3.2

**3.2.1 IScheduler**

The scheduler interface manages schedule generation as well as validation of conflicts, prerequisites and student preferences.

**Class Scheduler**

* addCourse(course\_id)
  + Description: Adds course to scheduler.
  + Valid Parameters: Course id must correspond to existing course id in current semester.
  + Return Type: Void
* generateSchedules()
  + Description: Generates a set of possible schedules.
  + Valid Parameters: None
  + Return Type: Array of Schedules
* generate(generator\_course\_list)
  + Description: Generates a schedule based off courses added.
  + Valid Parameters: Course List must correspond to available courses in current semester.
  + Return Type: Schedule
* commitSchedule(schedule)
  + Description: Replaces users current schedule with given schedule.
  + Valid Parameters: Schedule object must be generated from Scheduler class.
  + Return Type: Void
* addPrefrences(Preference [\*] )
  + Description: Adds selected preferences to current user preferences.
  + Valid Parameters: An array of preferences containing preference objects that correspond to list of available preferences.
  + Return Type: Void

**3.2.2 IDatabase**

The database interface connects the application and the database together. The interface allows for retrieval, insertion, and deletion of information within the database.

**Class DatabaseDriver**

* query(query\_string)
  + Description: Returns database results depending on the query.
  + Valid Parameters: Query string must correspond to SQL query format.
  + Return Type: Return type varies according to Query.

**3.2.3 IAuthentication**

The authentication interface validates the login of users as well as providing security and restrictions for every user session or requests.

**Class User**

* authenticate(login\_name, password)
  + Description: Validates user login.
  + Valid Parameters: Login name and password must be 6 to 255 characters long.
  + Return Type: Boolean
* is\_admin(user\_id)
  + Description: Determines if user is an administrator to grant more access.
  + Valid Parameters: User id must be 6 to 255 characters long.
  + Return Type: Boolean

**Class Login**

* index()
  + Description: Validates if authentication method has succeeded by tracking the user’s session. Redirects to homepage or an error message according to result.
  + Valid Parameters: None
  + Return Type: Void

**Class CI\_Security**

* csrf\_verify()
  + Description: Validates sessions and POST requests.
  + Valid Parameters: None
  + Return Type: Session object

**3.2.4 ICourseManagement**

The course management interface is responsible for handling the assignment of course tutorials, laboratories, sections, location, and time of all courses within any given semester. The interface also authenticates course requisites and availabilities.

**Class Schedule**

* addSection(section)
  + Description: Attaches a section to the scheduler.
  + Valid Parameters: Section object containing an instructor.
  + Return Type: Void
* addTutorial(tutorial)
  + Description: Attaches a tutorial to the scheduler.
  + Valid Parameters: Tutorial object containing a teaching assistant.
  + Return Type: Void
* addLaboratory(tutorial)
  + Description: Attaches a laboratory to the scheduler.
  + Valid Parameters: Laboratory object containing a lab instructor.
  + Return Type: Void

**Class Course**

* isAvailable(semester)
  + Description: Determines if a course is offered in given semester.
  + Valid Parameters: A valid semester within the scheduler.
  + Return Type: Boolean
* getPrerequisites()
  + Description: Obtains all perquisites of a given course.
  + Valid Parameters: None
  + Return Type: An array of course objects
* arePrerequisitesMet()
  + Description: Verifies if course perquisites are satisfied.
  + Valid Parameters: None
  + Return Type: Boolean

**Class TimeBlock**

* overlaps(Timeblock)
  + Description: Determines if given timeblock conflicts with the parameter timeblock.
  + Valid Parameters: Timeblock object with start time before end time.
  + Return Type: Boolean