers(): vector<User\*>  createCourse(...): bool  deleteCourse(...): bool  searchCourse(...): Course\*  generateReport(): bool  loadFromFile():bool  saveToFile():bool |
| enum UserRole | { ADMIN, INSTRUCTOR, STUDENT } |  |

| **List Name** | **Selected DS** | **Reasons for selecting this DS** |
| --- | --- | --- |
| enrolledStudents | vector | Easy to add more students and iterate over them |
| waitingList | queue | To make sure that students enter the waiting list in FIFO order. |
| registeredCourses | vector | Easy to add more courses and iterate over them |
| assignedCourses | vector | Easy to add more courses and iterate over them |
| users | map | Users are stored using their username as the key for quick search. |
| courses | map | Courses are stored using their courseID as the key for quick search. |