*COMP 246 --* [*Smart Team: Part B Final Submission*](https://e.centennialcollege.ca/d2l/lms/dropbox/user/folder_submit_files.d2l?db=278043&grpid=480893&isprv=0&bp=0&ou=477093)

Students’ name:

Caique Bispo Ferreira

**Smart Meeting – Your system of smart attendance control.**

# Part B – System Design Document

## The purpose of SDD document and its intended audience

This System Design Document (SDD) is a written description of the Smart Meeting applications (Web Application, Mobile Application and Back-End Application), which provides overall guidance to the architecture of this project. This document not only describes design goals and considerations but also provides the high-level to the low-level overview of the system architecture, which allows all stakeholders (managers, coordinators, developers) to understand and discuss the solution according to the audience. Based on the C4 Model, this SDD provides different levels of abstraction and consists of a hierarchical set of software architecture diagrams for context, containers, components, and code. Moreover, some parts of this document can be shared with the client/user and other stakeholders whose input/approval is needed.

* 1. Table to identify multiplicities of the objects

## Navigation on Domain Classes

An event has multiple lectures

A lecture has one event

A lecture has one room

A lecture has one speaker

A lecture has multiple tickets

A lecture has multiple attendances

A room has multiple lectures

A speaker has multiple lectures

A ticket has one lecture

An attendance has one lecture

An attendance has one ticket or two tickets

An attendance has one current situation

A current situation has multiple attendances

A current situation has one student

A student has multiple current situations

A student has one user

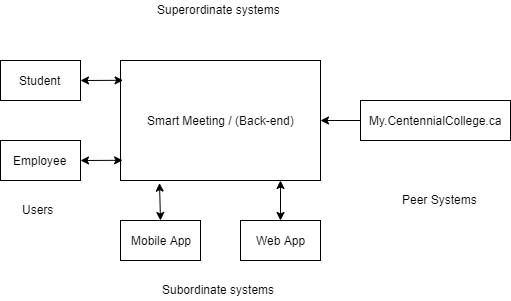
A user has one student

A user has one employee

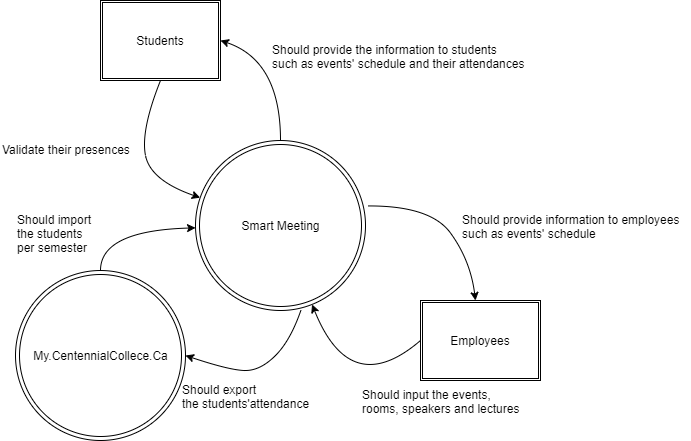
# 

1. Architecture Context Diagram

# How - Architectural Context Diagram

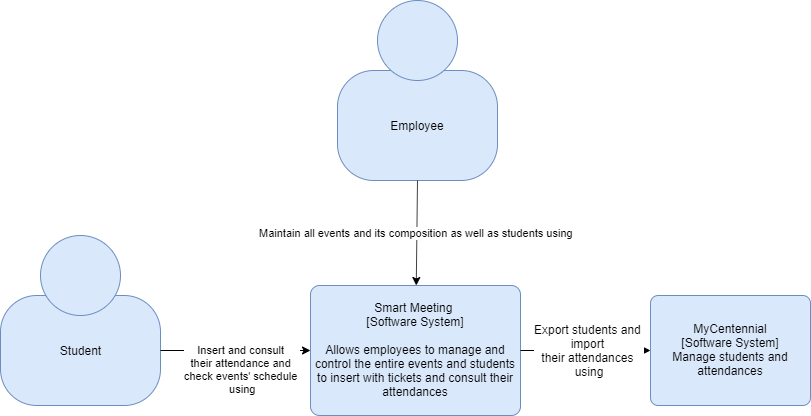


# What – Context Flow Diagram

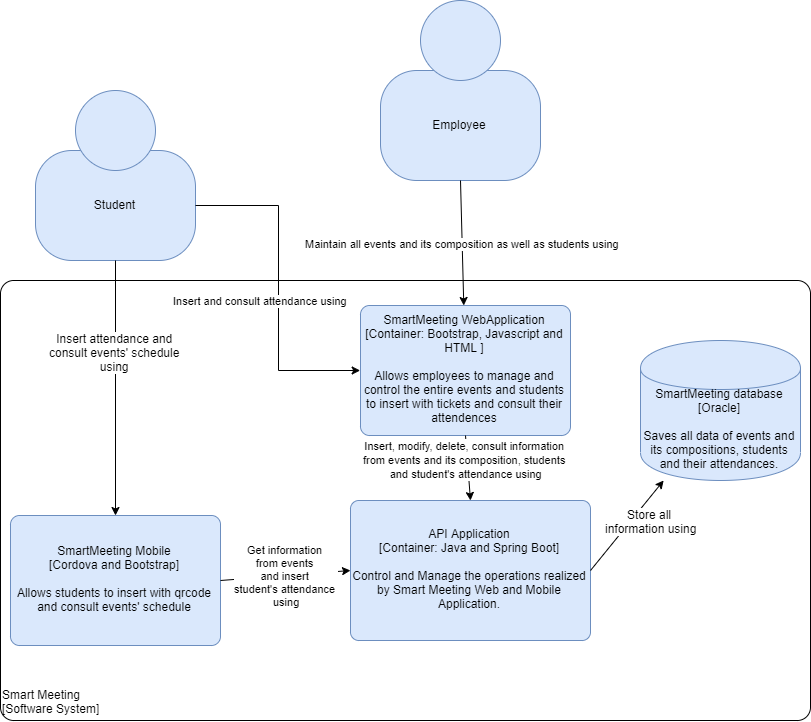


2.1. Container and Component Diagrams. Refer to <https://c4model.com/>

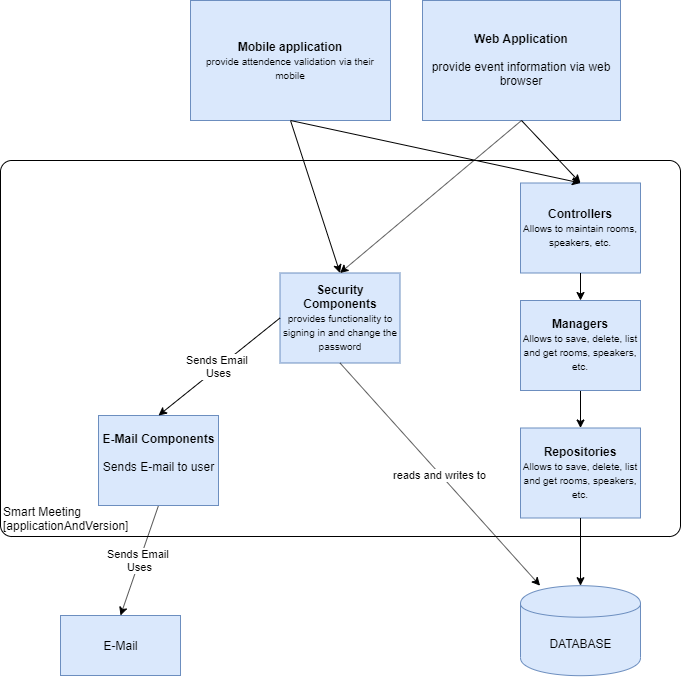
# Level 1 – Context Diagram



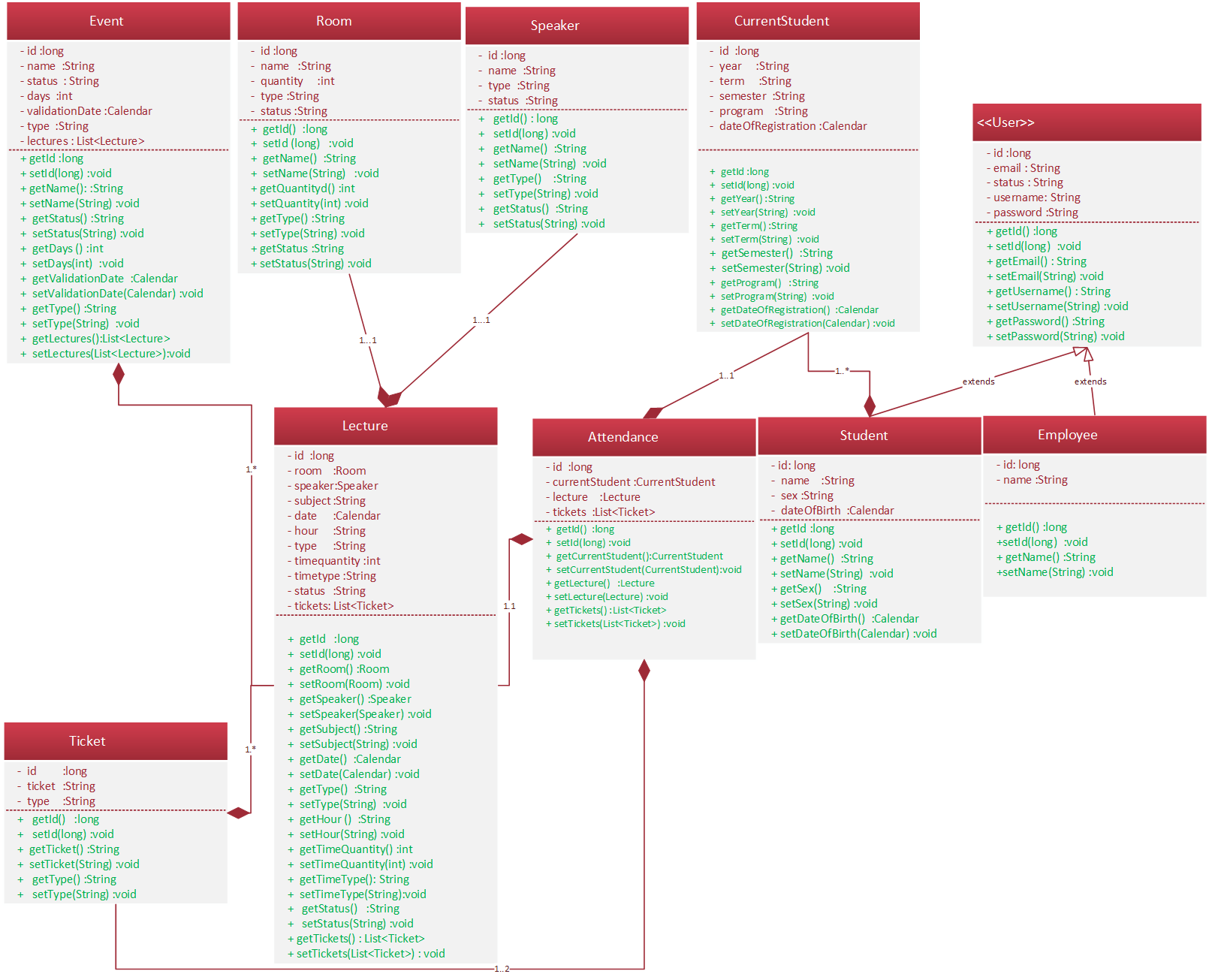
# Level 2 – Container Diagram

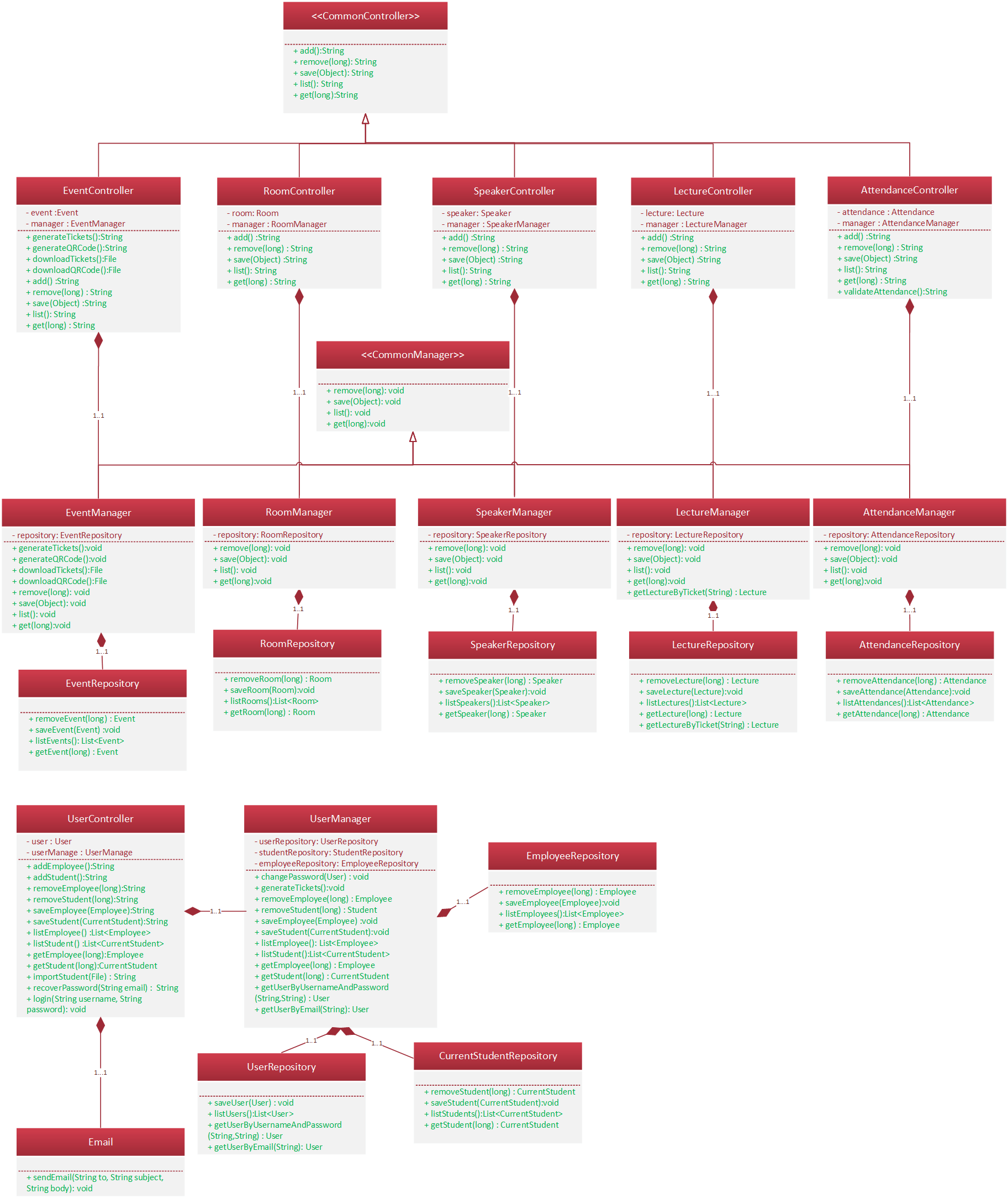


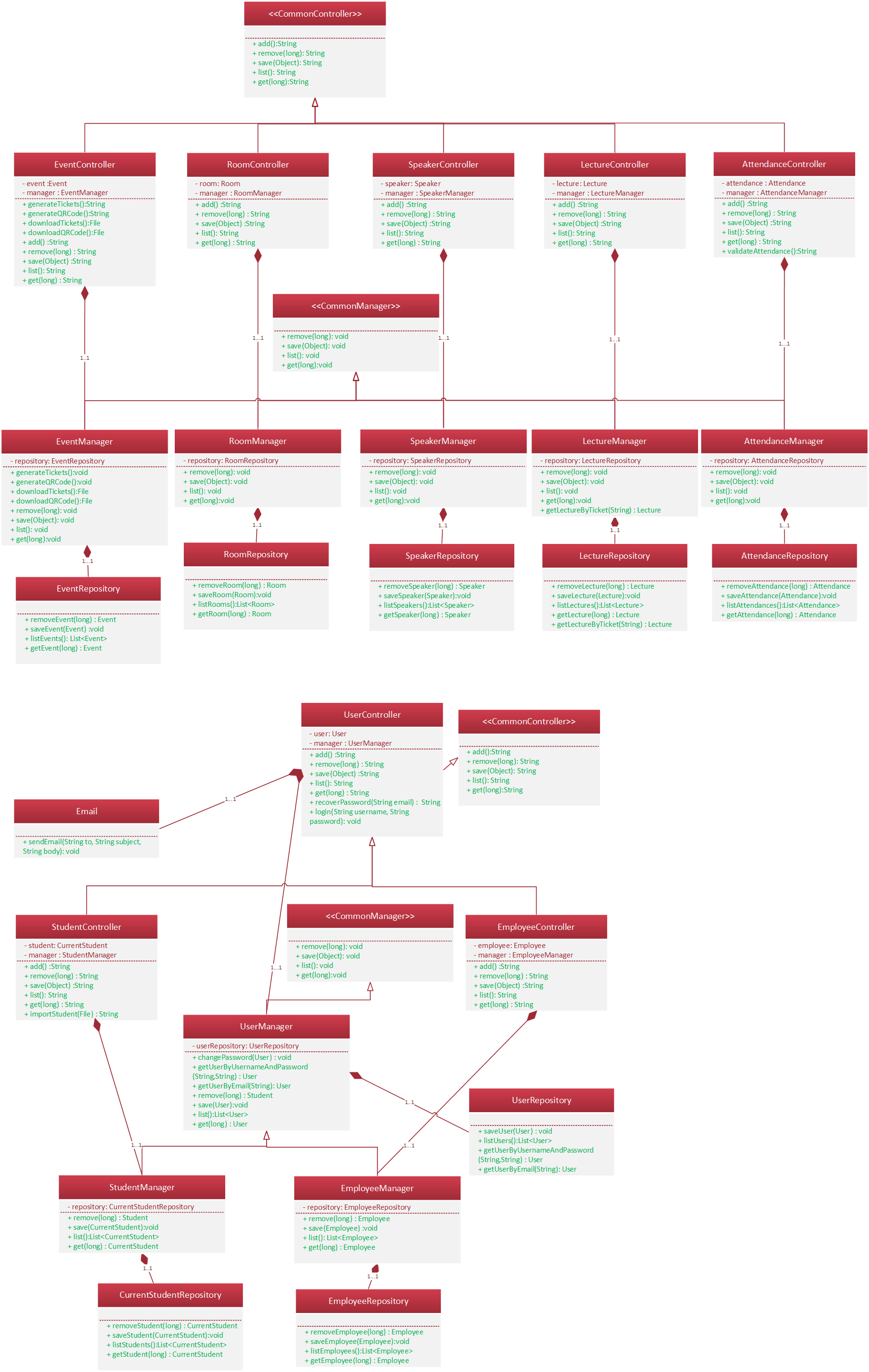
# Level 3 – Component Diagram



1. Design classes using the MVC Design Pattern Architecture







3.1 Create a table to show classes which are V, C and M

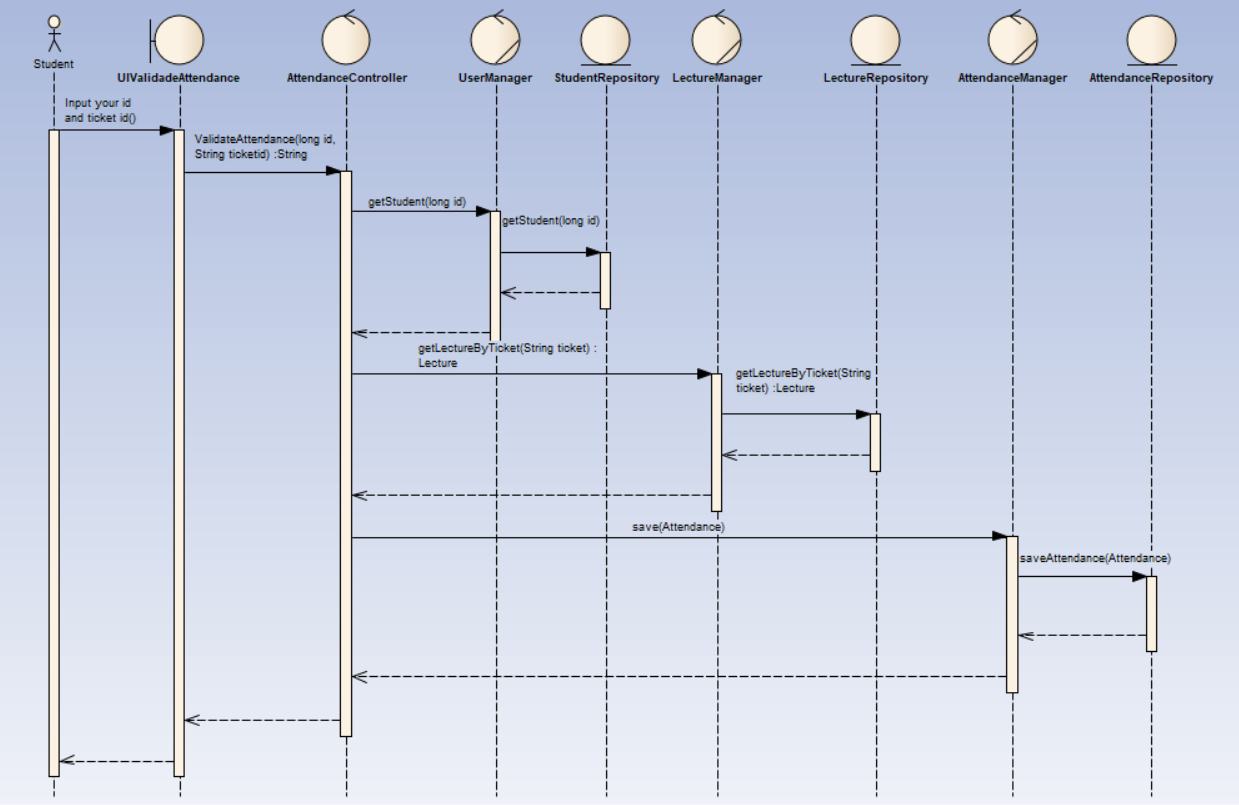
3.2 Mark classes as below

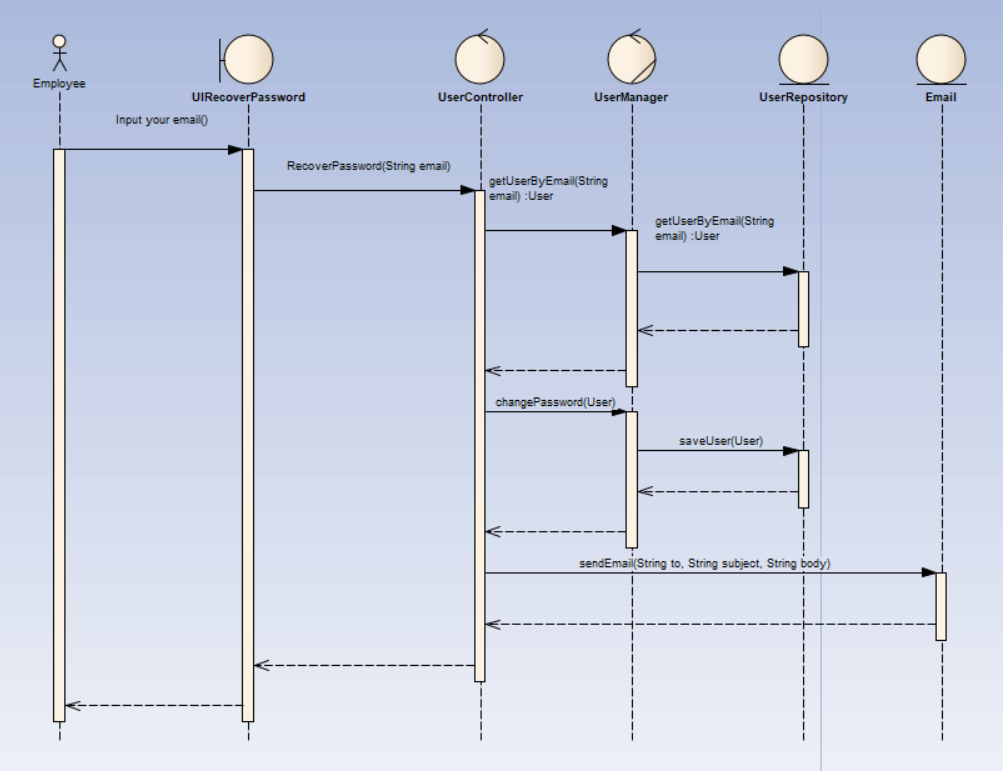
GUI or boundary classes GUI components (V)

Control Classes Components (C)

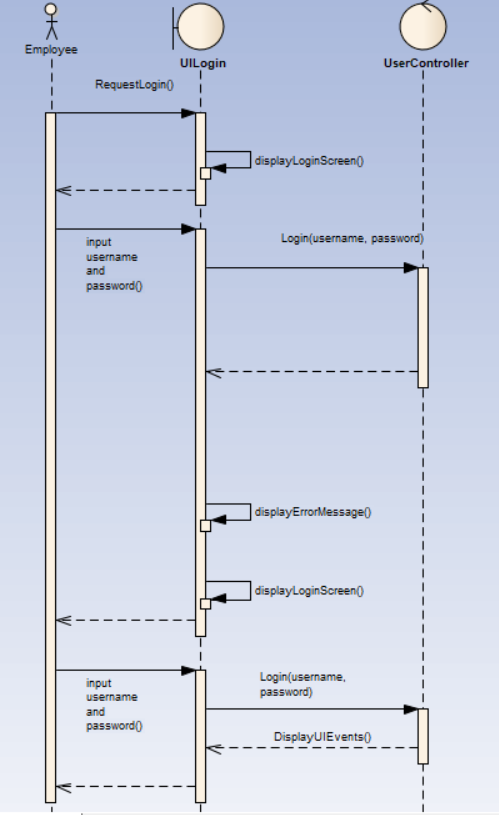
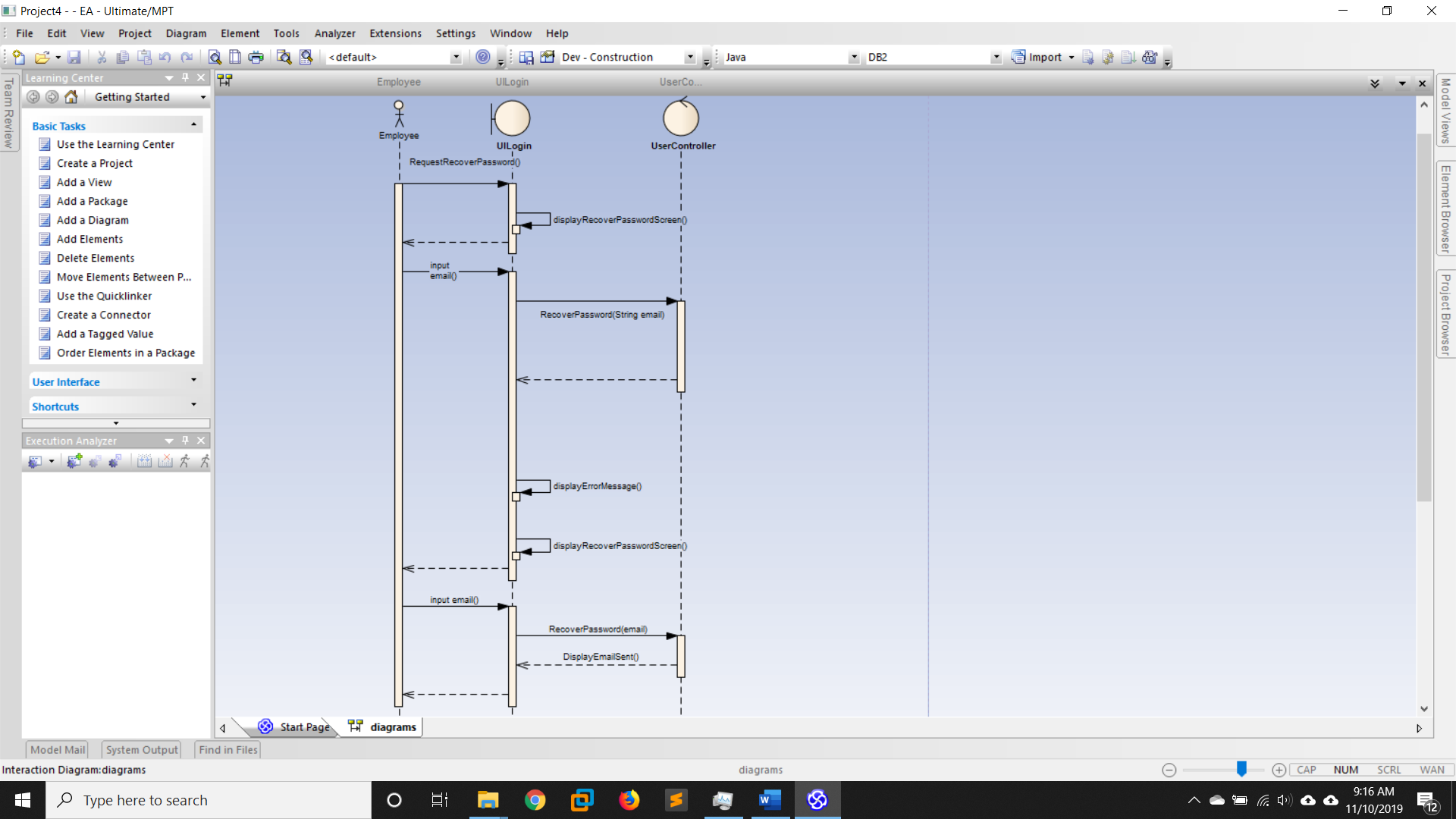
Object Entity Classes components ( M)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SmartMeeting** | | | | |
| **Model** | **Repository** | **Manager** | **Controller** | **View** |
| Event | EventRepository | EventManager | EventController | UIEvent and UIEvents |
| Room | RoomRepository | RoomManager | RoomController | UIRoom and UIRooms |
| Speaker | SpeakerRepository | SpeakerManager | SpeakerController | UISpeaker and UISpeakers |
| Lecture | LectureRepository | LectureManager | LectureController | UILecture and UILectures |
| Attendance | AttendanceRepository | AttendanceManager | AttendanceController | UIAttendance, UIAttendances and UIValidateAttendance |
| Student | StudentRepository | StudentManager | StudentController | UIStudent and UIStudents |
| Employee | EmployeeRepository | EmployeeManager | EmployeeController | UIEmployee and UIEmployees |
| User | UserRepository | UserManager | UserController | UIUser, UIUsers, UIRecoverPassword and UILogin |

3.4.1 Two full and complete Sequence diagrams of TWO separate goal Use Cases 



3.4.2 Draw Two separate UML diagrams to show the navigation through the life-lines to the completion of the tasks – messages are a key factor in this diagram.



# 4 Databases and files – Entity Relation Diagram (ERD Model)

