Covid Manager ICD

1 Covid Manager ICD

Identifier	Covid Manager Server
Functionality	Defines methods to provide functionality related with Client's Presenter - Server's Model interconnection.

1.1 Covid Manager Server

The **Covid Manager Server RMI** interface exposes the following members:

Properties

Property Type	Name	Description
-	-	-

Note: A property provides a mechanism to get or set the value of a private field (class data member) in Java

Methods

Return Type	Name	Description
ArrayList<Student>	fetchStudents()	Retrieves all students from the database.
Student	fetchStudent(int studentId)	Retrieves a student from the database.
int	login(String userName, String password)	Compares given username and password to usernames and passwords already in the database and gives access if valid.
String	totalCovidCases()	Returns the total number of covid cases reported on the system.
String	currentCovidCases()	Returns the number of students who are currently a covid case.
String	totalStudents()	Returns the total number of students in the database.
Student	addStudent (Student student)	Creates a document in the database and fills it with a Student object and a unique id number.
void	deleteStudent (Student student)	Deletes a student from the database based on the id number.
ArrayList<Student>	getStudent()	Returns all students from the database.
Schedule	fetchSchedule()	Returns the schedule from the database.
void	saveSchedule(Schedule schedule)	Saves the schedule in the database.

Return Type	Name	Description
ArrayList<Student>	fetchStudents()	Retrieves all students from the database.
Student	fetchStudent(int studentId)	Retrieves a student from the database.
String	potentiallyExposed()	Returns the students that were potentially exposed to the disease at some point.
void	updateStudentCourses(int studentId, List<Schedule.CoursesEnum> courses)	Updates the Courses list field.
List<Integer>	lastCovidCases()	Creates a list with covid cases in the last two weeks.
void	addCovidCase(int studentId, LocalDate caseDate)	Updates the Covid Case field

Note: A method is the equivalent to a function

Properties

None

1.1.1 Methods

1.1.1.1 login

Description

Compares the username and password given to every username and password in the database and allows access if it is found in the database.

If the credentials given match those of a student, it gives access as a student. If it matches those of a secretary, it gives access as a secretary

Syntax

int Login(
String userName,
String password)

Parameters

String userName

The username with which one wants to gain access to the program.

String password

The password with which one wants to gain access to the program.

Return value

Integer

If the function succeeds it returns 1 for secretary or 2 for student. Otherwise returns 0.

Throws

RemoteException

If something goes wrong during the remote connection (if the connection is refused to remote host or if an I/O exception is thrown during the remote connection).

1.1.1.2 totalCovidCases

Description

Sums all the cases ever reported involving the students in the program.

Syntax

String totalCovidCases()

Parameters

No parameters

Return value

String

The total number of covid cases involving students in the database.

Throws

RemoteException

If something goes wrong during the remote connection (if the connection is refused to remote host or if an I/O exception is thrown during the remote connection).

1.1.1.3 curentCovidCases

Description

Sums all the reported cases while subtracting the number of students who recovered

Syntax

String currentCovidCases()

Parameters

No parameters.

Return value

String

The current number of covid cases involving students in the database.

Throws

RemoteException

If something goes wrong during the remote connection (if the connection is refused to remote host or if an I/O exception is thrown during the remote connection).

1.1.1.4 totalStudents

Description

Sums all the Student objects created and currently in the database.

Syntax

String totalStudents()

Parameters

No parameters

Return value

String

The total number of students in the database.

Throws

RemoteException

If something goes wrong during the remote connection (if the connection is refused to remote host or if an I/O exception is thrown during the remote connection).

1.1.1.5 addStudent

Description

Creates a new document in the database and fills it with a student object as well as adding a unique id number.

Syntax

ArrayList<Student> addStudent(Student student)

Parameters

Student

An object that contains the information of the student that we want to add to the database.

Return value

void

Throws

RemoteException

If something goes wrong during the remote connection (if the connection is refused to remote host or if an I/O exception is thrown during the remote connection).

1.1.1.6 deleteStudent

Description

Deletes the information on the student from the database based on the id number of the student

Syntax

void *deleteStudent(Student student)*

Parameters

Student student

An object that contains the information of the student that we want to delete from the database.

Return value

void

Throws

RemoteException

If something goes wrong during the remote connection (if the connection is refused to remote host or if an I/O exception is thrown during the remote connection).

1.1.1.7 fetchStudents

Description

Retrieves all the students in the database and returns them in a list.

Syntax

ArrayList<Student> fetchStudents()

Parameter

void

Return value

ArrayList<Student>

A list with the information of all the students currently in the database.

Throws

RemoteException

If something goes wrong during the remote connection (if the connection is refused to remote host or if an I/O exception is thrown during the remote connection).

1.1.1.8 potentiallyExposed()

Description

Returns the students who could have been exposed to the virus.

Syntax

String potentiallyExposed()

Parameter

void

Return value

String

The number of students that are potentially affected.

Throws

RemoteException

If something goes wrong during the remote connection (if the connection is refused to remote host or if an I/O exception is thrown during the remote connection).

1.1.1.9 fetchShedule

Description

Finds the schedule using the id of the student and returns it.

Syntax

Schedule fetchSchedule()

Parameter

void

Return value

Schedule

The schedule of the student.

Throws

RemoteException

If something goes wrong during the remote connection (if the connection is refused to remote host or if an I/O exception is thrown during the remote connection).

1.1.1.10 saveSchedule

Description

Takes a schedule given by a user and saves it in the database.

Syntax

void saveSchedule(Schedule schedule)

Parameter

Schedule schedule

The schedule that is given by the user.

Return value

void

Throws

RemoteException

If something goes wrong during the remote connection (if the connection is refused to remote host or if an I/O exception is thrown during the remote connection).

1.1.1.11 fetchStudent

Description

Retrieves one student from the database using their id.

Syntax

Student fetchStudent(int studentId)

Parameter

void

Return value

Student

The information of the student whose id was used.

Throws

RemoteException

If something goes wrong during the remote connection (if the connection is refused to remote host or if an I/O exception is thrown during the remote connection).

1.1.1.12 updateStudentCourses

Description

Updates the list of courses that will be available on the schedule list.

Syntax

updateStudentCourses(int studentId, List<Schedule.CoursesEnum> courses)

Parameter

int studentId

The id of the student whose courses the user wants updated.

List<Schedule.CoursesEnum> courses

The new set of courses the user wants set.

Return value

void

Throws

RemoteException

If something goes wrong during the remote connection (if the connection is refused to remote host or if an I/O exception is thrown during the remote connection).

1.1.1.13 lastCovidCases

Description

Searches for all the covid cases reported during the last fourteen days (The value assigned to the QUARANTINE_DAYS field).

Syntax

List<Integer> lastCovidCases()

Parameter

void

Return value

List<Integer>

A list containing covid cases in the last fourteen days.

Throws

RemoteException

If something goes wrong during the remote connection (if the connection is refused to remote host or if an I/O exception is thrown during the remote connection).

1.1.1.14 addCovidCase

Description

Updates a student document in the database by adding the date of the confirmed covid case.

Syntax

addCovidCase(int studentId, LocalDate caseDate)

Parameter

int

The id of the student that is a confirmed covid case.

LocalDate

The date of the confirmed case.

Return value

List<Integer>

A list containing covid cases in the last fourteen days.

Throws

RemoteException

If something goes wrong during the remote connection (if the connection is refused to remote host or if an I/O exception is thrown during the remote connection).

1.2 Student class

Identifier	**Student**
Functionality	Defines methods that allow the creation and handling of a Student object

The **Student class** interface exposes the following members:

Properties

Property Type	Name	Description
ObjectId	getId()	Returns the value of the id field.
void	setId(int idNumber)	Sets the value of the id field.
String	getPassword()	Returns an encrypted version (based on the Base64 encryption) of the given password.
void	setPassword (String password)	Sets the value of the Password field.
int	getIdNumber()	Returns the value of the idNumber field.
void	setIdNumber	Sets the value of the idNumber field.
int	getCovidProbability()	Returns the value of the CovidProbability field.
void	setCovidProbability(int covidProbability)	Sets the value of the CovidProbability field.
int	getFirstName()	Returns the value of the FirstName field.
void	setFirstName (String firstName)	Sets the value of the FirstName field.
String	getLastName()	Returns the value of the LastName field.
void	setLastName (String lastName)	Sets the value of the LastName field.
String	getBirthday()	Returns the value of the Birthday field.
void	setBirthday(LocalDate birthday)	Sets the value of the Birthday field in the form of “DD-MM-YYYY”.
LocalDate	getCovidCase()	Returns the value of the CovidCase
void	setCovidCase(LocalDate covidCase)	Sets the value of the CovidCase field in “DD-MM-YYYY” format.
List<Courses>	getCourses()	Returns a List with all the courses that the Student.
void	setCourses(List<Schedule.CoursesEnum> courses)	Sets the value of the Courses field.
double	getCovidProbabilityTmp()	Returns the value of the CovidProbabilityTmp field.
void	setCovidProbabilityTmp(double covidProbabilityTmp)	Sets the value of the CovidProbabilityTmp field.

Note: A property provides a mechanism to get or set the value of a private field (class data member) in Java

Methods

Return Type	Name	Description
boolean	equals(Object obj)	Compares the id number of the given object to the id number of the specific student
String	decPassword()	Decrypts the students password.
int	daysUntilRecovery	Finds the days it took for the student to recover
-	Student()	A constructor that initializes a Student object with no parameters
-	Student(String firstName, String lastName, LocalDate birthday, int idNumber)	A constructor that initializes a Student object without the CovidCase field.
-	public Student(ObjectId id, String firstName, String lastName, LocalDate birthday, LocalDate covidCase, int idNumber, String password, List<Courses> courses)	A constructor that initializes a Student object filling all the field.

Note: A method is the equivalent to a function

Properties

1.2.1.1 getId

Description

Returns the unique id of a student

Syntax

ObjectId getId()

Parameter/return value

Return value: Object Id

The unique id of the student object calling the method

1.2.1.2 setId

Description

Sets the value of the id field and locks the given value

Syntax

void setId(final ObjectId id)

Parameter/return value

Parameter: final Object Id

Id field set as final making it unable to be changed later.

1.2.1.3 getPassword

Description

Returns the encrypted version of the password of the student object calling the method.

Syntax

String getPassword()

Parameter/return value

Return value: password

The password field set for this Student

1.2.1.4 setPassword

Description

Sets the value of the password field with the given password without an encryption.

Syntax

void *setPassword(String password)*

Parameter/return value

Parameter: password

A string of any size given as the student’s password.

1.2.1.5 getIdNumber

Description

Returns the value of the idNumber field.

Syntax

int getIdNumber()

Parameter/return value

Return value: idNumber

An integer id for the Student object.

1.2.1.6 setIdNumber

Description

Sets the value of the idNumber field with the given idNumber.

Syntax

void setIdNumber(int idNumber)

Parameter/return value

Parameter: idNumber

An integer id for the Student object.

1.2.1.7 getCovidProbability

Description

Gets the value of the CovidProbability field.

Syntax

int getCovidProbability()

Parameter/return value

Return Value: covidProbability

An integer showing the probability of the Student having covid.

1.2.1.8 setCovidProbability

Description

Sets the value of the CovidProbability field.

Syntax

void setCovidProbability(int covidProbability)

Parameter/return value

Parameter: covidProbability

An integer showing the probability of the Student having covid.

1.2.1.9 getFirstName

Description

Gets the value of the FirstName field.

Syntax

String getFirstName()

Parameter/return value

Return Value: firstName

A String containing the first name of the student.

1.2.1.10 setFirstName

Description

Sets the value of the FirstName field.

Syntax

void setFirstName()

Parameter/return value

Parameter: firstName

A String containing the first name of the student.

1.2.1.11 getLastName

Description

Gets the value of the LastName field.

Syntax

String getLastName()

Parameter/return value

Return value: lastName

A String containing the last name of the student.

1.2.1.12 setLastName

Description

Sets the value of the LastName field.

Syntax

void setLastName(String lastName)

Parameter/return value

Parameter: lastName

A String containing the last name of the student.

1.2.1.13 getBirthday

Description

Gets the value of the Birthday field.

Syntax

LocalDate getBirthday()

Parameter/return value

Return value: birthday

A Local Date field containing the birthday of the student .

1.2.1.14 setBirthday

Description

Sets the value of the Birthday field.

Syntax

void setBirthday(LocalDate birthday)

Parameter/return value

Parameter: birthday

A Local Date field containing the birthday of the student.

1.2.1.15 getCovidCase

Description

Gets the value of the covidCase field.

Syntax

LocalDate getCovidCase()

Parameter/return value

Return value: covidCase

A Local Date field containing the day the student got infected.

1.2.1.16 setCovidCase

Description

Sets the value of the covidCase field.

Syntax

LocalDate setCovidCase()

Parameter/return value

Parameter: covidCase

A Local Date field containing the day the student got infected.

1.2.1.17 getCourses

Description

Gets the value of the Courses field.

Syntax

List<Courses> getCourses()

Parameter/return value

Return value: courses

A List<Courses> containing the courses that are selected for the student.

1.2.1.18 setCourses

Description

Sets the value of the Courses field.

Syntax

void setCourses(List<Courses> courses)

Parameter/return value

Parameter: courses

A List<Courses> containing the courses that are selected for the student.

1.2.1.19 getCovidProbabilityTmp

Description

Gets the value of the CovidProbabilityTmp field.

Syntax

getCovidProbabilityTmp()

Parameter/return value

Return Value: CovidProbabilityTmp

A double field that contains the covid probability.

1.2.1.20 setCovidProbabilityTmp

Description

Sets the value of the CovidProbabilityTmp field.

Syntax

setCovidProbabilityTmp(double covidProbabilityTmp)

Parameter/return value

Parameter: CovidProbabilityTmp

A double field that contains the covid probability.

1.2.2 Methods

1.2.2.1 equals

Description

Confirms that the Object given is a Student object. If it is, it compares the idNumber of the object to the idNumber of the Student object it was called on and returns true if they match and false if they don’t match. Also returns false if the Object is not a Student object

Syntax

boolean equals(Object obj)

Parameters

Object obj

The object that the user wants to compare with the Student

Return value

Return false if the object is not a Student object or if it is but the ids are not equal, else returns true.

1.2.2.2 decPassword

Description

Returns the original String that was used for the encryption in the *encPasswordAndSet(String password)* method using the decoder method found in the Base64 class.

Syntax

String decPassword()

Parameters

none

Return value

The decoded password submitted during the creation of the Student object.

1.2.2.3 daysUntilRecovery

Description

Calculates the time period between the confirmation of a covid case and the current date to find the days it takes to recover.

Syntax

Int daysUntilRecovery()

Parameters

none

Return value

The days between confirmed case and recovery.

1.2.2.4 Student

Description

Creates a Student object filling all the fields with their default values.

Syntax

Student()

Parameters

none

Return value

none

1.2.2.5 Student

Description

Creates a Student object filling some of the fields with values given by the user (the password field is filled with a basic 12345 password and the courses List and CovidProbability field are left with their default value).

Syntax

Student(
String firstName,
String lastName,
LocalDate birthday,
int idNumber)

Parameters

String firstName

The first name of the student we are adding to the database.

String *lastName*

The last name of the student we are adding to the database.

LocalDate birthday

The birthday of the student.

int idNumber

The idNumber of the student.

Return value

none

1.2.2.6 Student

Description

Creates a Student object for a student that is a covid case filling all the fields with values given by the user while also adding the CovidCase date.

Syntax

```
Student(
        ObjectId id,
        String firstName,
        String lastName,
        LocalDate birthday,
        LocalDate covidCase,
        int idNumber,
        String password,
        List<Schedule.CoursesEnum> courses)
```

Parameters

ObjectId id,

The unique id of the student object being created.

String firstName

The first name of the student we are adding to the database.

String *lastName*

The last name of the student we are adding to the database.

LocalDate birthday

The birthday of the student.

LocalDate covidCase,

The date of the confirmed covid case.

int idNumber

String password

The password with which the student can gain access to the program (it goes through a simple encryption before being set as the value of the password field)

List<Schedule.CoursesEnum> courses

A list with the courses the student will be participating in.

Return value

none

1.3 Schedule class

The **Courses class** interface exposes the following members:

Properties

Property Type	Name	Description
String	getId()	Returns the value of the id field.
void	setId(final String id)	Sets the value of the id field.
HashMap<String, List<DayOfWeek>>	getScheduleMap()	Returns the value of the ScheduleMap field.
void	setScheduleMap(HashMap<String, List<DayOfWeek>> scheduleMap)	Sets the value of the ScheduleMap field

Note: A property provides a mechanism to get or set the value of a private field (class data member) in Java

Methods

Return Type	Name	Description
-	Schedule()	A constructor that initializes a Schedule object.

Note: A method is the equivalent to a function

Remarks

None.

1.3.1 Properties

1.3.1.1 getId

Description

Gets the value of the id field

Syntax

String getId()

Parameter/return value

Return Value: String Id

The id of the Student whose schedule the user is currently seeing.

1.3.1.2 setId

Description

Sets the value of the id field

Syntax

void setId(final String id)

Parameter/return value

Parameter: final String Id

The id of the student whose schedule the user is currently seeing. It’s set as final to prevent the value of the id from being changed.

1.3.1.3 getScheduleMap

Description

Gets the value of the ScheduleMap field

Syntax

HashMap<String, List<DayOfWeek>> getScheduleMap()

Parameter/return value

Return Value: HashMap<String, List<DayOfWeek>> ScheduleMap

The Map containing the student’s schedule, in days and courses in each day (if this method is called before the setScheduleMap method then the Map will contain every course in the CoursesEnum field in every available day).

1.3.1.4 setScheduleMap

Description

Sets the value of the ScheduleMap field

Syntax

void setScheduleMap (HashMap<String, List<DayOfWeek>> scheduleMap)

Parameter/return value

Parameter: HashMap<String, List<DayOfWeek>> ScheduleMap

The Map containing the student’s schedule, in days and courses in each day.

1.3.2 Methods

1.3.2.1 Schedule

Description

Initializes a Schedule object, fills the Map with all the courses in the CoursesEnum field and fills the id field with the default String value.

Syntax

Schedule()

Parameters

none

Return value

none