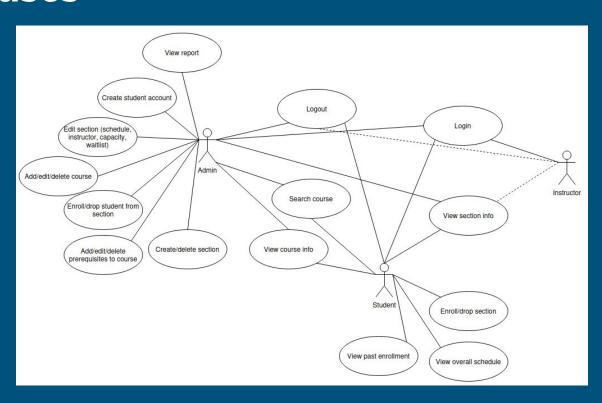
# Phase Two: College Course Enrollment System

By: Roy Alkhoury, Anthony Josh Legrama, Janelle Kwofie, Bach Ngo

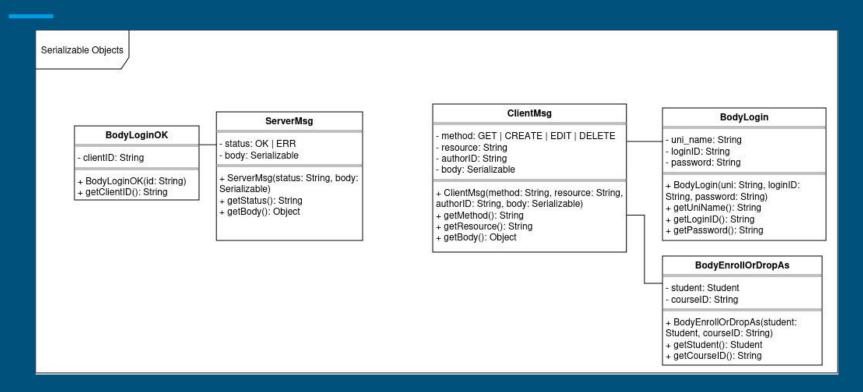
## **Use Cases**



## Serializable Objects

Serializable Objects University - id: Strina name: String - location: String admins: List<Administrator> catalog: Map<String, Course> students: Map<String, Student> instructors: Map<String, Instructor> University(name: String, location: String) getAllCourses(): List<Course> Administrator getCoursesBvFilter(filter: (course): boolean): List<Course> getCourseByID(courseID: String): Course addCourse(course; Course); void account: Account delCourse(courseID: String): void name: String + editCourse(course: Course): void addAdmin(admin: Administrator): void + Administrator(name: String, account: Account) addStudent(student: Student): void + getID(): String + addInstructor(instructor: Instructor): void + getAccount(): Account + getters/setters + getName(); String Course id: String - prefix: String - id: String number: String account: Account description: String name: String 0..n (unique) - past\_enrollments: List<Section> - prerequisites: List<Course> sections: List-Section> enrolling: List<Section> + Course(prefix: String, number: String, + Student(name: String, account: Account) description: String) + enroll(section: Section): void + insertPrereg(course; Course); void + drop(sectionID: String): void + delPrereg(courseID: String); void + getters insertSection(section: Section): void delSection(sectionID; String); void + getters/setters - id: String account: Account | null name: String teaching: List<Section> + Instructor(name: String, account: Account) Section + getID(): String + getAccount(): Account | null id: String <<Enumeration>> + getName(): String - course: Course EnrollStatus number: String - active: boolean + Enrolled Account - max\_capacity: int + Waitlisted - max\_wait: int + Unsuccessful email: String - instructor: Instructor - password: String - enrolled: List<Student> - waitlisted: List<Student> ScheduleEntry + Account(email: String, password: String) - schedule: ScheduleEntry[] + verify(email: String, password: String); boolean location: String + Section(course: Course, number: String, cap: is sync: boolean int, wait: int, instructor: Instructor) day of week: String + enrollStudent(student: Student): EnrollStatus start time: Time + dropStudent(studentID: Student); void end time: Time + isFull(): boolean + isActive(); boolean ScheduleEntry(location: String, is\_sync: + getters boolean, day of week: String, starttime: Time | + setNumber(num: String): void null, endtime: Time | null) setActiveState(state: boolean): void getTime(): Tuple<Time> + setMaxCapacity(cap: int): void + getLocation(): String + setMaxWaitSize(wait: int): void getDavOfWeek(); String + setInstructor(instructor: Instructor): void isSync(): boolean setSchedule(schedule: ScheduleEntry[]): void

## Serializable Objects (continue)



#### Server

Server StudentSessionHandler student: Student + StudentSessionHandler(Socket, ObjectInputStream, AdminSessionHandler ObjectOutputStream, University, Student) ServerMain + run(): void admin: Administrator - fetchCourses(request: ClientMsg): ServerMsg universities: University∏ - fetchSchedule(request: ClientMsg): ServerMsg + AdminSessionHandler(Socket, ObjectInputStream, enroll(request: ClientMsg): ServerMsg main(): void ObjectOutputStream, University, Administrator) fetchPastEnrollment(request: ClientMsg): ServerMsg loadInfoFromFile(filename: String): void + run(): void logout(request: ClientMsg): ServerMsg - fetchReport(request: ClientMsg): ServerMsg createStudent(request: ClientMsg): ServerMsg addSection(request: ClientMsg): ServerMsg delCourse(request: ClientMsg): ServerMsg enrollStudent(request: ClientMsg): ServerMsg SessionHandler fetchCourses(request: ClientMsg): ServerMsg 0..n logout(request: ClientMsg): ServerMsg # socket: Socket ClientHandler # istream: ObjectInputStream # ostream: ObjectOutputStream socket: Socket «interface» # university: University universities: University∏ <□-----Runnable InstructorSessionHandler # SessionHandler(Socket, ObjectInputStream, + ClientHandler(Socket, University[]) ObjectOutputStream, University) + run(): void instructor: Instructor + run(): void + InstructorSessionHandler(Socket, ObjectInputStream, ObjectOutputStream, University, Instructor) + run(): void viewSections(request: ClientMsg): ServerMsg logout(request: ClientMsg): ServerMsg

#### **GUI**

**Initial UI:** 

Courses Catalog
Schedule
Calendar

**Options** 

Course name: Course number: Login:
Course prefix: Instructor:

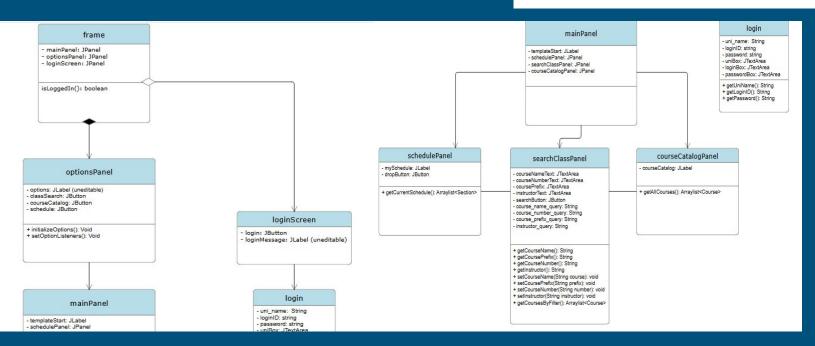
Enter

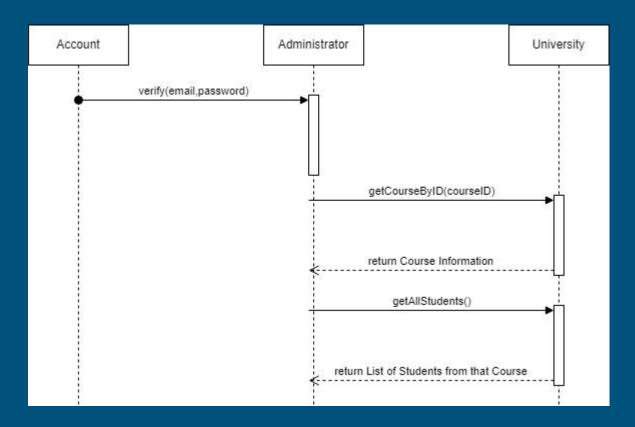
CS 101-(num) Professor (Name) Course Details

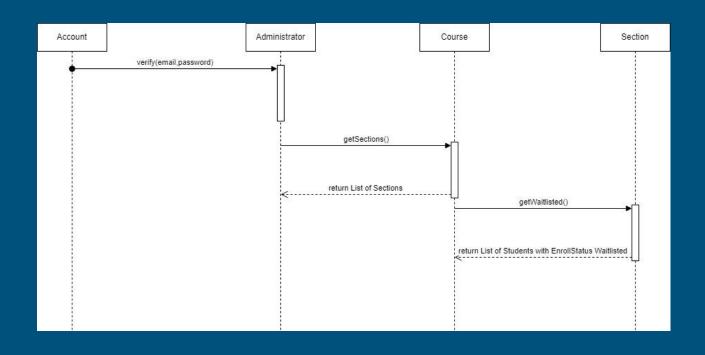
CS 201-(num) Professor (Name) Course Details

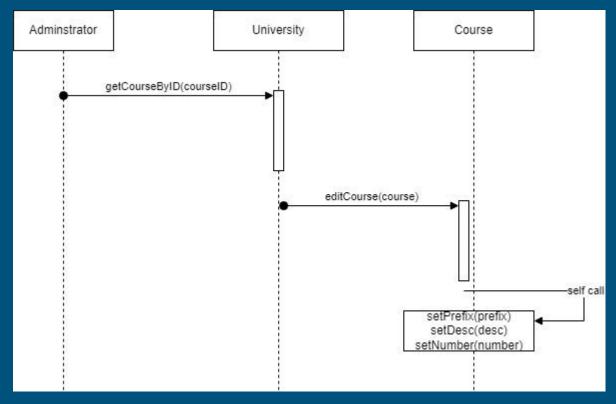
Enroll

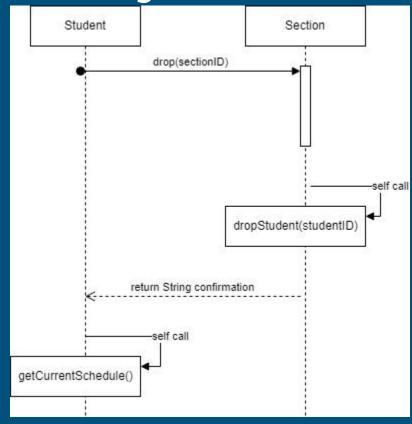
#### Final design:











## Links

- GitHub repository:

https://github.com/MikeJollie2707/CS401-College-Enrollment