Course Scheduler Design Layout

The classes that end with Entry are the classes that describe what is contained in one row of each table. They describe the columns of the database table.

Classes	Properties	Methods
Semester – this is not actual a separate class because it only has one property which we can represent with the String class.	*Semester (String)	
CourseEntry	*CourseCode (String) Description (String)	Constructor getters
ClassEntry	*Semester (String) *CourseCode (String) Seats (integer)	Constructor getters
ClassDescription	CourseCode (String) Description (String) Seats (int)	Constructor getters
StudentEntry	*StudentID (String) FirstName (String) LastName (String)	Constructor getters
ScheduleEntry	*Semester (String) *CourseCode (String) *StudentID (String) Status (String) – "S" or "W" Timestamp (Timestamp)	Constructor getters

The classes that end with Queries are the classes that access and update the database tables. They return ArrayLists of the Entry classes when the information in the rows of the database table is requested.

Classes	Return value	Methods	Part 2
SemesterQueries	Void	addSemester(String semester)	
	ArrayList<String>	getSemesterList()	
CourseQueries	Void	addCourse(CourseEntry course)	
	ArrayList<String>	getAllCourseCodes(String semester)	
ClassQueries	Void	addClass(ClassEntry class)	
	ArrayList<String>	getAllCourseCodes(String semester)	
	Int	getClassSeats(String semester, String courseCode)	
	void	dropClass(String semester, String courseCode)	x
StudentQueries	void	addStudent(StudentEntry student)	
	ArrayList<StudentEntry>	getAllStudents()	
	StudentEntry	getStudent(String studentID)	x
	void	dropStudent(String studentID)	x
ScheduleQueries	void	addScheduleEntry(ScheduleEntry entry)	
	ArrayList<ScheduleEntry>	getScheduleByStudent(String semester, String studentID)	
	int	getScheduledStudentCount(currentSemester, courseCode)	
	ArrayList<ScheduleEntry>	getWaitlistedStudentsByClass(String semester, String courseCode)	x
	void	dropStudentScheduleByCourse(String semester, String studentID, String	x
	void	courseCode)	
	void	dropScheduleByCourse(String semester, String courseCode)	x
	void	updateScheduleEntry(ScheduleEntry entry)	x
MultiTableQueries	Arraylist<ClassDescription>	getAllClassDescriptions(String semester)	
	ArrayList<StudentEntry>	getScheduledStudentsByClass(String semester, String courseCode)	x
	ArrayList<StudentEntry>	getWaitlistedStudentsByClass(String semester, String courseCode)	x

Note: The methods with an x in the Part 2 column are only needed for Part 2 of the Final Project.

Note: There may be other methods needed in your classes. This list is not all inclusive.

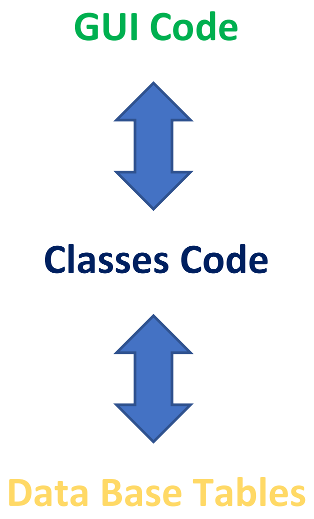
Data Base Tables

The Data Base tables have a one-to-one correspondence with the classes above that contain properties. This is the advantage of Object-Oriented Design, the data from a table is the data for a class object. So, in this project, the tables would be:

Semester
Course
Class
Student
Schedule

Then we need to figure out what the columns of each table will be. That is also easy, it is just the properties of each of the classes above. For example, the Course Table would have the columns: CourseCode, Description. The only other thing we need to figure out is what is the primary key for each table. That would be the columns from each table that will make an entry unique. The properties that have * in front of them make up the primary key for each Table.

3 Tier Application Development



GUI Code

Classes Code

Data Base Tables