

# Student Management System

---

Developer: Sapnesh Naik [sapnesh@kerneldev.com](mailto:sapnesh@kerneldev.com)

---

## Common Terms Used:

- AR: Action Required
- TBD: To Be Decided
- TBC: To Be Confirmed

## Overview:

---

Student Management System provides a complete all-in-one solution to manage all aspects of a student in an academic institution.

## Application modules:

### 1. Base Module:

- Registration of all valid entities.
- API token based authentication (Login and Logout).
- Updating entity details.
- Role and permission based access control.
- Removing an entity.
- View Student/Parent/Staff/Admin individual profile
- View all Student/Parent/Staff/Admin

### 2. Attendance Management:

3. Details TBD

### 4. Class Management:

5. Details TBD

### 6. Hostel Management:

7. Details TBD

## 8. Student Management:

### 9. Details TBD

## Tech Stack:

---

- Backend API - Laravel 6
- Frontend - Vue.js v2
- HTML Template - given
- SMS Gateway - TBD, AR Deepak
- Email Gateway - TBD, AR Deepak

## Base Module:

---

- **Permissions:**

Permissions are a mechanism which enable granular level access control to entities in the SMS.

Permissions are atomic, static and resource specific. These part of the base system and cannot be modified.

For example: It does not make sense to add a new permission "Add Property" without there being a property resource supported by the application.

### **All available permissions:**

1. Can Register Student
2. Can Update Student
3. Can Delete Student
4. Can View Students
5. Can View Student Profile
6. Can Register Parent
7. Can Update Parent
8. Can Delete Parent
9. Can View Parents
10. Can View Parent Profile
11. Can Register Staff
12. Can Update Staff
13. Can Delete Staff
14. Can View Staff
15. Can View Staff Profile

16. Can Create Admin
17. Can Delete Admin
18. Can Update Admin
19. Can View Admins
20. Can View Admin Profile

- **Roles/Entities:**

Roles can be defined as a logical collection of permissions. Every role has the default permission to view/update his/her own profile. The below 4 roles are created by default.

**TBD:**

1. Whether to give an option to add more roles with other permissions? For Ex. An accountant role: with account related permissions can be assigned to a Staff user by admin. (I recommend this) this
2. Or give the admin the option to add/revoke individual permissions to a user.

**Roles and their respective granted permissions:**

3. Student
4. Parent
5. Staff
6. Teaching (role)
7. Non-teaching (role)
8. Admin

- All permissions

- **Registration:**

UI of the registration page will be taken from the given template. Registration page will be an authenticated page (user needs to be logged in) and the user should have “Can Register Student” permission or Admin role.

1. **Student:**

- Who which entity should be able to add/create a student entity?
- Admin or Any Staff user with “Can Register Student” permission or is assigned a role which has “Register Student” permission.
- Registration flow?:
  - AR Deepak
    - Student details required for registration:
- Ex. Password strength (registration SMS) login only on admission
- Verification:

- Phone number should be verified? no
- Email address should be verified? no
- Deletion:
  - Who can delete a student?
- active / inactive
- Deletion criteria.
  - AR Deepak
- Active/Inactive/FA (Final Account)

## 2. Parent:

- Which entity should be able to add/create a parent entity?
- Admin or Any Staff user with “Register Parent” permission or is assigned a role which has “Register Parent” permission.
- Registration flow?:
- AR Deepak
- Parent details required for registration:
- AR Deepak
- Deletion criteria
- Verification:
- Phone number should be verified?
- Email address should be verified?

## 3. Admin:

- Initial admin will created from the backend by the developer.
- Can the initial admin create another new admin?
- If yes, Admin will have “Register Admins” permission
- Admin details required:
- AR Deepak, comm details. personal details
- Deletion criteria, super admin can disable other admins
- admins created by super admin cannot disable super admin
- Verification:
- For initial admin:
- Not required
- For future admins?: Phone/Email?
- AR Deepak Not required

#### 4. Staff:

- Which entity should be able to add/create a staff entity?
- A user with “Can Register Staff” permission or is assigned a role which has “Can Register Staff” permission.
- Which entity should be able to assign a sub-role of teaching/non-teaching staff to a staff entity?
- A user with “Can Update Staff” permission or is assigned a role which has “Can Update Staff”
- Registration flow?: Admin/ whoever has the register staff role
- Staff details required: AR Deepak TBD
- Teaching staff specific Details: AR Deepak, Not required, same as staff
- Non-teaching staff specific details: AR Deepak Not required, same as staff
- Verification: AR Deepak

- **Login:**

- Requires:

1. Username (Email? or Phone Number?) TBD AR Deepak (userid created during admission)
2. Password:
3. Remember me (generates a token with longer expiry time)

- Upon successful login a JWA token will be created by the API and returned to the requester. The user can then use this token for future requests.

- Login UI:

- From template.

- Locations (TBC Deepak):

- For Student/Parent/Staff main landing page.

- For Admin: <webiste.com/admin>

- **Dashboard:**

The UI will be taken from template provided. Future modules will add to the same dashboard's menus. Menu options enabled by current module are listed below.

1. **Student:**

- Home page content? TBD AR Deepak
- View Profile
- Update Profile

## 2. Parent:

- Home page content? TBD AR Deepak
- View Profile
- Update Profile
- View ward profile

## 3. Staff (Teaching/ non teaching):

- Home page content? TBD AR Deepak
- View Profile
- Update profile
- Register a student? (Conditional, based on Permission/role)

## 4. Admin:

- Home page content? TBD AR Deepak
- View Profile
- Update Profile
- View all staff
- View all students
- View all parents
- Delete a Parent/Staff/Student? TBD Deepak
- Create a role.
- Update a role.
- Delete a role (Which is not currently assigned to any user).
- Assign a role to a user