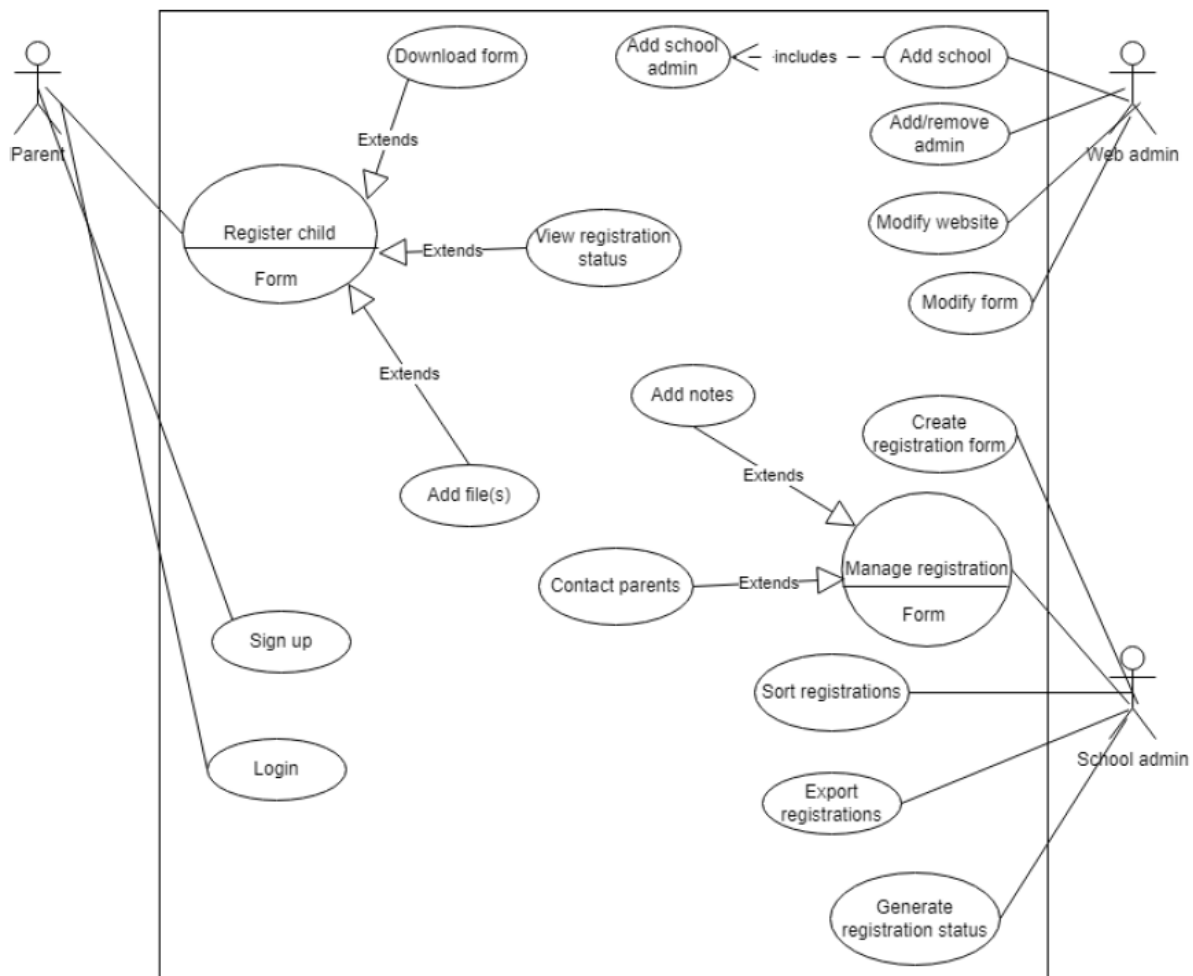


UML Case Diagram:



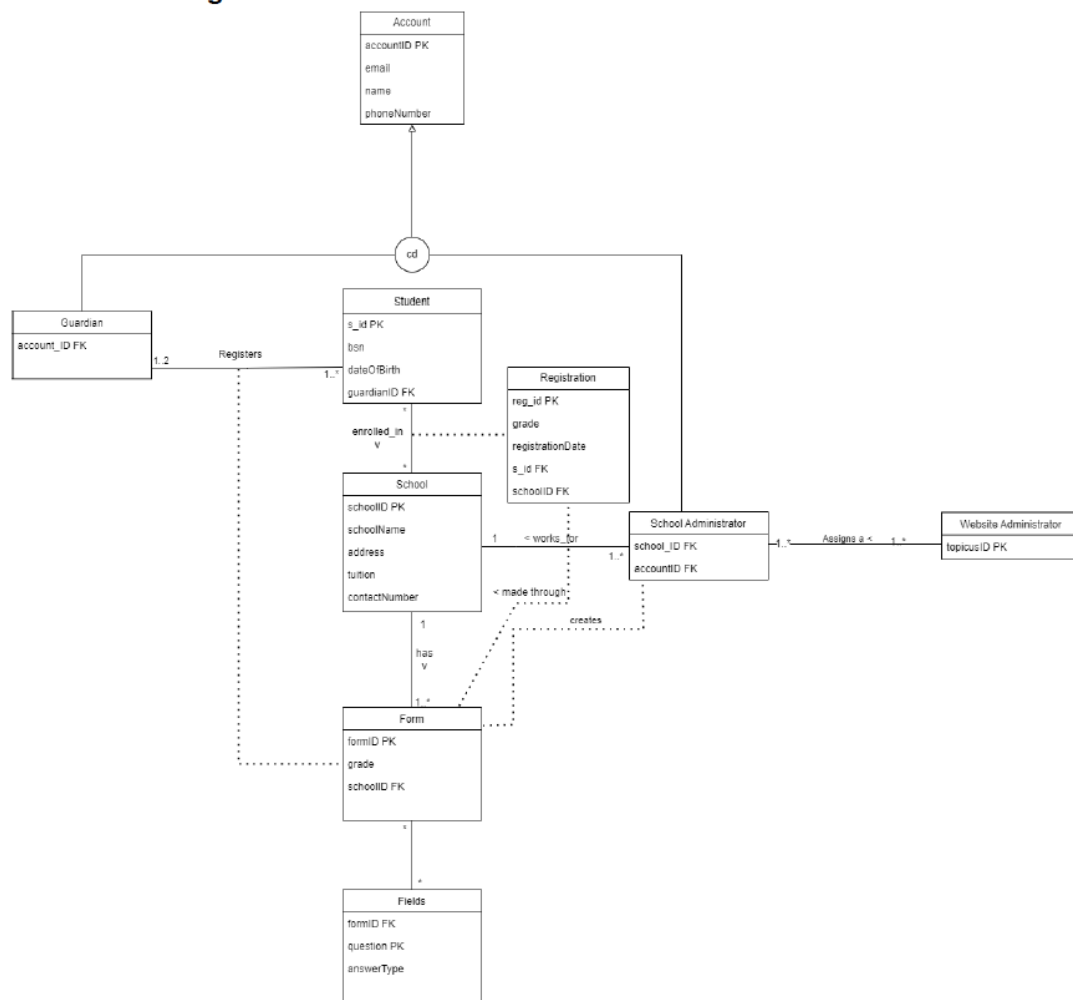
In this use case diagram, the functionalities accessible by each user in our system is outlined. The parent can register the child through a form, this form is downloadable, its status can be accompanied and it is also possible to add files to the form.

The parent may choose to sign up on our website, though that is not mandatory. If they do have an account, they may log in.

The school admin may create a registration form, and manage existing registrations made by parents through the form. They may update the status of the registration, add notes or contact the parents through the application. They are able to sort the registration in the database by alphabetical order or date. It is also possible for them to export all registrations into an Excel file.

The web admin is a user isolated from our system. He has a completely different website environment. He may modify the web application itself, add or remove schools and manage their respective administrators.

UML Class Diagram:



This is the class diagram that we have developed in order to derive the DB schema. It is possible to see that there are two types of accounts: Guardians (parents) and School Administrators. The parent is able to register a child at a school by filling out a form. This form is created by the school admin and may have multiple fields. This diagram illustrates the main functionality of the system.

DB Schema

This is the schema that was derived from the aforementioned class diagram. Constraints were not illustrated, and as of 30 June 2023 - changes were made to the initial document.

account(account_id, email, phone_number, name, address,
PK(account_id)

)

guardian(guardian_id,
FK(guardian_id) REF account(account_id) ON DELETE CASCADE

)

```
student(student_id, bsn, name, birth_date, guardian_id,  
        PK(student_id),  
        FK(guardian_id) REF guardian(guardian_id) ON DELETE CASCADE  
)
```

```
webadmin(topicus_id, account_id,  
        PK(topicus_id),  
        FK(account_id) REF account(account_id) ON DELETE CASCADE  
)
```

```
school(school_id, school_name, address, tuition, contact_number,  
        PK(school_id)  
)
```

```
schooladmin(school_id, account_id,  
        FK(account_id) REF account(account_id) ON DELETE CASCADE,  
        FK(school_id) REF school(school_id) ON DELETE CASCADE  
)
```

```
registration(registration_id, grade, registration_date, student_id, school_id,  
        status, allowedit,  
        PK(registration_id),  
        CHECK (registration_id >= 0 AND registration_id <= 9999999)  
        FK(school_id) REF school(school_id) ON DELETE CASCADE,  
        FK(student_id) REF student(student_id) ON DELETE CASCADE  
)
```

```
form(form_id, grade, school_id,  
        PK(form_id),  
        FK(school_id) REF school(school_id) ON DELETE CASCADE  
)
```

```
fields(form_id, question, input_type, question_id,  
        PK(question_id),  
        FK(form_id) REF form(form_id) ON DELETE CASCADE,  
        CHECK (input_type::text = 'email'::text OR input_type::text = 'number'::text OR  
        input_type::text = 'date'::text OR input_type::text = 'text'::text OR input_type::text =  
        'tel'::text OR input_type::text = 'file'::text OR input_type::text = 'header'::text)  
)
```