

## **COMP6030 Software Engineering Coursework**

**Group 15**

**TeamPart**

**We have provided you with access to the repository and Google Drive if you haven't received it please let us know via email.**

**Github link:**

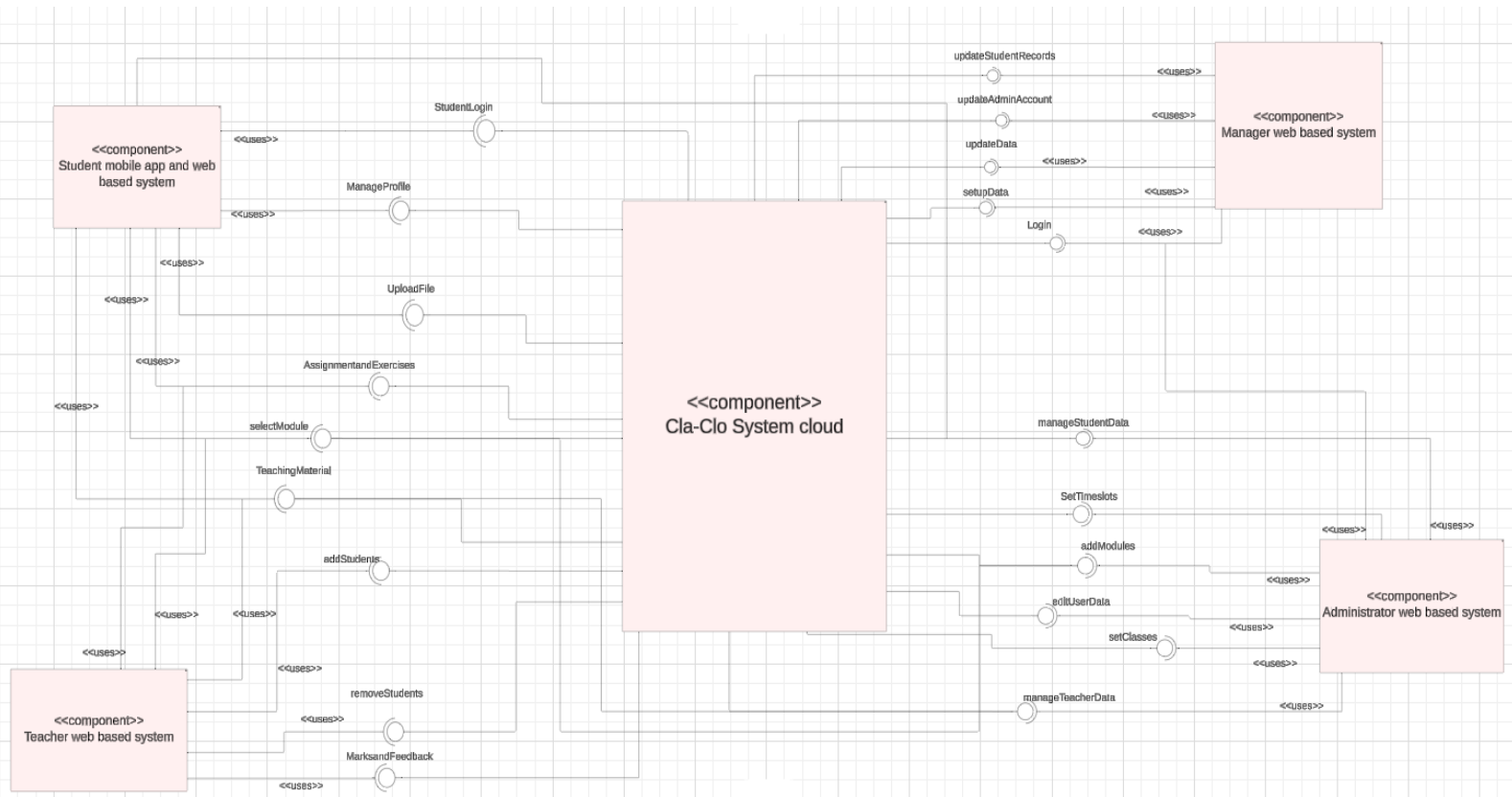
**[https://github.com/miche2400/Meetings\\_G15.git](https://github.com/miche2400/Meetings_G15.git)**

**Google drive link:**

**[https://drive.google.com/drive/folders/1Nb\\_NfZpQMRDUlyhy\\_6oHvbXb44CBLece?usp=drive\\_link](https://drive.google.com/drive/folders/1Nb_NfZpQMRDUlyhy_6oHvbXb44CBLece?usp=drive_link)**

## Architecture of wholeSystem - UML Component

For a clear view please us this link: [🔗 Lucidchart](#)



## Architecture of wholeSystem - UML Component- connected cloud

For clearer view please use the link: [🔗 Lucidchart](#)



## **Software architecture documentation**

### **Specification of components- (administrator)**

Components	Administrator GUI - web based system
Description	The microservices will be running on a desktop with a web-based interface to manage teachers and student data such as assigning modules, assigning classes, setting roles for teachers. User can also edit modules data and set weekly timeslots for both teachers and students
Type	Microservice
Required Interface	Login Edit userData Add Modules Remove Modules Add Programmes Edit Programmes Add TeachersData Remove TeachersData Add StudentData Remove StudentData Set Timetable
Provided interface	none

Components	Administrator data manager
Description	The component will be running on both web-based interface and on cloud based system to manage data such as edit user data and it also requires interface Check Authentications.
Type	Microservice
Required Interface	Check Permission
Provided interface	Edit userData

Components	Module manager
Description	The microservices will be running on Cloud with a web-based interface to manage Modules such as edit modules, remove modules, and add modules.
Type	Microservice
Required Interface	Upload changes GetSessioncode Check Permission
Provided interface	Get ModuleData Remove modules Add modules Update modules

Components	Programme manager
Description	The microservices will be running on Cloud with a to manage programmes such as addition and removal of programmes, and assigning modules that accept the programme.
Type	Microservice
Required Interface	Upload changes Assign modules GetSessioncode Get ModuleData Check permission
Provided interface	Add programmes Edit programmes

Components	Teacher manager
Description	The microservices will be running on a desktop with a web-based interface to manage the following tasks, such as assigning modules, Remove teacher, assigning classes and assigning roles to teacher.
Type	Micro services
Required Interface	Assign Modules Remove teacher Assign classes Assign roles
Provided interface	Manage teacher GetSessioncode

Components	Student manager
Description	The microservices will be running on a desktop with a web-based interface to manage student modules and classes.
Type	Micro services
Required Interface	Set modules Set Classes
Provided interface	Manage students GetSessioncode

Components	Timetable manager
Description	The microservices will be running on a desktop with a web-based interface to manage student weekly time slots and classes.
Type	Micro services
Required Interface	Manage Classes Weekly timeslots
Provided interface	Set timetable Getsessioncode

Components	Authentication manager
Description	The microservices will be running on a desktop with a Claclo cloud system to manage the authentication modules.
Type	Micro services
Required Interface	Login Check credentials
Provided interface	Check Auth

Components	Roles manager
Description	The microservices will be running on a desktop with a Claclo cloud system to manage the roles of each users.
Type	Micro services
Required Interface	Check Auth
Provided interface	Get roles

Components	Classes manager
Description	The microservices will be running on a desktop with a Claclo cloud system and web-bases interface to manage classes, required classes and get classes.
Type	Micro services
Required Interface	Get Modules data Retrieve classes
Provided interface	Required classes Get classes Manage classes

### Specifications of interfaces- (admin)

Name	Teacher manager	
Provider	Cloud Manager	
Operation: Set Modules	Signature	setModules(teacherID: String, modules: List<String>): Boolean
	Function	To set the modules to the respective teacher.
Operation : Assign classes	Signature	assignClasses(day: String, classDetails: Map<String, String>): Boolean
	Function	To assign classes for specific day.
Operation: Assign roles	Signature	Assign roles: assignRoles(teacherID: String, role: String): Boolean
	Function	To assign specific roles to teachers (e.g., department head, coordinator, etc.).

Name	Student manager	
Provider	Cloud Manager	
Operation: Set modules	Signature	setModules(studentID: String, modules: List<String>): Boolean
	Function	To assign modules to the respective student.
Operation: Set classes	Signature	setClasses(studentID: String, classes: List<String>): Boolean
	Function	To allocate class to the student with specific day and time.

Name	Timetable manager	
Provider	Cloud Manager	
Operation: Manage classes	Signature	manageClasses(classID: String, schedule: Map<String, String>): Boolean
	Function	To manage and update class schedules, including timings and assigned classrooms
Operation: Weekly Timeslots	Signature	setWeeklyTimeslots(classID: String, timeslots: List<Map<String, String>>): Boolean
	Function	To define and assign weekly timeslots for specific classes, ensuring no schedule conflicts.

Name	Programme manager	
Provider	Cloud Manager	
Operation: Assign Modules	Signature	assignModules(programmeID: String, modules: List<String>): Boolean
	Function	To assign specific modules to a given programme.
Operation: Upload changes	Signature	uploadChanges(programmeID: String, changes: Map<String, String>): Boolean
	Function	To upload updates or modifications made to the programme details (e.g., curriculum changes, module updates).
Operation: Add programme	Signature	addProgramme(programmeName: String, programmeDetails: Map<String, String>): Boolean
	Function	To create a new programme with its associated details (e.g., modules, schedule).
Operation: Edit programme	Signature	editProgramme(programmeID: String, updatedDetails: Map<String, String>): Boolean
	Function	To modify existing programme details, such as updating modules or schedule, for an already created programme



Name	Modules Manager	
Provider	Cloud Manager	
Operation: Add modules	Signature	addModules(moduleName: String, moduleDetails: Map<String, String>): Boolean
	Function	To add new modules to the system, including their details such as content and requirements.
Operation: Remove Modules	Signature	removeModules(moduleID: String): Boolean
	Function	To remove a specific module from the system, deleting its content and related information.
Operation: Upload Changes	Signature	uploadChanges(moduleID: String, changes: Map<String, String>): Boolean
	Function	To upload updates or modifications made to a specific module (e.g., content updates, changes in requirements).

Name	Administrator User manager	
Provider	Cloud Manager	
Operation: Admin registration	Signature	adminRegistration(adminID: String, adminDetails: Map<String, String>): Boolean
	Function	To register new administrators by adding their details to the system.
Operation: Edit user data	Signature	editUserData(userID: String, updatedData: Map<String, String>): Boolean
	Function	To edit or update user data (e.g., for teachers, students, or admins) in the system.

Name	Cloud Connection manager	
Provider	Cloud Manager	
Operation: Login	Signature	login(username: String, password: String): Boolean
	Function	To authenticate a user by verifying their credentials (username and password).
Operation: Getsessioncode	Signature	getSessionCode(username: String): String
	Function	To retrieve a session code for an authenticated user, typically for session management.

**Specification of databases:**

<b>Database Name</b>	<b>Student Database</b>
<b>Description</b>	Stores information about students, including their personal details, enrollment status, courses, and grades.
<b>Type</b>	Relational Database
<b>Design (Tables)</b>	Student, Enrollment, Grades
<b>Student</b>	Student_Id, First_Name, Last_Name, Date_of_Birth, Email, Phone_Number, Address, Enrollment_Status
<b>Enrollment</b>	Enrollment_Id, Student_Id, Programme_Id, Enrollment_Date, Status
<b>Grades</b>	Grade_Id, Student_Id, Module_Id, Grade, Date_Assigned

<b>Database Name</b>	<b>Administrator Database (User Database)</b>
<b>Description</b>	Stores information about the system's administrators, their credentials, roles, and privileges.
<b>Type</b>	Relational Database
<b>Design (Tables)</b>	Admin, User_Roles, Admin_Actions
<b>Admin</b>	Admin_Id, First_Name, Last_Name, Email, Password, Phone_Number, Role
<b>User_Roles</b>	Role_Id, Admin_Id, Role_Name, Permissions
<b>Admin_Actions</b>	Action_Id, Admin_Id, Action_Type, Date_Performed, Description

<b>Database Name</b>	<b>Teacher Database</b>
<b>Description</b>	Stores information about teachers, including their personal details, courses taught, and qualifications.
<b>Type</b>	Relational Database
<b>Design (Tables)</b>	Teacher, Course_Assignments, Qualifications
<b>Teacher</b>	Teacher_Id, First_Name, Last_Name, Email, Phone_Number, Date_of_Birth, Department
<b>Course_Assignments</b>	Assignment_Id, Teacher_Id, Course_Id, Assignment_Date, Status
<b>Qualifications</b>	Qualification_Id, Teacher_Id, Degree, Institution, Year_Graduated

<b>Database Name</b>	<b>Programmes Database</b>
<b>Description</b>	Stores information about academic programmes, their modules, and related details.
<b>Type</b>	Relational Database
<b>Design (Tables)</b>	Programme, Modules, Programme_Curriculum
<b>Programme_Curriculum</b>	Curriculum_Id, Programme_Id, Module_Id, Sequence_Order, Semester
<b>Programme</b>	Programme_Id, Programme_Name, Programme_Description, Start_Date, End_Date
<b>Modules</b>	Module_Id, Module_Name, Module_Description, Credits, studylevel

<b>Database Name</b>	<b>Module Database</b>
<b>Description</b>	Stores information about modules, including their details, contents, and associated programmes.
<b>Type</b>	Relational Database
<b>Design (Tables)</b>	Module, Module_Content, Module_Programme_Link
<b>Module</b>	Module_Id, Module_Name, Module_Description, Credits, Duration, studylevel
<b>Module_Content</b>	Content_Id, Module_Id, Content_Type (Lecture, Assignment, etc.), Content_Details, Date_Added
<b>Module_Programme_Link</b>	Link_Id, Module_Id, Programme_Id

### **Specification of components**

Component	Teacher Web-Based System
Description	This microservice provides a user interface for teachers to log in, manage exercises, assignments, materials, and student data, and access marking and feedback functionalities.
Type	Microservice
Required Interface	Authentication Exercise Manager Class Manager Material Manager Feedback Manager Assessment Manager
Provided interface	None

Component	Authentication
Description	This microservice runs on the cloud and handles secure authentication for teachers. It verifies credentials, grants access tokens for secure API requests, and manages password reset functionalities
Type	Microservice
Required Interface	User Data Manager (to fetch and validate stored teacher data)
Provided interface	Teacher Login Reset password

Component	Exercise Manager
Description	This microservice is responsible for creating, modifying, and deleting exercises assigned to students. It validates the exercise parameters (e.g., type, passing score, deadlines) and ensures proper integration with class and assessment modules.
Type	Microservice
Required Interface	Class manager(to ensure exercises are tied to existing classes). AssessmentManager (to manage assessments linked to exercises).
Provided interface	Manage exercises

Component	Class manager
Description	Handles operations related to student and class data management. This microservice supports adding/removing students to/from classes, retrieving class rosters, and linking students with coursework and assessments.
Type	Microservice
Required Interface	None
Provided interface	Manageclasses

Component	Feedback manager
Description	Provides teachers with tools to review submissions, assign marks, and give feedback. It supports multiple grading formats, such as letter grades, percentages, or pass/fail.
Type	Microservice
Required Interface	AssessmentManager (to link feedback to specific assessments). UserDataAccess (to retrieve student data).
Provided interface	Review submission

Component	Assessment manager
Description	This microservice oversees the setup, modification, and monitoring of assessments. It integrates with exercises, manages assessment deadlines, and ensures compatibility with class data.
Type	Microservice
Required Interface	ExerciseManager(to link exercises to assessments)
Provided interface	Manage assessments

Component	User data Manager
Description	This microservice provides centralized access to teacher and student data, ensuring secure retrieval and storage. It supports authentication, class management, and student-related processes.
Type	Microservice
Required Interface	None
Provided interface	User data access

### **Specification of interface**

Name	ManageExercises	
Provider	Exercise manager	
Operation name: createExercise	Signature	createExercise(exerciseName: String, type: String, passingScore: Int, deadline: String, classID: String): Boolean
	Function	To create and assign exercises to a class with a specific deadline and passing score.
Operation name: getExerciseDetails	Signature	getExerciseDetails(exerciseID: String): ExerciseDetails
	Function	To retrieve exercise information such as name, type, deadline, and status.

Name	ManageAssessments	
Provider	Exercise manager	
Operation name: createAssessment	Signature	createAssessment(assessmentName: String, type: String, totalMarks: Int, passingScore: Int, classID: String, deadline: String): Boolean
	Function	To define new assessments for students and assign them to classes.
Operation name: deleteAssessment	Signature	deleteAssessment(assessmentID: String): Boolean
	Function	To delete an existing assessment..

Name	ManageTeachingMaterials	
Provider	Teaching Material Manager	
Operation name: uploadMaterial	Signature	uploadMaterial(fileName: String, fileType: String, fileSize: Float, contentType: String, classID: String): MaterialResponse
	Function	Upload teaching materials linked to a specific class.
Operation name: deleteMaterial	Signature	deleteMaterial(materialID: String): Boolean
	Function	Remove teaching material.

Name	MarkSubmissions	
Provider	Student work manager	
Operation name: markSubmission	Signature	markSubmission(submissionID: String, marks: Int, feedbackText: String): Boolean
	Function	To mark student submissions, assign grades, and provide feedback.
Operation name: getMarks	Signature	getMarks(studentID: String, exerciseID: String): MarksDetails
	Function	Retrieve the marks assigned to a student for a specific exercise.



Name	ManageClasses	
Provider	Classmanager	
Operation name: addStudentToClass	Signature	addStudentToClass(studentID: String, classID: String): Boolean
	Function	Add a student to a class.
Operation name: removeStudentFromClass	Signature	removeStudentFromClass(studentID: String, classID: String): Boolean
	Function	Remove a student from a class

Name	ManageFeedback	
Provider	Feedback manager	
Operation name: addFeedback	Signature	addFeedback(studentID: String, exerciseID: String, marksObtained: Int, feedbackText: String): Boolean
	Function	To add feedback for a student's work.
Operation name: getFeedback	Signature	getFeedback(studentID: String, exerciseID: String): FeedbackDetails
	Function	Retrieve feedback provided for a student's exercise.

## Specification of components

<b>Component</b>	Student Web based system
<b>Description</b>	This microservice will run on a mobile app to enable the student to access their profile and perform activities, like enrollments, working on assignments, managing their profile and looking at teaching materials or feedback through their web system.
<b>Type</b>	microservice
<b>Required interface</b>	EnrollModule, QuitModule, EnrollClass, AccessAssignment, SubmitAssignment, ViewFeedback, DownloadMaterial, UpdateProfile, ViewProfile
<b>Provided interface</b>	SubmitQueries, SubmitOfflineAssignments

<b>Component</b>	Student Mobile app
<b>Description</b>	This microservice will run on a mobile app to enable the student to access their profile and perform activities, like enrollments, working on assignments, managing their profile and looking at teaching materials or feedback through their app.
<b>Type</b>	microservice
<b>Required interface</b>	EnrollModule, QuitModule, EnrollClass, AccessAssignment, SubmitAssignment, ViewFeedback, DownloadMaterial, UpdateProfile, ViewProfile
<b>Provided interface</b>	SubmitQueries, SubmitOfflineAssignments

<b>Component</b>	Profile Manager
<b>Description</b>	This microservice allows the students to view, update and download their academic profile for printing or sharing
<b>Type</b>	microservice
<b>Required interface</b>	UpdateProfile, DownloadProfile
<b>Provided interface</b>	ProvideProfileDetails

<b>Component</b>	Module Manager
<b>Description</b>	This microservice allows the student to manage their modules by enrolling, withdrawing, accessing classes and assignments
<b>Type</b>	microservice
<b>Required interface</b>	EnrollModule, QuitModule, GetModuleDetails
<b>Provided interface</b>	ViewModules

<b>Component</b>	Class Manager
<b>Description</b>	This microservice allows the student to enroll into a specific class within a module if many classes are available}, and access all of that class related activities
<b>Type</b>	microservice
<b>Required interface</b>	EnrollClass, QuitClass, GetClassSchedule
<b>Provided interface</b>	ViewClassDetails

<b>Component</b>	Teaching Materials Manager
<b>Description</b>	This microservice allows the student to view or download all the teaching materials that were uploaded by the teachers.
<b>Type</b>	microservice
<b>Required interface</b>	GetTeachingMaterials, DownloadMaterial
<b>Provided interface</b>	ProvideMaterialLinks

<b>Component</b>	Assessment Manager
<b>Description</b>	This microservice allows the student to handle all of the assessment/homework related activities, such as online or offline working, submissions, grades and feedback
<b>Type</b>	microservice
<b>Required interface</b>	GetAssignments, WorkOnline, WorkOffline, SubmitAssignment, ViewFeedback
<b>Provided interface</b>	SubmitProgress, UploadFiles

### **Specifications of interfaces**

Name	login	
Provider	Authentication Manager	
Operation name: login	Signature	studentLogin(userID: String, password: String): Boolean
	Function	Checks student credentials and returns true or false if they match with the database saved details

Name	ManageProfile	
Provider	StudentProfile Manager	
Operation name: updateProfile	Signature	updateProfile(studentID: String, profileData: JSON): Boolean
	Function	Allows the student to update their profile based on recent changes related to academic studies
Operation name: viewProfile	Signature	Signature: viewProfile(studentID: String)
	Function	Allows the student to view their profile

Name	SelectModule	
Provider	Module Manager	
Operation name: selectClass	Signature	selectClass(studentID:String, classID:String)
	Function	Allows the student to select one of the available classes on that specific module
Operation name: chooseAssignment	Signature	chooseAssignment(classID:String, assignmentID:String)
	Function	Allows the student to select any of the available assignments

Name	SelectAssignment	
Provider	Assignment Manager	
Operation name: submitOnlineWork	Signature	submitOnlineWork(studentID:string, assignmentID:string, savedFiles:JSON)
	Function	Allows the student to submit their online work straight away
Operation name: getAssignment	Signature	getAssignment(classID: String): List<Assignment>
	Function	Gets all the available assignments for that specific class and displays them as a list for the student

Name	uploadFile	
Provider	FileManager	
Operation name: submitOfflineWork	Signature	submitOfflineWork(assignmentID: String, file: File): Boolean
	Function	Allows the student to submit their offline work
Operation name: validateSubmission	Signature	validateSubmission(file: File): Boolean
	Function	Checks if the submission format is right

Name	AccessTeachingMaterials	
Provider	TeachingMaterial Manager	
Operation name: fetchMaterials	Signature	fetchMaterials(moduleID: String): List<Material>
	Function	Allows the student to see the teaching materials
Operation name: downloadMaterials	Signature	downloadMaterials(materialID: String): File
	Function	Allows the student to download the available teaching materials

### Specification of components

<b>Component</b>	Web Platform System - ClaClo-Manager
<b>Description</b>	This microservice provides a user interface for managers to log in, set up and update the department administrators' account, the university account, and students' records, as well as delete and archive administrative staff records
<b>Type</b>	Microservice
<b>Required Interface</b>	Login, setUp/Update,view, DeleteAndArchive
<b>Provided Interface</b>	none

<b>Component</b>	Update Administrators Accounts
<b>Description</b>	This microservice provides an update on the administrator's accounts
<b>Type</b>	Microservice
<b>Required Interface</b>	none
<b>Provided Interface</b>	updateAccounts, checkCredentials,Set Up/Update

<b>Component</b>	Update University Account
<b>Description</b>	This microservice provides an update in either the degree Programme or the University Department
<b>Type</b>	Microservice
<b>Required Interface</b>	none
<b>Provided Interface</b>	checkCredentials,updateData,Set Up/Update,updateAccount

<b>Component</b>	Update Student Records
<b>Description</b>	This microservice provides an update on the student record
<b>Type</b>	Microservice
<b>Required Interface</b>	updateData
<b>Provided Interface</b>	updateStudentRecords, checkCredentials,Set Up/Update,view

<b>Component</b>	Update Administrative staff records
<b>Description</b>	This microservice provides an update in the Administrative staff record
<b>Type</b>	Microservice
<b>Required Interface</b>	none
<b>Provided Interface</b>	updateRecords, checkCredentials, DeleteAndArchive,Set Up/Update

<b>Component</b>	Authentication
<b>Description</b>	This microservice runs on the cloud and handles secure authentication for Managers. It verifies credentials, grants access tokens for secure API requests, and manages password reset functionalities
<b>Type</b>	Microservice
<b>Required Interface</b>	checkCredentials
<b>Provided Interface</b>	Login

### **Specification of interface**

Name	Update Accounts	
Provider	Admin User Manager	
Operation name:	Signature	updateAccount(accountID: String, accountData: JSON): Boolean
	Function	Allows the manager to update the accounts for admins

Name	Login	
Provider	Authentication Manager	
Operation name: Login	Signature	isLoggedIn()
	Function	Returns true if the login was successful and False if not.

Name	Update Student Records	
Provider	Student Manager	
Operation name: updateStudentRecords	Signature	updateStudentRecord(studentID: String, recordData: JSON): Boolean
	Function	Allows to update student records



Name	Update Administrative Records	
Provider	Admin User Manager	
Operation name: updateRecords	Signature	updateAdministrativeRecord(staffID: String, staffData: JSON): Boolean
	Function	Allows to update admin records such as job roles or contact information.

Name	View Updated Student Records	
Provider	Update Student Records	
Operation name: View	Signature	viewStudentRecords()
	Function	Allow to view the traces of student record changes