CS4013: Object Oriented Development – Interim Submission
17/11/2025 Group Number: 25

Name	Student ID
Luigi Curotto – Design & Javadocs	24423882
Thomas Griffin – CRC Cards	24354376
Martina Dumas González Izcaray – UML Diagram	24426636

Table of Contents

Design Overview

This project aims to make a timetabling system which would generate and manage module schedules while respecting constraints. The system is built using a **Model-View Controller (MVC) design** as per the assignment requirements. The **Model** layer has core classes like Room, Lecturer, Module, Programme, StudentGroup, Timeslot, ScheduledClass, Student, StudentGroup and Timetable, storing data loaded from CSV files. The **Controller** is responsible for timetable generation; constraint checking and **View** is a command-line interface for using the system. This modular design keeps responsibilities separate making it easy to add features in the future like a graphical interface and better scheduling methods.

CRC Cards

Programme	
• Store programme code • Store programme name • Store list of year modules • Provide access to programme data	Module Student StudentGroup

Lecturer	
• Store lecturer ID • Store lecturer name • Provide access to ID	Module Timetable

Module	
• Store module code • Store lecture, lab, and tutorial hours • Store list of lecturers • Provide access to module code	Lecturer Programme ScheduledClass

Rooms	
• Store room ID • Store room capacity • Store whether the room is a lab • Provide access to room details	ScheduledClass

ScheduledClass	
• Store module, room, day, and time • Store student group assigned to the class • Represents one scheduled instance of a module	Module Room StudentGroup Timetable

Timetable	
• Store scheduled classes • Add classes to the timetable • Retrieve classes for a lecturer (using lecturer ID filter)	ScheduledClass Lecturer

Student	
• Store student ID and name • Store programme, year, and group • Access basic student details • Update student's group	Programme StudentGroup

StudentGroup	
• Store group ID • Store programme the group belongs to • Store year and size • Provide access to group information	Programme Student ScheduledClass

ClashDetector	

• Check if a new ScheduledClass conflicts with existing ones
  • Detect clashes in:
  • Room usage
  • Lecturer availability
  • StudentGroup timetable conflicts
• Provide methods to validate a Timetable before adding sessions
	Timetable
	ScheduledClass
	Module
	Room
	StudentGroup

TimetableGenerator	
• Generate a complete timetable automatically	Timetable Module

• Assign rooms, times, and days to modules • Ensure no clashes (via ClashDetector) • Use available modules, groups, rooms, and lecturers to build schedule • Add ScheduledClass entries into Timetable	StudentGroup
	Room
	ClashDetector
	ScheduledClass
	Programme / ProgrammeStructure

CSVReader	
• Open and read CSV files line-by-line • Parse CSV rows into lists/arrays • Provide data to DataManager • Handle I/O exceptions gracefully	DataManager
	File system

CSVWriter	
• Write model data into CSV files • Convert objects (Student, Room, Module, etc.) into CSV rows • Save timetable outputs to CSV • Handle file writing and I/O errors	DataManager
	Timetable
	File system

DataManager	

• Load all system data from CSV files using CSVReader
• Save data using CSVWriter
• Construct objects:
  • Students
  • Rooms
  • Modules
  • Programme / ProgrammeStructure
  • StudentGroups
• Provide getters for loaded data to timetable generator / CLI
• Maintain in-memory collections (lists/maps)
	CSVReader

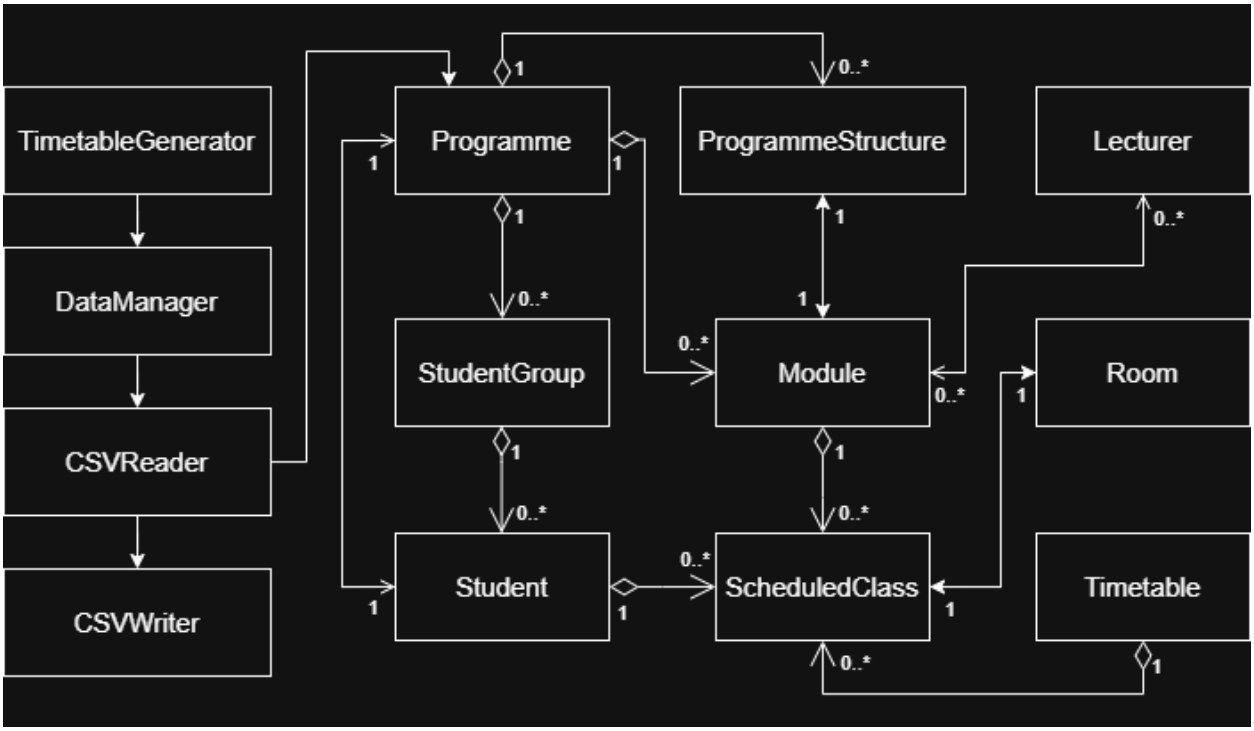
	CSVWriter
	Student, Room, Module, Programme, ProgrammeStructure, StudentGroup
	Timetable

ProgrammeStructure	
• Define which modules belong to each Programme for each year • Provide lookup for: • Modules for a given programme & year • Represent the curriculum structure	Programme Module StudentGroup DataManager

CLI	
• Provide text-based user interface • Display menus and prompt user for choices • Create, view, or modify timetables • Trigger TimetableGenerator • Display clashes or errors • Communicate with DataManager and Timetable	Main DataManager Timetable TimetableGenerator ClashDetector

Main	
• Entry point of the application • Initialize DataManager • Initialize CLI • Start application run loop • Handle startup errors (file loading, etc.)	CLI DataManager Timetable

UML Diagram


TimetableGenerator
DataManager
CSVReader
CSVWriter
Programme
StudentGroup
Student
ProgrammeStructure
Module
ScheduledClass
Lecturer
Room
Timetable
1
0..*