**3.2 Subsystem Interfaces Specifications**

The system consists of mainly three interfaces: IUserInteraction, IScheduleManagement, and ICourseManagement. Each of those interfaces provides a set of methods that components can publicly access. The classes that implements those interfaces are the ones presented in the architecture diagram (see section 3.1), and the use of the method calls are explained in more detail in dynamic design scenario (see section 5).

**3.2.1 IUserInteraction**

The user interaction interface provides methods for students to perform actions related to their own schedule.

|  |  |
| --- | --- |
| **Method** | |
| getAllSavedSchedules(studentID:long) list of Schedule objects | |
| **Class implementing this method** | **Specification** |
| UI | ***Description:***  Retrieves and returns all saved schedules of a student given its student ID.  ***Input parameter(s):***  studentID:long representing the student ID  ***Condition(s) for validity:***  studentID ≥ 0  ***Return type:***  list of Schedule objects |

|  |  |
| --- | --- |
| **Method** | |
| getSavedSchedule(scheduleID:long) Schedule | |
| **Class implementing this method** | **Specification** |
| UI | ***Description:***  Given a schedule ID, retrieves and returns a specific saved schedule that represents the schedule of a semester.  ***Input parameter(s):***  scheduleID:long representing the schedule ID  ***Condition(s) for validity:***  scheduleID ≥ 0  ***Return type:***  Schedule |

|  |  |
| --- | --- |
| **Method** | |
| saveSchedule(scheduleID:long) | |
| **Class implementing this method** | **Specification** |
| UI | ***Description:***  Saves a schedule given its ID to an account.  ***Input parameter(s):***  scheduleID:long representing the schedule ID  ***Conditions for validity:***  scheduleID ≥ 0  ***Return type:***  void |

|  |  |
| --- | --- |
| **Method** | |
| searchCourse(courseID:long) list of Section objects | |
| **Class implementing this method** | **Specification** |
| UI | ***Description:***  Retrieves and returns information about a course, including the course’s sections, given the course ID.  ***Input parameter(s):***  courseID:long representing the course ID  ***Conditions for validity:***  courseID ≥ 0  ***Return type:***  list of Section objects |

|  |  |
| --- | --- |
| **Method** | |
| selectWeek(date:int, month:int) Schedule | |
| **Class implementing this method** | **Specification** |
| UI | ***Description:***  Provides the caller with a weekly schedule view given the date and month as valid integer values for the targeted week.  ***Input parameter(s):***  date:int representing a date in the targeted week  month:int representing the month of the targeted week  ***Conditions for validity:***  1 ≤ date ≤ 31  1 ≤ month ≤ 12  ***Return type:***  Schedule |

|  |  |
| --- | --- |
| **Method** | |
| viewAcademicRecord(studentID:long) AcademicRecord | |
| **Class implementing this method** | **Specification** |
| UI | ***Description:***  Allows to view the academic record of a student given the student’s ID.  ***Input parameter(s):***  studentID:long representing the ID of the student  ***Conditions for validity:***  studentID ≥ 0  ***Return type:***  AcademicRecord |

|  |  |
| --- | --- |
| **Method** | |
| viewCourseInfo(courseID:long, sectionID:long) list of Section objects | |
| **Class implementing this method** | **Specification** |
| UI | ***Description:***  Retrieves and returns information about a course given the course ID and the section ID.  ***Input parameter(s):***  courseID:long representing the ID of the course  sectionID:long representing the ID of the section  ***Conditions for validity:***  courseID ≥ 0  sectionID ≥ 0  ***Return type:***  list of Section objects |

**3.2.2 IScheduleManagement**

The IScheduleManagement interface consists of a set of methods that provide another component the ability of create, modify and delete a schedule.

|  |  |
| --- | --- |
| **Method** | |
| addCourse(courseID:long, sectionID:long) Schedule | |
| **Classes implementing this method** | **Specification** |
| UI,  Schedule | ***Description:***  Adds a course given the course ID and the section ID to a student’s schedule.  ***Input parameter(s):***  courseID:long representing the ID of the course  sectionID:long representing the ID of the section  ***Conditions for validity:***  courseID ≥ 0  sectionID ≥ 0  ***Return type:***  Schedule |

|  |  |
| --- | --- |
| **Method** | |
| bookTimeSlot(times:list of int) | |
| **Class implementing this method** | **Specification** |
| Schedule | ***Description:***  Sets time slots in a scheduled as occupied.  ***Input parameter(s):***  times:list of int representing the IDs of the time blocks to be booked  ***Conditions for validity:***  ≥ 0  length of times > 0  ***Return type:***  void |

|  |  |
| --- | --- |
| **Method** | |
| deleteSchedule(scheduleID:long) | |
| **Class implementing this method** | **Specification** |
| Schedule | ***Description:***  Deletes a schedule given its schedule ID.  ***Input parameter(s):***  scheduleID:long representing the ID of the schedule to be deleted  ***Conditions for validity:***  scheduleID ≥ 0  ***Return type:***  void |

|  |  |
| --- | --- |
| **Method** | |
| freeCredits(studentID:long) int | |
| **Class implementing this method** | **Specification** |
| Schedule | ***Description:***  Returns the number of free credits a student is still allowed to take.  ***Input parameter(s):***  studentID:long representing the ID of the student  ***Conditions for validity:***  studentID ≥ 0  ***Return type:***  int |

|  |  |
| --- | --- |
| **Method** | |
| freeTimeBlock(startTime:int, endTime:int) | |
| **Class implementing this method** | **Specification** |
| Schedule | ***Description:***  Frees a time block of a schedule given the start and end time of the block.  ***Input parameter(s):***  startTime:int representing the time at which the time block starts  endTime:int representing the time at which the time block ends  ***Conditions for validity:***  startTime ≥ 0  endTime > 0  endTime > startTime  ***Return type:***  void |

|  |  |
| --- | --- |
| **Method** | |
| generateWeeklyView(startDate:String, endDate:String) Schedule | |
| **Class implementing this method** | **Specification** |
| Schedule | ***Description:***  Generates a weekly view of a schedule given the start date and end date  ***Input parameter(s):***  startDate:String representing the date at which the week starts (YYMMDD)  endDate:String representing the date at which the week ends (YYMMDD)  ***Conditions for validity:***  length of startDate = 6  length of endDate = 6  startDate and endDate contains only characters between ‘0’ and ‘9’  ***Return type:***  Schedule |

|  |  |
| --- | --- |
| **Method** | |
| getSavedSchedule(studentID:long) list of Schedule objects | |
| **Class implementing this method** | **Specification** |
| SavedSchedules | ***Description:***  Retrieves and returns all saved schedules of a student given the student’s ID.  ***Input parameter(s):***  studentID:long representing the ID of the student  ***Conditions for validity:***  studentID ≥ 0  ***Return type:***  list of Schedule objects |

|  |  |
| --- | --- |
| **Method** | |
| getSchedule(scheduleID:long) Schedule | |
| **Class implementing this method** | **Specification** |
| SavedSchedules | ***Description:***  Retrieves and returns the schedule of a semester given the ID of the schedule.  ***Input parameter(s):***  scheduleID:long representing the ID of the schedule  ***Conditions for validity:***  scheduleID ≥ 0  ***Return type:***  Schedule |

|  |  |
| --- | --- |
| **Method** | |
| isTimeSlotFree(times:list of int) boolean | |
| **Class implementing this method** | **Specification** |
| Schedule | ***Description:***  Verifies whether all time slots, which IDs are given as parameter, are available in a schedule. Returns true if all time slots are free in the schedule, false otherwise.  ***Input parameter(s):***  times:list of int representing the IDs of the time slots to be checked  ***Conditions for validity:***  times, ≥ 0  length of times > 0  ***Return type:***  boolean |

|  |  |
| --- | --- |
| **Method** | |
| removeCourse(courseID:long, sectionID:long, studentID:long) Schedule | |
| **Class implementing this method** | **Specification** |
| UI,  Schedule | ***Description:***  Removes a given course section from the schedule of a student.  ***Input parameter(s):***  courseID:long representing the ID of the course to be removed  sectionID:long representing the ID of the section to be removed  studentID:long representing the ID of the student  ***Conditions for validity:***  courseID ≥ 0  sectionID ≥ 0  studentID ≥ 0  ***Return type:***  Schedule |

|  |  |
| --- | --- |
| **Method** | |
| saveNewSchedule(scheduleID:long, studentID:long) list of Schedule objects | |
| **Class implementing this method** | **Specification** |
| SavedSchedule | ***Description:***  Saves a new schedule related to a student given both the schedule ID and the student ID, then returns an updated list of schedules that have been saved.  ***Input parameter(s):***  scheduleID:long representing the ID of the schedule to be saved  studentID:long representing the ID of the student linked to that schedule  ***Conditions for validity:***  scheduleID ≥ 0  studentID ≥ 0  ***Return type:***  list of Schedule objects |

|  |  |
| --- | --- |
| **Method** | |
| saveToAccount(scheduleID:long) list of Schedule objects | |
| **Class implementing this method** | **Specification** |
| Schedule | ***Description:***  Saves a schedule given its ID to an account and returns an updated list of the saved schedules.  ***Input parameter(s):***  scheduleID:long representing the ID of the schedule to be saved  ***Conditions for validity:***  scheduleID ≥ 0  ***Return type:***  list of Schedule objects |

|  |  |
| --- | --- |
| **Method** | |
| updateSequence(courseID:long) | |
| **Class implementing this method** | **Specification** |
| Sequence | ***Description:***  Updates a course sequence.  ***Input parameter(s):***  courseID:long representing the ID of the course  ***Conditions for validity:***  courseID ≥ 0  ***Return type:***  void |

**3.2.3 ICourseManagement**

The ICourseManagement interface allows to consult information about academic courses, including courses, sections and academic records. It also allows to perform modification on courses, such as adding and removing students from a course.

|  |  |
| --- | --- |
| **Method** | |
| addStudent(studentID:long) | |
| **Class implementing this method** | **Specification** |
| Section | ***Description:***  Adds a student to the list of registered student for a section given the student ID.  ***Input parameter(s):***  studentID:long representing the ID of the student  ***Conditions for validity:***  studentID ≥ 0  ***Return type:***  void |

|  |  |
| --- | --- |
| **Method** | |
| addToList(studentID:long) | |
| **Class implementing this method** | **Specification** |
| Course | ***Description:***  Adds a student to the list of registered student for a course given the student ID.  ***Input parameter(s):***  studentID:long representing the ID of the student  ***Conditions for validity:***  studentID ≥ 0  ***Return type:***  void |

|  |  |
| --- | --- |
| **Method** | |
| checkPassed(prereqs:Course) boolean | |
| **Class implementing this method** | **Specification** |
| CoursesPassed | ***Description:***  Verifies that a student has passed a prerequisite course. Returns true if the student has passed, false otherwise.  ***Input parameter(s):***  courseID:long representing the ID of the prerequisite course  ***Conditions for validity:***  courseID ≥ 0  ***Return type:***  boolean |

|  |  |
| --- | --- |
| **Method** | |
| checkPrereqs(studentID:long, courseID:long) boolean | |
| **Class implementing this method** | **Specification** |
| Course | ***Description:***  Verifies that a student has passed a prerequisite course provided the student ID and the course ID. Returns true if the student has passed, false otherwise.  ***Input parameter(s):***  studentID:long representing the ID of the student  courseID:long representing the ID of the prerequisite course  ***Conditions for validity:***  studentID ≥ 0  courseID ≥ 0  ***Return type:***  boolean |

|  |  |
| --- | --- |
| **Method** | |
| gatherCoursesPassed(studentID:long) AcademicRecord | |
| **Class implementing this method** | **Specification** |
| CoursesPassed | ***Description:***  Creates and returns an academic record of all completed courses by a student.  ***Input parameter(s):***  studentID:long representing the ID of the student  ***Conditions for validity:***  studentID ≥ 0  ***Return type:***  AcademicRecord |

|  |  |
| --- | --- |
| **Method** | |
| generateRecord(passedCourses:list of Course objects, studentID:long) AcademicRecord | |
| **Class implementing this method** | **Specification** |
| AcademicRecord | ***Description:***  Creates and returns an academic record of all completed courses by a student.  ***Input parameter(s):***  passedCourses:list of Course objects representing all the courses passed  studentID:long representing the ID of the student  ***Conditions for validity:***  studentID ≥ 0  ***Return type:***  AcademicRecord |

|  |  |
| --- | --- |
| **Method** | |
| getAllSections(courseID:long) list of Section objects | |
| **Class implementing this method** | **Specification** |
| Section | ***Description:***  Returns a list containing all the sections of a course.  ***Input parameter(s):***  courseID:long representing the ID of the course to be consulted  ***Conditions for validity:***  courseID ≥ 0  ***Return type:***  list of Section objects |

|  |  |
| --- | --- |
| **Method** | |
| getCourseInfo(courseID:long) list of Section objects | |
| **Class implementing this method** | **Specification** |
| Course | ***Description:***  Returns a list containing all the sections of a course.  ***Input parameter(s):***  courseID:long representing the ID of the course to be consulted  ***Conditions for validity:***  courseID ≥ 0  ***Return type:***  list of Section objects |

|  |  |
| --- | --- |
| **Method** | |
| getCredits(courseID:long) int | |
| **Class implementing this method** | **Specification** |
| Course | ***Description:***  Returns the number of assigned credits of a course.  ***Input parameter(s):***  courseID:long representing the ID of the course  ***Conditions for validity:***  courseID ≥ 0  ***Return type:***  int |

|  |  |
| --- | --- |
| **Method** | |
| getSectionInfo(sectionID:long) Section | |
| **Class implementing this method** | **Specification** |
| Section | ***Description:***  Returns a Section object holding information about the section which has the corresponding section ID as the one passed as parameter.  ***Input parameter(s):***  sectionID:long representing the ID of the section to be consulted  ***Conditions for validity:***  sectionID ≥ 0  ***Return type:***  Section |

|  |  |
| --- | --- |
| **Method** | |
| getTimes(courseID:long, sectionID:long) list of int | |
| **Class implementing this method** | **Specification** |
| Course | ***Description:***  Returns a list of integer-formatted IDs of the time slots that a section of a course holds in a week.  ***Input parameter(s):***  courseID:long representing the ID of the course  sectionID:long representing the ID of the section  ***Conditions for validity:***  courseID ≥ 0  sectionID ≥ 0  ***Return type:***  list of int |

|  |  |
| --- | --- |
| **Method** | |
| getTimes(sectionID:long) list of int | |
| **Class implementing this method** | **Specification** |
| Section | ***Description:***  Returns a list of integer-formatted IDs of the time slots that a section holds in a week.  ***Input parameter(s):***  sectionID:long representing the ID of the section  ***Conditions for validity:***  sectionID ≥ 0  ***Return type:***  list of int |

|  |  |
| --- | --- |
| **Method** | |
| removeStudent(studentID:long) | |
| **Class implementing this method** | **Specification** |
| Course,  Section | ***Description:***  Removes a student from the list of registered student of a course.  ***Input parameter(s):***  studentID:long representing the ID of the student to be removed  ***Conditions for validity:***  studentID ≥ 0  ***Return type:***  void |