CS4013 – Group 11

Fabian Hartnett Ferreira – 24432024

Mikael Eramian - 24427233

Karen Kotowska - 24403148

Samuel Wilson – 24426881

Functional Requirements

Admin can:

- Log in using a username and password
- Create and manage modules, rooms and student groups
- Create and edit timetable slots (day, time, room, group, session type)

Lecturer can:

- Log in
- View their timetable

Student can:

- Log in
- View their timetable specific to their programme / group

Rationale:

- **The admin role is the only role that can modify data. This reduces the risk of inconsistent changes from multiple roles**
- **Lecturers and Students can only view timetables relevant to them. Allowing Lecturers to edit their own slots is something that could be added but we didn't to avoid conflict resolution between Admins and Lecturers**

Architecture – MVC

Model:

- User
- Admin
- Lecturer
- Student
- Programme
- Module
- Room
- Timetable
- TimetableSlot
- StudentGroup
- Subgroup
- SessionType
- WeeksPattern

View:

- ConsoleUI
- StudentMenu
- LecturerMenu
- AdminMenu

Controller:

- UserController
- AdminController
- TimetableController

Repository:

- TimetableRepository - Interface
- CSVTimetableRepository – Loading/saving timetables from/to CSV files.

(Add util)

Data Model

Classes: (maybe include the methods for these too and what they do)

User:

- Fields: id, name, password, role
- Used for authentication and authorisation

Admin, Lecturer and Student:

- Extends User
- Lecturer is associated with Modules and TimetableSlots
- Student belongs to a StudentGroup which is indirectly apart of a Programme

Programme:

- Represents a degree course (CS Year 1)
- Associated with Modules and StudentGroups

Module:

- Fields: code, title, weeks, lectureHours, labHours, tutorialHours, year, semester
- Linked to one or more timetableSlots

Room:

- Fields: number, capacity, type (lecture room, lab)

StudentGroup and Subgroup:

- StudentGroup represents a group of students that do the same programme
- Subgroup represents the lab / tutorial group the StudentGroups belong to. (EG. Lab 2D)

Timetable:

- Represents the complete Timetable
- Contains a collection of TimetableSlots

TimetableSlot:

- Fields: module, day, startTime, endTime, semester, weeks, room, lecturer, studentGroup, subgroup, type
- 1 scheduled session in the timetable (Monday, 10:00, 11:00, Semester2, week 3, Room CSG024, Lecture, CS4013, Michael English)

SessionType: (Enum)

- Lecture, Lab, Tutorial

WeeksPattern:

-

Rationale:

Users and roles: We used a base User class that lets Admin, Student and Lecturer extend it. This avoids repeating code and lets each role have different behaviours

Programme, Module, Groups: We seperated these because they represent different concepts. A Programme defines what students study. A Module represents the subject taught. StudentGroups and Subgroups make scheduling easier by grouping students rather than handling every individual.

Timetable and TimetableSlot: A Timetable is made up of TimetableSlots. A TimetableSlot object represents a single session. All relevant information can be added here

Relationships:

- A **Programme** has many **Modules**
- A **Programme** has many **StudentGroups**
- A **StudentGroup** has many **Students**
- A **Timetable** has many **TimetableSlots**
- A **TimetableSlot** has 1 **Module**, **Room**, **SessionType**, **StudentGroup**, **Lecturer**

Rationale:

- A programme has many modules because all students in that programme take those modules
- A timetable has many slots because each slot is 1 scheduled class
- A slot links to exactly 1 module, room, group and lecturer because classes don't use multiple rooms or lecturers at one time.