Phase 4 Report

State of the Project

Currently a user is able to open the website up across any browser, register an account with First and Last Name, Email, Username and a password that needs to be confirmed. Each of these fields have a validation check as to existence. And the email field checks that what's entered is actually an email. After Registering the account is defaulted to `inactive` until activated by an admin account. Once this is complete the user is able to login using their username and password. An invalid password will arise in an error. An Inactive account will arise in an error. Within an account there are 3 levels Instructor, Chair and Admin. Upon loging in the user can now view there own schedule. Add a course which will let them create a course and add it to their schedule. Delete a course and Switch two courses around. The user can modify their profile which has which department they're in. Their Address, course list and which role they are. If the user is an admin they can view an entire departments schedule, List users and edit their profiles. There is also a notification bubble at the top right which shows changes to the schedule.

Iteration Plan

For our iteration plan, please see our account on rallydev.com to view the user stories we completed and the hours allotted. You can also view the burn down chart for each iteration.

System Modeling and Design

1.

Departments
+ name: string [1] + numberOfLecturers: int [1]
+ __unicode__(s: self): unicode

Course
+ code: string [1] + name: string [1] + department: string [1]
+ __unicode__(s: self): unicode

Room
+ code: string [1] + capacity: int [1]
+ __unicode__(s: self): unicode + getSchedule(): CourseSchedule [*]

Notifications
+ data: string [1]

CourseSchedule
+ code: string [1] + room: string [1] + capacity: int [1] + dayOfWeek: string [1] + startTime: TimeField [1] + endTime: TImeField [1] + typeOfSession: string [1] + enrollment: string [1]
+ time_range(s: self): unicode + getLength(s: self): timedelta

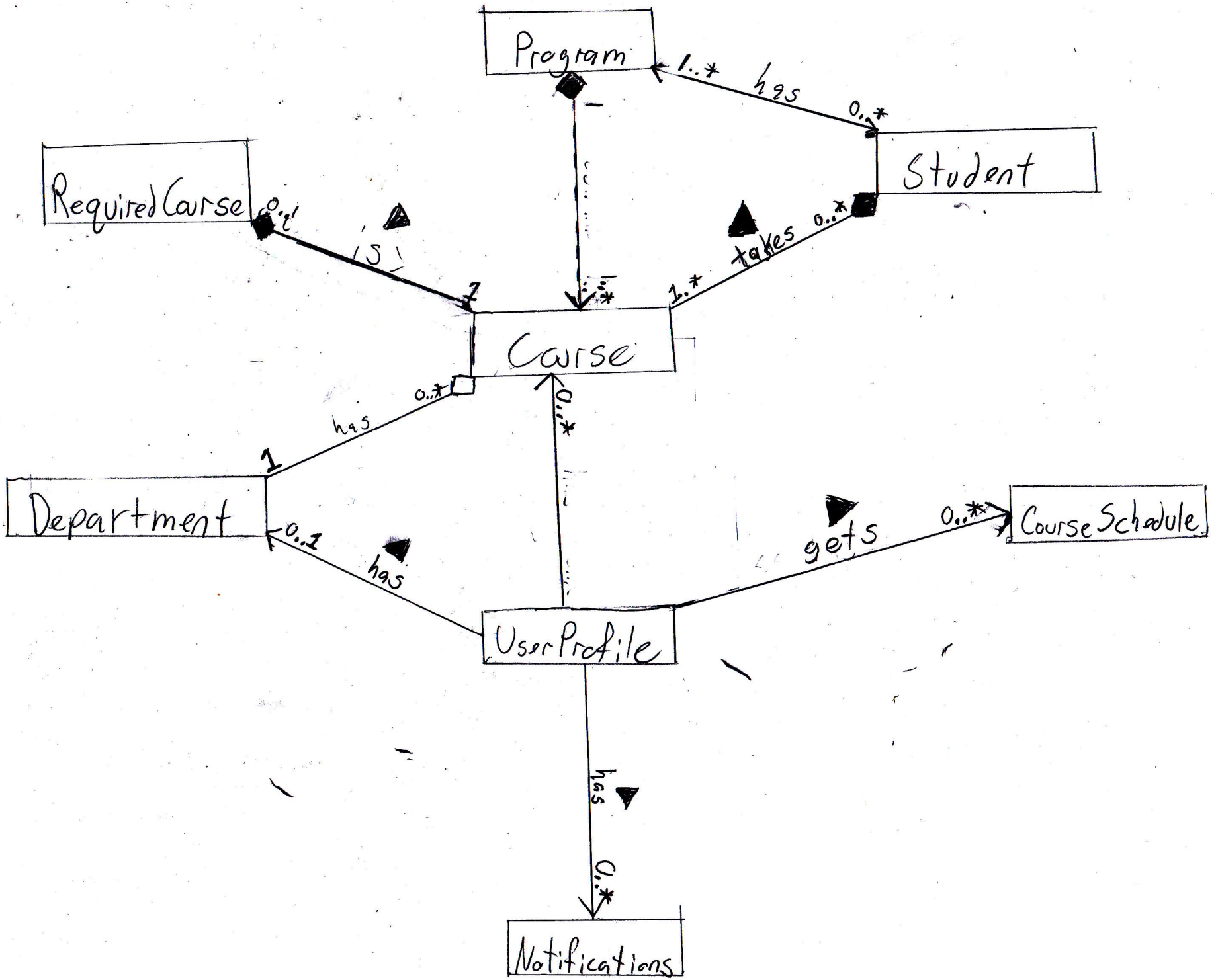
UserProfile
+ user: string [1] + department: string [1] + address: string [1] + myCourses: Course [*] + role: stirng [1] + notifications: stirng [*] + read_notifications: string [*] + notifiy: boolean [1] + active: boolean [1]
+ __str__(s: self): string + is_active(): boolean

Student
+ utorid: string [1] + studentNumber: string [1] + lastName: string [1] + firstName: string [1] + email: string [1] + programCode: string [1] + courses: Course [*]

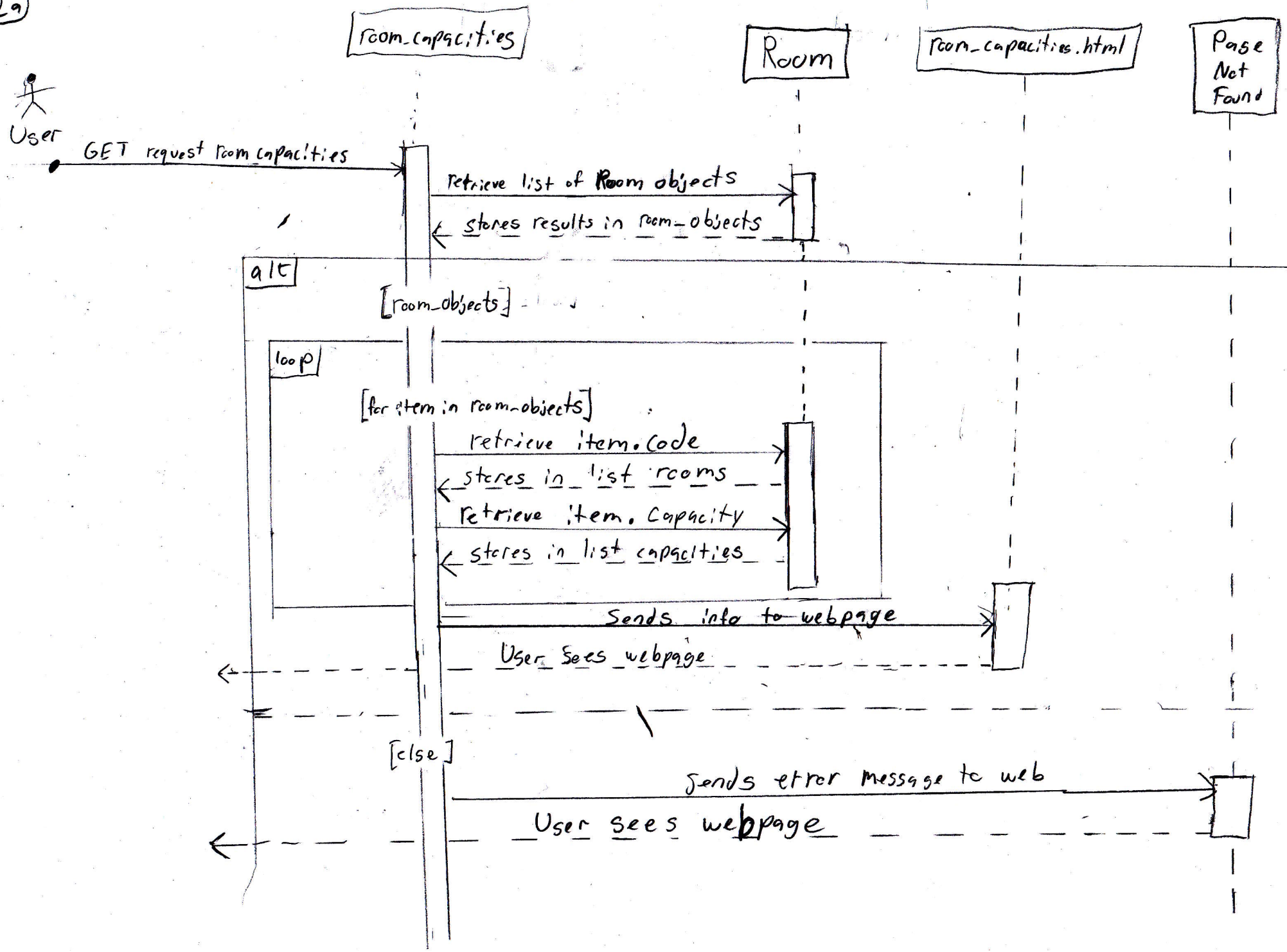
Program
+ code: string [1] + name: string [1] + requiredCourses: Coure [*]

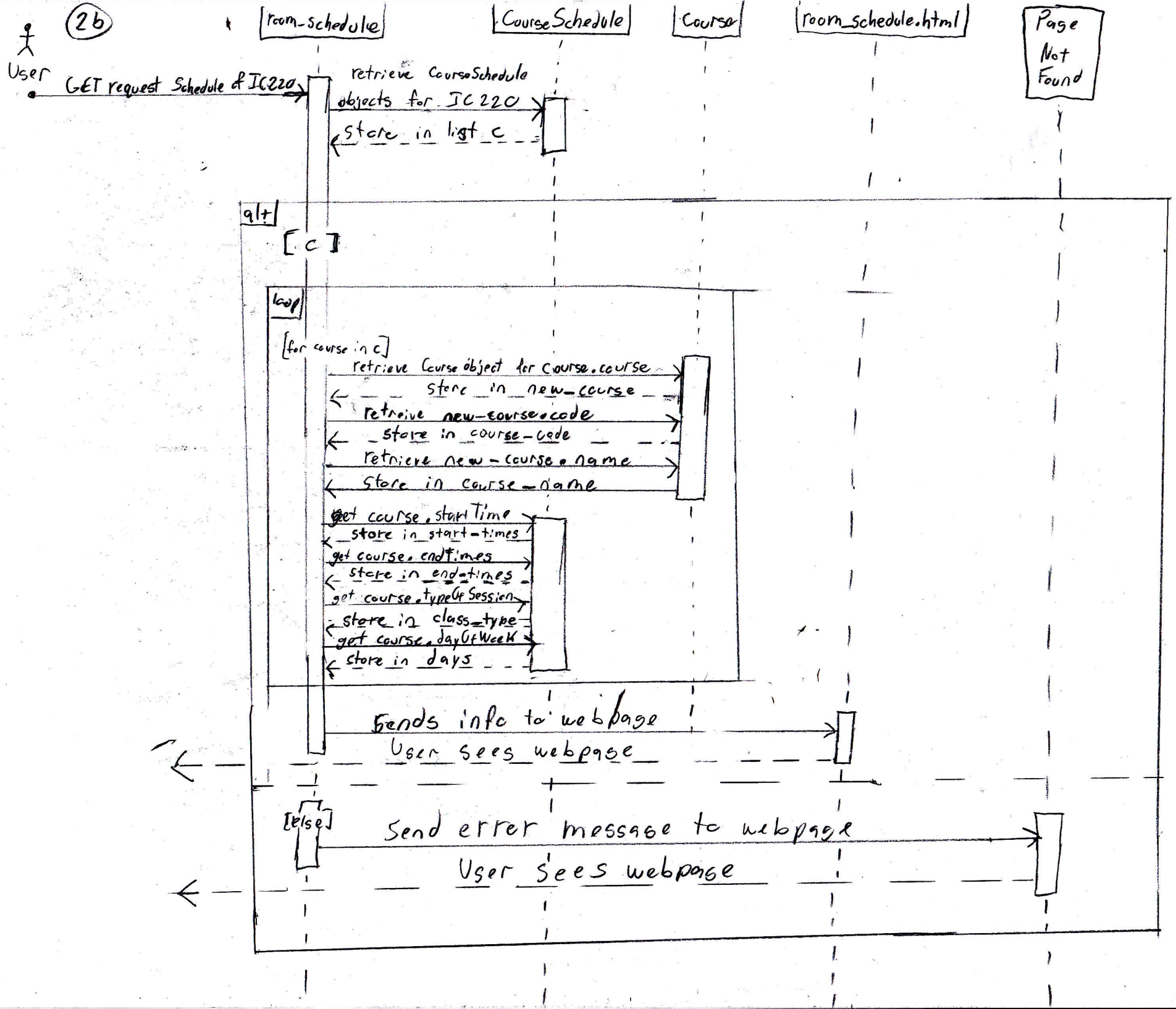
RequiredCourse
+ code: ForeignKey [1] + req_type: CharField [1]

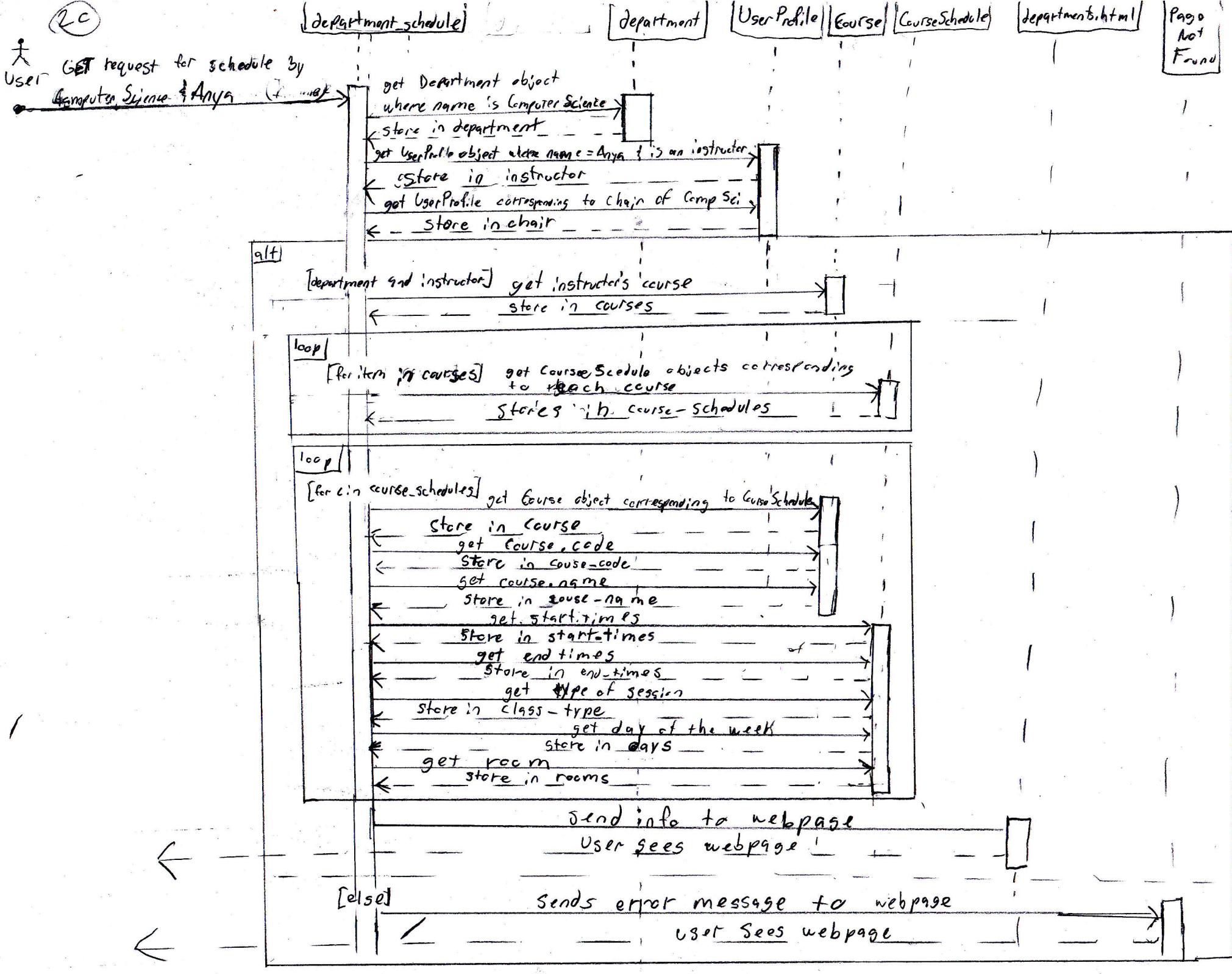
(11)


Program
1..*
has
0..*
Student
RequiredCourse
0..1
is
1
1..*
takes
0..*
Course
0..*
has
1
Department
0..1
has
0..*
gets
0..*
CourseSchedule
UserProfile
has
0..*
Notifications

2a


room_capacities
Room
room_capacities.html
Page Not Found
User
GET request room capacities
retrieve list of Room objects
stores results in room_objects
alt
[room_objects]
loop
[for item in room_objects]
retrieve item.code
stores in list rooms
retrieve item.capacity
stores in list capacities
Sends info to webpage
User sees webpage
[else]
Sends error message to web
User sees webpage


2b
User
room_schedule
CourseSchedule
Course
room_schedule.html
Page Not Found
GET request Schedule of IC220
retrieve CourseSchedule objects for IC220
store in list c
alt
[c]
loop
[for course in c]
retrieve Course object for course.course
store in new_course
retrieve new_course.code
store in course_code
retrieve new_course.name
store in course_name
get course.startTime
store in start_times
get course.endTimes
store in end_times
get course.typeOfSession
store in class_type
get course.dayOfWeek
store in days
Sends info to webpage
User sees webpage
[else]
Send error message to webpage
User sees webpage


2c
department_schedule
department
UserProfile
Course
CourseSchedule
departments.html
Page Not Found
User
GET request for schedule by Computer Science & Anya
get Department object where name is Computer Science
store in department
get UserProfile object where name = Anya & is an instructor
store in instructor
get UserProfile corresponding to chair of Comp Sci
store in chair
alt
[department and instructor]
get instructor's course
store in courses
loop
[for item in courses]
get CourseSchedule objects corresponding to each course
stores in course_schedules
loop
[for c in course_schedules]
get Course object corresponding to CourseSchedule
store in course
get course.code
store in couse_code
get course.name
store in couse_name
get start_times
store in start_times
get end_times
store in end_times
get type of session
store in class_type
get day of the week
store in days
get room
store in rooms
send info to webpage
User sees webpage
[else]
sends error message to webpage
user sees webpage