School Master Restful Web API

Table of Contents

[Overview 2](#_Toc65859667)

[Phase I 2](#_Toc65859668)

[Phase II 2](#_Toc65859669)

[Use Cases – Phase I 2](#_Toc65859670)

[Administrator 2](#_Toc65859671)

[Instructor 3](#_Toc65859672)

[Student 3](#_Toc65859673)

[Guardian 3](#_Toc65859674)

[Class Diagram 4](#_Toc65859675)

[HTTP Endpoints 4](#_Toc65859676)

[Authenticate 5](#_Toc65859677)

[Administrator 5](#_Toc65859678)

[POST 5](#_Toc65859679)

[PUT 5](#_Toc65859680)

[GET (By Id) 5](#_Toc65859681)

[GET (By Login) 7](#_Toc65859682)

[DELETE 9](#_Toc65859683)

[Instructor 9](#_Toc65859684)

[POST 9](#_Toc65859685)

[PUT 9](#_Toc65859686)

[GET 9](#_Toc65859687)

[DELETE 9](#_Toc65859688)

[Student 9](#_Toc65859689)

[POST 9](#_Toc65859690)

[PUT 9](#_Toc65859691)

[GET 9](#_Toc65859692)

[DELETE 9](#_Toc65859693)

[Guardian 9](#_Toc65859694)

[POST 9](#_Toc65859695)

[PUT 9](#_Toc65859696)

[GET 10](#_Toc65859697)

[DELETE 10](#_Toc65859698)

[Data Store 10](#_Toc65859699)

[Person 11](#_Toc65859700)

[Role 11](#_Toc65859701)

[PersonRoleMap 11](#_Toc65859702)

[Address 11](#_Toc65859703)

[Email 12](#_Toc65859704)

[Phone 12](#_Toc65859705)

[Administrator 12](#_Toc65859706)

[Instructor 12](#_Toc65859707)

[Student 12](#_Toc65859708)

[Guardian 13](#_Toc65859709)

[StudentGuardianMap 13](#_Toc65859710)

[Useful Links 13](#_Toc65859711)

# Overview

The School Master restful web API is used to implement support for managing school personal, students, and student guardians as well as managing classes, schedules, assignments, and grading.

## Phase I

Phase I will include the implementation of the web API support for managing school personal, students, and student guardians.

## Phase II

Phase II will include the implementation of the web API support for managing classes, schedules, assignments, and grading.

# Use Cases – Phase I

## Administrator

Administrators have the ability to access and modify all aspects of the system.

When the web API is installed a default administrator account will be created in the database.

Administrators will be able to create, access, update, and delete Administrators, Instructors, Students, and Guardians. This will include being able to reset passwords for any of these users.

## Instructor

Instructors will be able to update their personal information and reset their passwords.

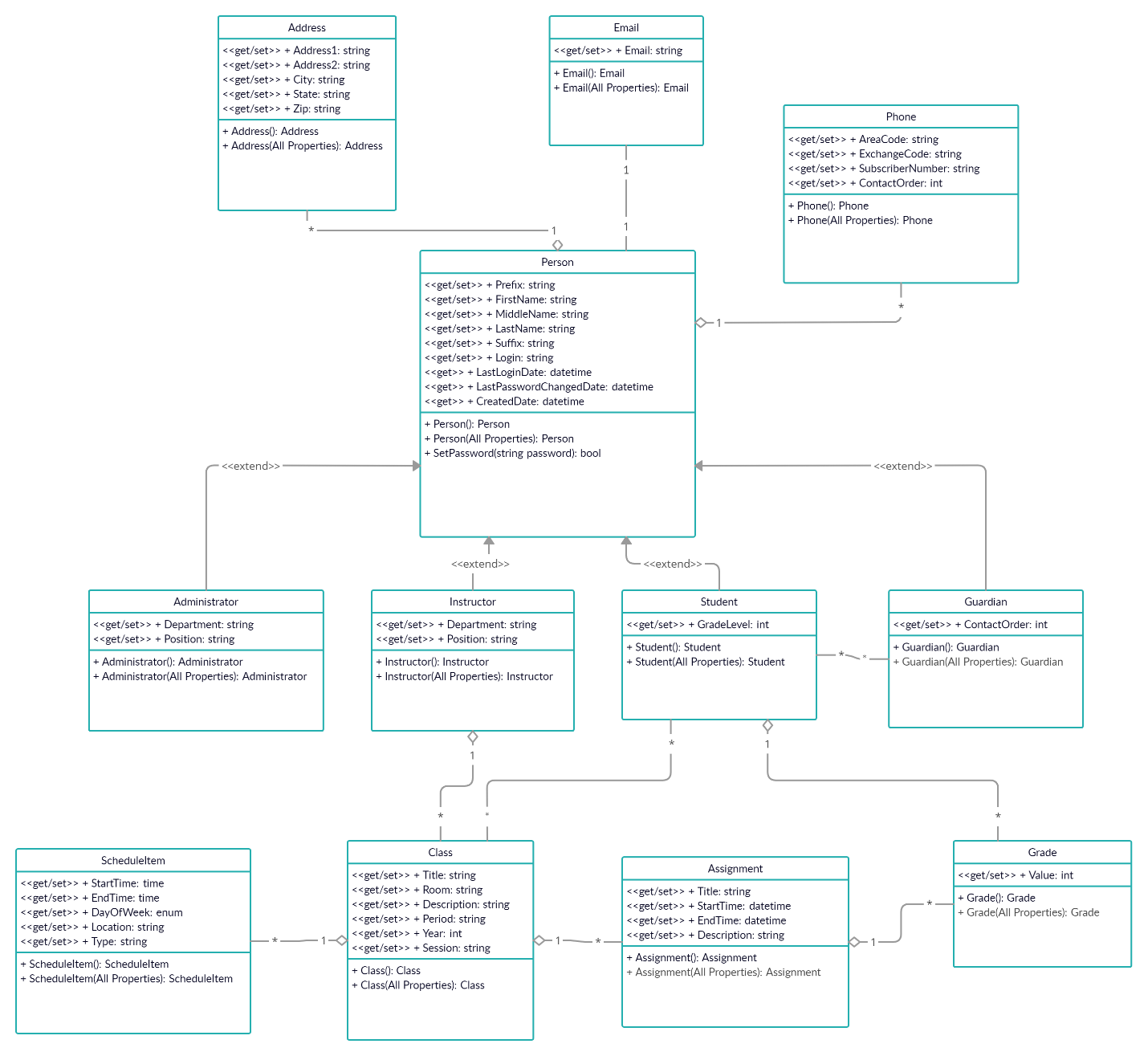
## Student

Students will be able to update their personal information and reset their passwords.

## Guardian

Guardians will be able to update their personal information and reset their passwords. Guardians will also be able to update the personal information and reset the password of any student that they are guardian of. Guardians will also be able to specify and/or modify the order in which guardians are contacted.

# Class Diagram



Additional information pertaining to this class diagram is TBD.

# HTTP Endpoints

The base url for the School Master web API is:

https://localhost:<port>/SchoolMaster/v1

## Authenticate

There will be four methods for authenticating a user and retrieving a token to be used in other calls to the web API. The four authenticate APIs are shown below.

/authenticate/administrator

/authenticate/instructor

/authenticate/student

/authenticate/guardian

## Administrator

/administrator

### POST

Submitting this request will create a new administrator.

### PUT

Submitting this request will update an existing administrator.

### GET (By Id)

Submitting this request will return the detailed information concerning the specified administrator.

URL: [https://SchoolMaster/api/v1/Administrator/ById?AdminId=<int32>](https://SchoolMaster/api/v1/Administrator/ById?AdminId=%3cint32%3e)

“AdminId” must be a valid administrator id.

Returned JSON:

{

    "id": 2,

    "department": "Department02",

    "position": "Position01",

    "prefix": "Mrs.",

    "firstName": "FirstName02",

    "middleName": "MiddleName02",

    "lastName": "LastName02",

    "suffix": "III",

    "login": "Person02Login",

    "lastLoginDate": "2021-03-02T21:46:26.08",

    "lastPasswordChangedDate": "2021-03-02T21:46:26.08",

    "createdDate": "2021-03-02T21:46:26.08",

    "email": "email02@website.com",

    "addresses": [

        {

            "address1": "Address102A",

            "address2": "Address202A",

            "city": "City02",

            "state": "MA",

            "zip": "22222"

        },

        {

            "address1": "Address102B",

            "address2": "Address202B",

            "city": "City02",

            "state": "MA",

            "zip": "22222"

        }

    ],

    "phoneNumbers": [

        {

            "areaCode": "22A",

            "exchangeCode": "22A",

            "subscriberNumber": "222A",

            "contactOrder": 1

        },

        {

            "areaCode": "22B",

            "exchangeCode": "22B",

            "subscriberNumber": "222B",

            "contactOrder": 2

        }

    ]

}

### GET (By Login)

Submitting this request will return the detailed information concerning the specified administrator.

URL: [https://SchoolMaster/api/v1/Administrator/ByLogin?AdminLogin=<string>](https://SchoolMaster/api/v1/Administrator/ByLogin?AdminLogin=#)

“AdminId” must be a valid administrator id.

Returned JSON:

{

    "id": 2,

    "department": "Department02",

    "position": "Position01",

    "prefix": "Mrs.",

    "firstName": "FirstName02",

    "middleName": "MiddleName02",

    "lastName": "LastName02",

    "suffix": "III",

    "login": "Person02Login",

    "lastLoginDate": "2021-03-02T21:46:26.08",

    "lastPasswordChangedDate": "2021-03-02T21:46:26.08",

    "createdDate": "2021-03-02T21:46:26.08",

    "email": "email02@website.com",

    "addresses": [

        {

            "address1": "Address102A",

            "address2": "Address202A",

            "city": "City02",

            "state": "MA",

            "zip": "22222"

        },

        {

            "address1": "Address102B",

            "address2": "Address202B",

            "city": "City02",

            "state": "MA",

            "zip": "22222"

        }

    ],

    "phoneNumbers": [

        {

            "areaCode": "22A",

            "exchangeCode": "22A",

            "subscriberNumber": "222A",

            "contactOrder": 1

        },

        {

            "areaCode": "22B",

            "exchangeCode": "22B",

            "subscriberNumber": "222B",

            "contactOrder": 2

        }

    ]

}

### DELETE

Submitting this request will delete the specified administrator.

URL: [https://SchoolMaster/api/v1/Administrator?AdminId=<int32>](https://SchoolMaster/api/v1/Administrator?AdminId=%3cint32%3e)

“AdminId” must be a valid administrator id.

## Instructor

/instructor

### POST

Submitting this request will create a new instructor.

### PUT

Submitting this request will update an existing instructor.

### GET

Submitting this request will return the detailed information concerning the specified instructor.

### DELETE

Submitting this request will delete the specified instructor.

## Student

/student

### POST

Submitting this request will create a new student.

### PUT

Submitting this request will update an existing student.

### GET

Submitting this request will return the detailed information concerning the specified student.

### DELETE

Submitting this request will delete the specified student.

## Guardian

/guardian

### POST

Submitting this request will create a new guardian.

### PUT

Submitting this request will update an existing guardian.

### GET

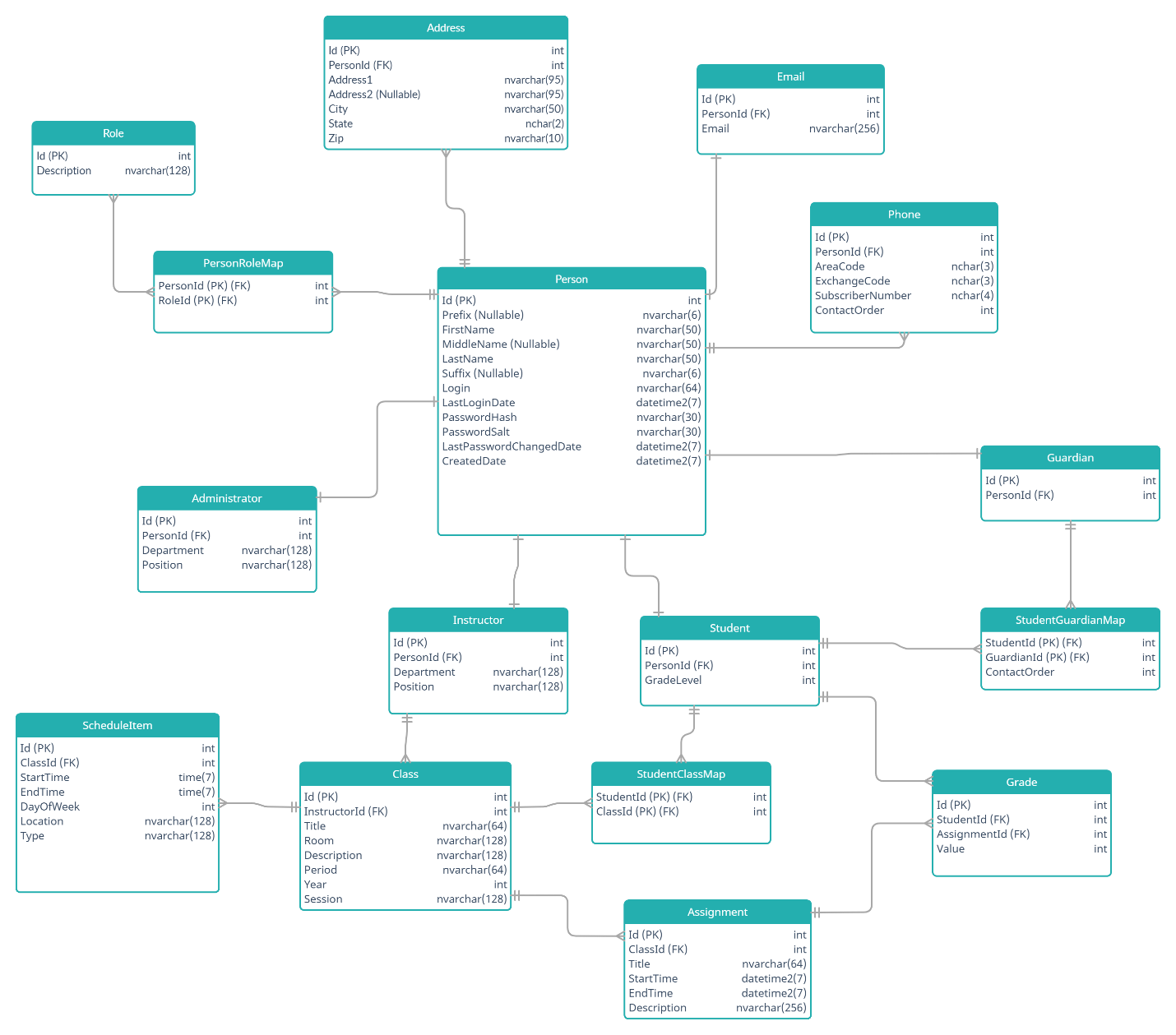
Submitting this request will return the detailed information concerning the specified guardian.

### DELETE

Submitting this request will delete the specified guardian.

# Data Store

We will be using Microsoft SQL Server 2019 as our data store.



## Person

This table represents all of the information that is common to an Administrator, Instructor, Student, and Guardian.

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Required** | **Description** |
| Id | int | Yes | Unique identifier and primary key. |
| Prefix | nvarchar(6) | No | Ex. “Mr.”, “Mrs.”, etc. |
| FirstName | nvarchar(50) | Yes |  |
| MiddleName | nvarchar(50) | No |  |
| LastName | nvarchar(50) | Yes |  |
| Suffix | nvarchar(6) | No | Ex. “Sr.”, “Jr.”, “III”, etc. |
| Login | nvarchar(64) | Yes | Must be unique. |
| LastLoginDate | datetime2(7) | Yes |  |
| PasswordHash | nvarchar(30) | Yes | User’s password after encryption. |
| PasswordSalt | nvarchar(30) | Yes | Salt used to encrypt user’s password. |
| LastPasswordChangeDate | datetime2(7) | Yes |  |
| CreatedDate | datetime2(7) | Yes |  |

## Role

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Required** | **Description** |
| Id | int | Yes | Unique identifier and primary key. |
| Description | nvarchar(128) | Yes |  |

## PersonRoleMap

This table is used to implement a many-to-many relationship between the Person and Role tables. The Primary key consists of PersonId and RoleId. This is because a person can only be assigned to a specific role one time.

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Required** | **Description** |
| PersonId | int | Yes | Foreign key to Id in Person table |
| RoleId | int | Yes | Foreign key to Id in Role table |

## Address

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Required** | **Description** |
| Id | int | Yes | Unique identifier and primary key. |
| PersonId | int | Yes | Foreign key to Id in Person table. |
| Address1 | nvarchar(95) | Yes |  |
| Address2 | nvarchar(95) | No |  |
| City | nvarchar(50) | Yes |  |
| State | nchar(2) | Yes | Two letter abbreviation. |
| Zip | nvarchar(10) | Yes | 5 digit or 9 digit with hyphen. |

## Email

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Required** | **Description** |
| Id | int | Yes | Unique identifier and primary key. |
| PersonId | int | Yes | Foreign key to Id in Person table. |
| Email | nvarchar(256) | Yes |  |

## Phone

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Required** | **Description** |
| Id | int | Yes | Unique identifier and primary key. |
| PersonId | int | Yes | Foreign key to Id in Person table. |
| AreaCode | nchar(3) | Yes |  |
| ExchangeCode | nchar(3) | Yes |  |
| SubscriberNumber | nchar(4) | Yes |  |
| ContactOrder | int | Yes | Determines the order in which phone numbers are listed. |

## Administrator

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Required** | **Description** |
| Id | int | Yes | Unique identifier and primary key. |
| PersonId | int | Yes | Foreign key to Id in Person table. |
| Department | nvarchar(128) | Yes |  |
| Position | nvarchar(128) | Yes |  |

## Instructor

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Required** | **Description** |
| Id | int | Yes | Unique identifier and primary key. |
| PersonId | int | Yes | Foreign key to Id in Person table. |
| Department | nvarchar(128) | Yes |  |
| Position | nvarchar(128) | Yes |  |

## Student

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Required** | **Description** |
| Id | int | Yes | Unique identifier and primary key. |
| PersonId | int | Yes | Foreign key to Id in Person table. |
| GradeLevel | int | Yes |  |

## Guardian

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Required** | **Decription** |
| Id | int | Yes | Unique identifier and primary key. |
| PersonId | int | Yes | Foreign key to Id in Person table. |

## StudentGuardianMap

This table is used to implement a many-to-many relationship between the Student and Guardian tables. The Primary key consists of StudentId and GuardianId.

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Required** | **Description** |
| StudentId | int | Yes | Foreign key to Id in Student table. |
| GuardianId | int | Yes | Foreign key to Id in Guardian table. |
| ContactOrder | int | Yes | Determines the order in which guardians are listed for each specific student. |

# Useful Links

[creately UML Class Diagrams](https://creately.com/diagram-type/class-diagram)

[creately UML Class Diagrams Tutorial](https://creately.com/blog/diagrams/class-diagram-tutorial/)