Updated 14.08.2024

The DSS is international and will be use in differnet languages (English, French and Arabic), it seperated in 2 levels backend (Microservice Java ) and frontend (Angular) and contain following services

* School Service
* BFF Service
* DM Service (Saving and reporting files)
* Student Service
* Teacher Service
* Timetable Service

## Data Model Representation (ER Diagram)

## 1. School (1) ==> Students (many)

## 2. School (1) ==> Teachers (many)

## 3. School (1) ==> Employees (many)

## 4. School (1) ==> Courses (many)

## 5. Class (many) ==> Students (many)

## 6. Class (many) ==> Teachers (many)

## 7. Teacher (many) ==> Courses (many)

## 8. Student (many) ==> Courses (many)

## 9. Address (1) ==> Student (many)

## 10. Address (1) ==> Teacher (many)

## 11. Address (1) ==> Employee (many)

## DSS Backend

### Interaction Flow

1. Frontend requests a report (e.g., a PDF or Excel) via the BFF service.
2. BFF calls the Student and/or Teacher service to fetch relevant data.
3. BFF forwards the data to the Document Management Service (DMS) along with the report format (PDF, Excel, etc.).
4. DMS generates the report using JasperReports and stores the file.
5. BFF provides the download link or returns the report file to the Frontend.

### School Service

Contain a main information about the school and management the module of the school

1. School name
2. school contact (adress) infos
3. School registration number + starting date
4. Logo
5. School address + google Map guiding to school
6. Add or remove Module
7. Fee information and Promotion
8. Show summary of all activity in the school
9. Manage Guardians
10. Summary of all students’ achievements
11. Show Diagram with Teacher achievements
12. Attendance
13. Fee Management
14. Income & Expenses Management
15. Salaries Management
16. Note
17. Others

### Backend-For-Frontend (BFF) Service

### Orchestrates Requests from Frontend (or External Client) to

### Student, Teacher and Document Management (DM) Service

### Makes API calls to fetch the Required data from Student, Teacher Services.

### Passes this Data to DM for reporting gernation

### Return the generate reports back to Frontend

### Authentication Service

### Document Management (DM) Service

* Handle **report generation** using **JasperRports**
* Stores generated reports (PDF, Excel, etc) in the file system or Cloud Storage service (like AWS S3)
* Expose APIs to generate and download reports based on data received ftom other services (e.g. from BFF)
* Stores repot template (JRXML files) and uses the to generate reports dynamically.

### Student Service (Students management)

* Focused on manging student data (CRUD operstaions, fetching student records)
* Provides an API to retrive student data.

1. Add Student ------------------------------------------------- done
   1. Additional documents for registration done
      1. Original Certified Birth Certificate
      2. Original Social Security Card
      3. Current Immunization Form
      4. Current Proofs of Residence (ex. power, water or gas bill)
      5. Other
2. Get All Students done
3. get Student by (Id, name, lastname, email ….) done
4. Password forgotten
   1. [http://localhost:8080/dss/api/v1/students/student/passwordforgot](http://localhost:8080/um/api/v1/students/student/passwordforgot)

link will be called with **email or username** you well receive a link to add a new password

* 1. After you receive this link you must click at it and give your password and confirm password the will be changed.

1. Reset password
   1. [http://localhost:8080/um/api/v1/students/student/{**student\_id**}/resetpassword](http://localhost:8080/um/api/v1/students/student/%7Bstudent_id%7D/resetpassword)

Add a old password

Add a new password 1

Add a new password 2

Dann sent it become a message that your password changed successfully or not

* 1. After you receive a email with link to change or reset your password link look like this [http://localhost:8080/um/api/v1/students/student/{student\_id}/{resetpassword\_hashcode}/resetpassword](http://localhost:8080/um/api/v1/students/student/%7Bstudent_id%7D/%7Bresetpassword_hashcode%7D/resetpassword)

1. Add profile foto
2. update Student
3. Add student note
4. Add student report
5. Edit Student info ----------------------------------------------------- done
6. Show all Students ----------------------------------------------------- done
7. Add Address to student
8. Add Student to class
9. Add Student to Exam or test
10. Show Student report
11. Show Student exams
12. Import and export Students as Excel file or pdf files
13. Show Student notes (Result)
14. Add contact person

**Schulregistrierungsformular**

Ein typisches Schulregistrierungsformular enthält Felder, um die grundlegenden Informationen über den Schüler sowie die Kontaktinformationen der Eltern oder Erziehungsberechtigten zu erfassen. Es könnte ungefähr so aussehen:

**Schulregistrierungsformular done**

1. **Schülerinformationen**

Vorname : [Textfeld]

Nachname : [Textfeld]

Geburtsdatum : [Datumsauswahl]

Geburtsort : [Textfeld]

Geschlecht :

Männlich

Weiblich

Divers

Staatsangehörigkeit : [Textfeld]

Adresse :

Straße : [Textfeld]

Hausnummer : [Textfeld]

Postleitzahl : [Textfeld]

Stadt : [Textfeld]

Klasse (für die Anmeldung): [Dropdown-Auswahl]

1. **Eltern-/Erziehungsberechtigte Informationen:**

Name des Erziehungsberechtigten :

Vorname : [Textfeld]

Nachname : [Textfeld]

Beziehung zum Schüler :

-Mutter

-Vater

- Andere: [Textfeld]

Telefonnummer : [Textfeld]

E-Mail-Adresse : [Textfeld]

Adresse (falls unterschiedlich von Schüleradresse):

Straße : [Textfeld]

Hausnummer : [Textfeld]

Postleitzahl : [Textfeld]

Stadt : [Textfeld]

#### 3. Notfallkontakte:

Name des Notfallkontakts : [Textfeld]

Telefonnummer : [Textfeld]

Beziehung zum Schüler : [Textfeld]

#### 4. Vorherige Schulbildung:

Name der vorherigen Schule : [Textfeld]

Schulabschluss/Klasse : [Textfeld]

#### 5. Medizinische Informationen:

Medizinische Bedingungen (z.B. Allergien, chronische Krankheiten): [Textfeld]

Name des Hausarztes : [Textfeld]

Kontakt des Hausarztes : [Textfeld]

#### 6. Einwilligungen:

[ ] Ich stimme der Teilnahme meines Kindes an außerschulischen Aktivitäten zu.

[ ] Ich stimme der Verarbeitung der personenbezogenen Daten gemäß der Datenschutzrichtlinie der Schule zu.

#### 7. Dokumente Hochladen:

Geburtsurkunde : [Datei-Upload]

Impfpass/Medizinische Nachweise : [Datei-Upload]

Letztes Schulzeugnis : [Datei-Upload]

#### 8. Unterschrift:

Unterschrift des Erziehungsberechtigten : [Feld für elektronische Unterschrift oder ausdruckbar]

#### 9. Absenden:

- [Schaltfläche "Absenden"]

Diese Felder können je nach den Anforderungen der Schule variieren, aber dies ist eine allgemeine Übersicht der benötigten Informationen.

### Timetable Service

1. Show all timetables
2. Add new item to Timetable

### Document Storage Service

1. New Service for image + files uploading and downloading (it used for document, exams…us)

### Teacher Service

### **Employee Service**

1. Manage Staff
2. Show Employees ----------------------------------------------------- done
3. Add new Employee ----------------------------------------------------- done
4. Edit Employee ----------------------------------------------------- done
5. Deactivate Employee

### Classes Service (Class management)

1. Add new Class ----------------------------------------------------- done
2. Show all classes ----------------------------------------------------- done
3. Edit class ----------------------------------------------------- done

### Report Service

1. Annual Report
2. Teacher’s Reports
3. Director ‘s Repots

**DSS Areas**

* Administration’s Area
* Students
* Parent Area
* Teacher’s Area
* Employee Area

## DSS Frontend

Frontend is user interface and what user seen and react with it. It conatin following moulde

### Login

### Password forgotten

* 1. If you have forgot your password
     + 1. Click on password forgotten
       2. <http://localhost:8080/um/api/v1/students/student/forgotpassword> link will be called
       3. With following information **email or username** then email will be send to user with link to reset password in time < 1 day
       4. Give your email to become a link to add a new password
       5. Give password and confirm password must be equals
       6. You become message your password successful changed
  2. After that you can login with new password

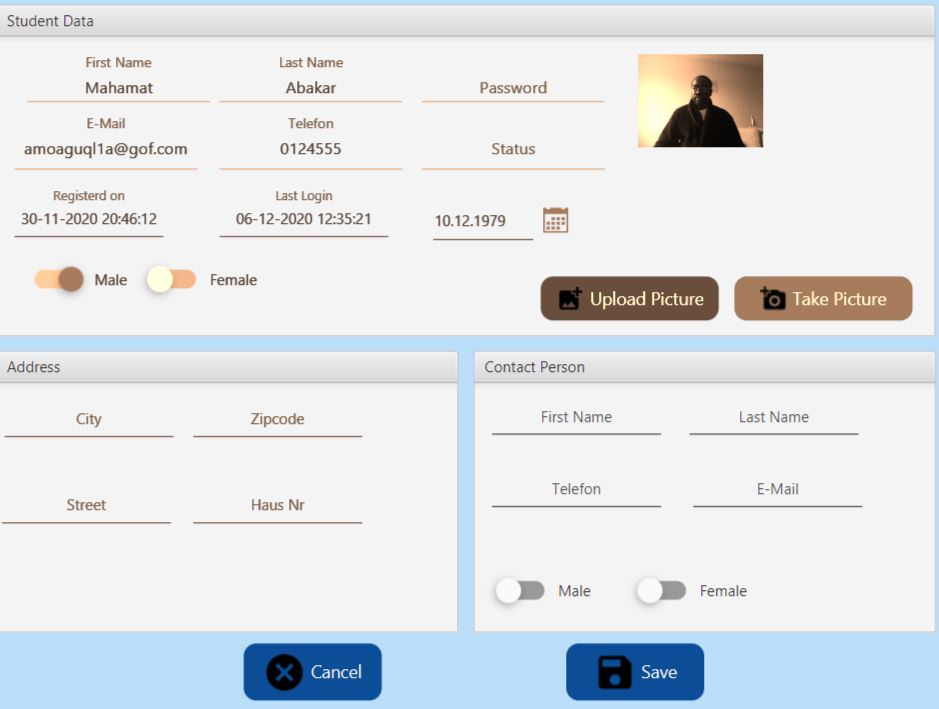
### Reset password

* 1. if dfsd If you want reset your password
     + 1. Click reset password
       2. [http://localhost:8080/um/api/v1/students/student/{**student\_id**}/resetpassword](http://localhost:8080/um/api/v1/students/student/%7Bstudent_id%7D/resetpassword) this link will be called and automatically email will be send to user with link for changing password
       3. Add a old password
       4. Add a new password
       5. Add a confirmation password same as new password
       6. Dann sent it become a message that your password changed successfully or not
  2. After you receive a email with link to change or reset your password link look like this [http://localhost:8080/um/api/v1/students/student/{student\_id}/{resetpassword\_hashcode}/resetpassword](http://localhost:8080/um/api/v1/students/student/%7Bstudent_id%7D/%7Bresetpassword_hashcode%7D/resetpassword)

After you receive this link you must click at it and give your password and confirm password the will be changed.

Registeration

### Student Registeration Information



Name  First Name  Last Name

Birth Date MonthDayYearPick a DateDate Picker Icon

Gender 

Nationality 

Email Address

Entry Year

 Grade  Semester 

Have you previously applied to or attended this school?

YesNo

Current Residence Information

Address

 Street Address

 Street Address Line 2

 City State / Province

 Postal / Zip Code

Home Phone Number

Primary Residence Information (if different from above)

Address

 Street Address

 Street Address Line 2

 CityState / Province

 Postal / Zip Code

Parent/Guardian Residence Information (if different from above)

Address

 Street Address

 Street Address Line 2

 City  State / Province

 Postal / Zip Code

**Future functions**

1. Biometric & Manuel Attendance
2. Biometric Attendance
3. ID-Cards (for Student, Teacher, Employee ….)
4. Exams & Results Card (Certificate)
   1. Add exam
   2. Show all exams Results
5. SMS Alerts (For Parent, Headmaster, Supervisor---)
6. Daily Auto Backups
7. View Psersonal infos
8. View Exam’s Mark
9. Timetable
   1. Show all timetables
   2. Add new item to Timetable
10. Connect with Teacher and School
11. Employee Area (every person can see his own infos)
    1. Show Employees
    2. Add new Employee
    3. Edit Employee
    4. Annual Report
    5. Exam
    6. Transfer class
       1. Teacher’s Reports
       2. Director ‘s Repots
12. Schooler’s Area
    1. Employee
13. Teacher :-
    1. Add Student to his Class Room
    2. View time table
    3. Add/edit/delete a script
    4. Add Students Marks
    5. Connect with Parent to send a report
14. Student:-
    1. View Psersonal infos
    2. View Time Table
    3. View Exam’s Mark
    4. Connect with Teacher and School
15. Parent:-
    1. View a report about his Child
    2. View report about the school level (Marks)
    3. View the activities of the school
    4. Connect Director or Teacher
16. Dashboard
    1. Reports
    2. Show summary of all activity in the school
    3. Show Diagram with Teacher achievements
    4. Summary of all students’ achievements
    5. Show summary of school activities
    6. Show all exams Results
17. Settings
    1. Change theme
    2. Change font size

**Future Extensions**

1. Webinar for online Seminars
2. Online Exam
3. App for DSS

### Permission service

* 1. Roles
     1. Manager Role (can do anything)
        1. Have all Privileges
     2. Student Role (see only his area)
     3. Contact Person Role
     4. Admin Role (create, update, delete, read)
     5. Teacher Role (Create, Read, Update)
     6. Student Role (read only own infos)
     7. Emplyee Role (read only own infos)
  2. Privileges
     1. Create new user as Student you must have a right to create a new user and there is
     2. Update user or Edit user
     3. Delete User
     4. Access management
     5. Add Role to users
     6. Delete Role form User
     7. Create a time table for Teachers and Student
     8. Room management
     9. Staff management
     10. Create a new Employee
     11. Involve Parent bei Registration
     12. Add a read and write Rights to partent
     13. Add a new Class Room

| **Privilege** | **Description** |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |

**Chat GPT**

I have following backend services school-service managed (school, Address, classes, employee) , student-service managed (student, Guardian, Constent), teacher-service managed (Teacher, Course ) , document-service managed (all documents), authentication-service (authentication management) and BFF-service managed complex request from frontend to backend) all serivces are finished and ready for work. I need now good and attractive design idee for the frontend with Angular with followings:- scss, angular material, bootstrap, routing?

For your frontend with Angular, SCSS, Angular Material, Bootstrap, and routing, here’s an attractive and user-friendly design approach that can integrate all of your services:

### 1. \*\*Layout Structure\*\*

- \*\*Header (Navbar)\*\*: Use Angular Material’s toolbar and Bootstrap navbar components. It will include links for navigating between the different modules such as:

- \*\*Home\*\*

- \*\*Students\*\*

- \*\*Teachers\*\*

- \*\*Documents\*\*

- \*\*Classes\*\*

- \*\*Authentication\*\*

- \*\*Sidebar\*\*: Angular Material’s `sidenav` component is perfect for a collapsible sidebar that houses navigation for each service-specific section. It could include icons for each service, creating a clear visual distinction.

- \*\*Main Content\*\*: The main section will be dynamic based on routing and will load content related to the chosen service (e.g., students, teachers, documents).

- \*\*Footer\*\*: A sticky footer with links to external resources or a quick overview of the app could use Bootstrap’s footer components.

### 2. \*\*Color Scheme and Typography\*\*

- Choose a \*\*minimalistic and modern color palette\*\*: shades of blue, white, and grey are often associated with educational systems and look professional.

- Use \*\*Angular Material’s typography\*\* system to ensure the design stays modern. Pair it with Bootstrap’s responsive utilities to ensure readability on different devices.

### 3. \*\*Landing Page (Home)\*\*

- \*\*Hero Section\*\*: A bold hero section using Angular Material’s card and grid system to highlight key features or quick links to important parts of the app. Consider using a background image related to education.

- \*\*Quick Links\*\*: Buttons or tiles that link to frequently used sections like “View Students”, “Manage Teachers”, or “Upload Documents”.

- \*\*Responsive Grid\*\*: Use Bootstrap’s grid system combined with Angular Material’s grid list to ensure smooth responsiveness.

### 4. \*\*Individual Service Views\*\*

Each section (students, teachers, documents) will have a consistent layout but with service-specific details:

- \*\*Students Page\*\*:

- A table (using Angular Material Table with Bootstrap styling) to display all students.

- Use Angular Material’s card for individual student profiles, with buttons for viewing, editing, and deleting.

- \*\*Teachers Page\*\*:

- Similar structure with a table of teachers and a detailed card view for each teacher.

- \*\*Document Management\*\*:

- Use file upload features from Angular Material and a detailed list view for managing documents.

- Include search and filter options for easy document retrieval.

### 5. \*\*Angular Material Components\*\*

- \*\*Forms\*\*: For student, teacher, or employee data entry, use Angular Material’s form field with `mat-select`, `mat-input`, etc. SCSS can customize these for branding and uniformity.

- \*\*Tables\*\*: Angular Material’s data tables with sorting, pagination, and filtering would be ideal for managing lists (e.g., students, teachers).

- \*\*Dialogues\*\*: Use modals from Angular Material (with SCSS customization) for pop-up forms, confirmation boxes, or notifications.

### 6. \*\*Advanced Routing\*\*

- \*\*Lazy Loading\*\*: Set up lazy-loaded routes for each major section to improve performance.

- \*\*Role-based Routing\*\*: Integrate your authentication-service for protected routes (e.g., only admins can access the document upload section).

### 7. \*\*Reusable Components\*\*

- \*\*Card Component\*\*: A reusable Angular Material card component that can be used to display details for students, teachers, or documents.

- \*\*Search and Filter Bar\*\*: A consistent search bar across all major views to filter lists of students, teachers, or documents.

### 8. \*\*Responsiveness\*\*

- Leverage \*\*Bootstrap’s grid system\*\* and \*\*Angular Material’s flex layout\*\* for responsive design. Ensure it works well on both desktop and mobile devices.

### 9. \*\*Animations and Transitions\*\*

- Add subtle animations using Angular’s animation module for route transitions, button clicks, and hover effects. This adds a polished feel to your app.

### 10. \*\*SCSS Customization\*\*

- Use \*\*SCSS variables\*\* to manage consistent theming (e.g., primary, secondary colors). SCSS can also help in creating mixins for responsive typography and layouts.

- Custom CSS classes for shadows, spacing, and hover effects can give the interface a modern and interactive look.

This design leverages Angular Material for structure, Bootstrap for responsive layout, and SCSS for customizations to give it a clean, efficient, and modern interface. Let me know if you'd like further details on any of the sections!