**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 that a user, either an administrator or a student, will use to interact with the system.

|  |  |
| --- | --- |
| **Method** | |
| disconnect() | |
| **Class implementing this method** | **Specification** |
| SavedUsers | ***Description:***  Disconnects a user from the system.  ***Input parameter(s):***  none  ***Return type:***  void |

|  |  |
| --- | --- |
| **Method** | |
| getSavedSchedule(scheduleID:long) Schedule | |
| **Class implementing this method** | **Specification** |
| Student | ***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** | |
| login(adminID:long, password:String) LogonPage | |
| **Class implementing this method** | **Specification** |
| Administrator | ***Description:***  Logs an administrator into the system after verification has been done.  ***Input parameter(s):***  adminID:long representing the administrator ID  password:String representing the password entered by the user  ***Conditions for validity:***  adminID ≥ 0  password is not empty  ***Return type:***  LogonPage |

|  |  |
| --- | --- |
| **Method** | |
| logout(adminID:long) LogoutPage | |
| **Class implementing this method** | **Specification** |
| Administrator | ***Description:***  Logs an administrator out of the system.  ***Input parameter(s):***  adminID:long representing the administrator ID  ***Conditions for validity:***  adminID ≥ 0  ***Return type:***  LogoutPage |

|  |  |
| --- | --- |
| **Method** | |
| requestLogout() | |
| **Class implementing this method** | **Specification** |
| SavedUsers | ***Description:***  Logs a user out of the system.  ***Input parameter(s):***  none  ***Return type:***  void |

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

|  |  |
| --- | --- |
| **Method** | |
| verify(adminID:long, password:String) boolean | |
| **Class implementing this method** | **Specification** |
| SavedUsers | ***Description:***  Verifies that an adminID matches the password associated with it. Returns true if the given information is valid, false otherwise.  ***Input parameter(s):***  adminID:long representing the ID of the administrator  password:String representing the password the administrator entered  ***Conditions for validity:***  adminID ≥ 0  password is not empty  ***Return type:***  boolean |

|  |  |
| --- | --- |
| **Method** | |
| viewAcademicRecord(studentID:long) AcademicRecord | |
| **Class implementing this method** | **Specification** |
| Student | ***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 |

**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** |
| Student,  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(startTime:int, endTime:int) | |
| **Class implementing this method** | **Specification** |
| Schedule | ***Description:***  Sets time slots in a scheduled as occupied given the starting and ending time of the time block to book.  ***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:***  0 ≤ startTime ≤ endTime  ***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** | |
| 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:***  0 ≤ startTime ≤ endTime  ***Return type:***  void |

|  |  |
| --- | --- |
| **Method** | |
| getAllSavedSchedules(studentID:long) list of Schedule objects | |
| **Class implementing this method** | **Specification** |
| Student,  SavedSchedules | ***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** | |
| getFreeCredits(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** | |
| 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** | |
| getWeeklyView(startDay:int, startMonth:int) Schedule | |
| **Class implementing this method** | **Specification** |
| Schedule | ***Description:***  Generates a weekly view of a schedule given the starting date.  ***Input parameter(s):***  startDay:int representing the day on which the week starts  startMonth:int representing the month in which the week is found  ***Conditions for validity:***  1 ≤ startDay ≤ 31  1 ≤ startMonth ≤ 12  ***Return type:***  Schedule |

|  |  |
| --- | --- |
| **Method** | |
| isBlockFree(startTime:int, endTime:int) boolean | |
| **Class implementing this method** | **Specification** |
| Schedule | ***Description:***  Verifies whether a time block defined by the given starting and ending time is free in a schedule.  ***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:***  0 ≤ startTime ≤ endTime  ***Return type:***  boolean |

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

|  |  |
| --- | --- |
| **Method** | |
| saveSchedule(scheduleID:long) list of Schedule objects | |
| **Class implementing this method** | **Specification** |
| SavedSchedule | ***Description:***  Saves a new schedule related to a student given the schedule 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  ***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 after a course removal given the ID of the course removed.  ***Input parameter(s):***  courseID:long representing the ID of the course removed  ***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** | |
| addToList(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** | |
| checkPassed(prereqs:Course, studentID:long) 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  studentID:long representing the ID of the student to verify  ***Conditions for validity:***  courseID ≥ 0  studentID ≥ 0  ***Return type:***  boolean |

|  |  |
| --- | --- |
| **Method** | |
| gatherCoursesPassed(studentID:long) list of Course objects | |
| **Class implementing this method** | **Specification** |
| CoursesPassed | ***Description:***  Retrieves and returns a list of courses that a student has successfully completed.  ***Input parameter(s):***  studentID:long representing the ID of the student  ***Conditions for validity:***  studentID ≥ 0  ***Return type:***  list of Course objects |

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

|  |  |
| --- | --- |
| **Method** | |
| getTimes(sectionID:long) list of int | |
| **Class implementing this method** | **Specification** |
| Section | ***Description:***  Returns a list of integers representing the starting and ending time of each time block that a section holds.  ***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** |
| 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 |