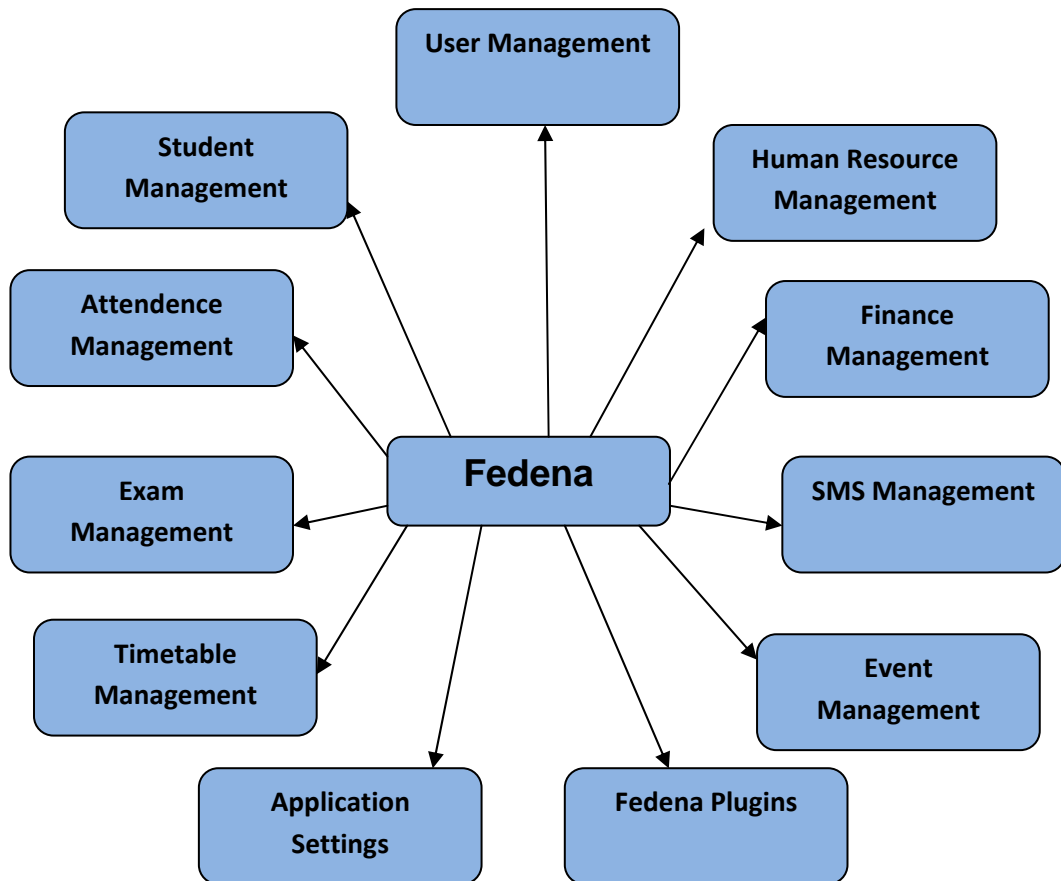


# Project Fedena Technical Documentation



# Project Fedena Technical Document



## **User Module**

The user management module handles the authentication and authorization of users for different pages.

### **Models and Tables**

#### **Associated models**

**User**  
**Student**  
**Employee**

#### **Database Structure**

##### **User**

**id**( Integer )  
**username**( String ) Username is "admin" for the super-admin that is created automatically, admission no. for students and employee ids for employees.  
**first\_name**( String )  
**last\_name**( String )  
**email**( String )  
**admin**( Boolean ) Set for administrators  
**student**( Boolean ) Set for Students  
**employee**( Boolean )Set for Employees  
**hashed\_password**( String )  
**salt**( String )  
**reset\_password\_code**( String ) The reset password code sent to the email of user for resetting password after user fills out the forgot password form.  
**reset\_password\_code\_until**( Datetime ) Expiry date for the reset password option  
**created\_at**( Datetime )  
**updated\_at**( Datetime )  
**parent**(Boolean)

### **Controllers**

The following controllers are associated with the user management module.

#### **UserController**

##### **Actions/Methods**

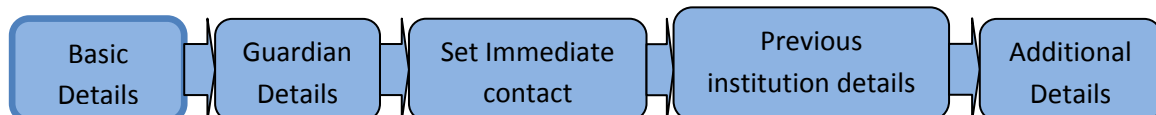
**change\_password** Action for initiating the password change of user  
**choose\_layout** Method for choosing action to login page / forgot password page / dashboard according to user request  
**create** Method to create User Object  
**current\_user** Method to return the User Object of current logged in user  
**dashboard** Shows links to all the modules based on the user's permission  
**delete** Deletes a User object

**edit** Edit a User Object  
**edit\_privilege** Edit privileges of a user  
**find\_finance\_managers** Finds the Finance **manager(s)** for the User  
**forgot\_password** Intiate the forgot\_password view  
**header\_link** Action for showing the header at the top of the user page showing user name and messages  
**list\_employee\_user** List all users with role Employee  
**list\_student\_user** List all users with role student  
**list\_parent\_user** List all users with role parent  
**list\_user** List all users  
**login** Default login page  
**logout** Action for destroying the current user's session  
**profile** Show a user's profile  
**reset\_password** Proceed to reset password the current user's password after checking the password\_reset\_feild  
**search\_user\_ajax** Ajax search of users  
**set\_new\_password** Set a new password after a valid forgot password attemp on a password change action  
**user\_change\_password** intiate change password action for a users  
**successful\_user\_login**  
**all** To find all users

## **Student management**

To Manages students , guardians and former students.

### **Steps in students admission**



### **Models and Tables**

#### **Associated models**

***Student***  
***Guardian***  
***ArchivedStudent***  
***Subject***  
***StudentSubject***  
***Batch***  
***Course***  
***StudentAdditionalDetail***  
***StudentAdditionalField***  
***StudentCategory***  
***StudentPreviousData***  
***StudentPreviousSubjectMark***  
***SubjectLeave***

## **Database Structure**

### **Student**

**id**( Integer )  
**admission\_no**( String ) Admission number of the student , generated automatically or manually entered  
**class\_roll\_no**( String )  
**admission\_date**(date)  
**first\_name**( String )  
**middle\_name**( String )  
**last\_name**( String )  
**batch\_id**( Integer )  
**date\_of\_birth**(date)  
**gender**( String )  
**blood\_group**( String )  
**birth\_place**( String )  
**nationality\_id**( Integer )  
**language**( String )  
**religion**( String )  
**student\_category\_id**( Integer )  
**address\_line1**( String )  
**address\_line2**( String )  
**city**( String )  
**state**( String )  
**pin\_code**( String )  
**country\_id**( Integer )  
**phone1**( String ) Required if SMS is available for the student  
**phone2**( String )  
**email**( String )  
**immediate\_contact\_id**( Integer )  
**is\_sms\_enabled**( Boolean ) Set if SMS alerts is to be available for the student.  
**photo\_filename**( String )  
**photo\_content\_type**( String ) Image data for the user's image  
**photo\_data**( Binary ) Image data for the user's image  
**status\_description**( String )  
**is\_active**( Boolean )  
**is\_deleted**( Boolean )  
**created\_at**( Datetime )  
**updated\_at**( Datetime )  
**has\_paid\_fees**( Boolean )  
**photo\_file\_size**( Integer )  
**user\_id**( Integer )

### **ArchivedStudent**

Once a student is removed from rolls of an institution his/her details are stored in archived students table

**id**( Integer )  
**admission\_no**( String )  
**class\_roll\_no**( String )  
**admission\_date**(date)

**first\_name**( String )  
**middle\_name**( String )  
**last\_name**( String )  
**batch\_id**( Integer )  
**date\_of\_birth**(date)  
**gender**( String )  
**blood\_group**( String )  
**birth\_place**( String )  
**nationality\_id**( Integer )  
**language**( String )  
**religion**( String )  
**student\_category\_id**( Integer )  
**address\_line1**( String )  
**address\_line2**( String )  
**city**( String )  
**state**( String )  
**pin\_code**( String )  
**country\_id**( Integer )  
**phone1**( String )  
**phone2**( String )  
**email**( String )  
**photo\_filename**( String )  
**photo\_content\_type**( String )  
**photo\_data**( Binary )  
**status\_description**( String )  
**is\_active**( Boolean )  
**is\_deleted**( Boolean )  
**immediate\_contact\_id**( Integer )  
**is\_sms\_enabled**( Boolean )  
**created\_at**( Datetime )  
**updated\_at**( Datetime )  
**former\_id**( String )  
**photo\_file\_size**(integer)

### **Guardian**

**id**( Integer )  
**ward\_id**( Integer ) The child's student Id  
**first\_name**( String )  
**last\_name**( String )  
**relation**( String ) Relationship to the child ( father / mother / .. )  
**email**( String )  
**office\_phone1**( String )  
**office\_phone2**( String )  
**mobile\_phone**( String ) Number for sending alerts  
**office\_address\_line1**( String )  
**office\_address\_line2**( String )  
**city**( String )  
**state**( String )  
**country\_id**( Integer )  
**dob**(date)  
**occupation**( String )  
**income**( String )

**education**( String )  
**created\_at**( Datetime )  
**updated\_at**( Datetime )  
**user\_id**(integer)

### **StudentAdditionalDetail**

**id**(Integer)  
**student\_id**(Integer)  
**additional\_field\_id**(Integer)  
**additional\_info**(String)

### **StudentAdditionalField**

**id**(Integer)  
**name**(String)  
**status**(Boolean)

### **StudentCatgeory**

**id**(Integer)  
**name**(String)  
**is\_deleted**(Boolean)

### **StudentPreviousData**

**id**(Integer)  
**student\_id**(Integer)  
**institution**(String)  
**year**(String)  
**course**(String)  
**total\_mark**(String)

### **Controllers**

#### **StudentController**

#### **Actions / Methods**

**academic report all** :Academic report for all the students

**admission1**:First form of admission. Asks for basic details.

**Admission2**: Second form in student admission for adding guardian detail for the student

**admission3** : Third form in student admission for selecting immediate contact from guardian.

**admission3 1**: Form for editing the guardian details once entered

**previous data** : Fetch details of previous institution

**previous data edit** : To edit the previous data of a student

**previous subject** :Fetch details of subjects from previous institution

**save previous subject** : Save subject details of previous details

**delete previous subject** : To delete the previous subject mark of a student

**admission4** : Form to add Additional Details for the student

**edit admission4** : Edit the previous details

**add additional details** : Add Additional details for a student.

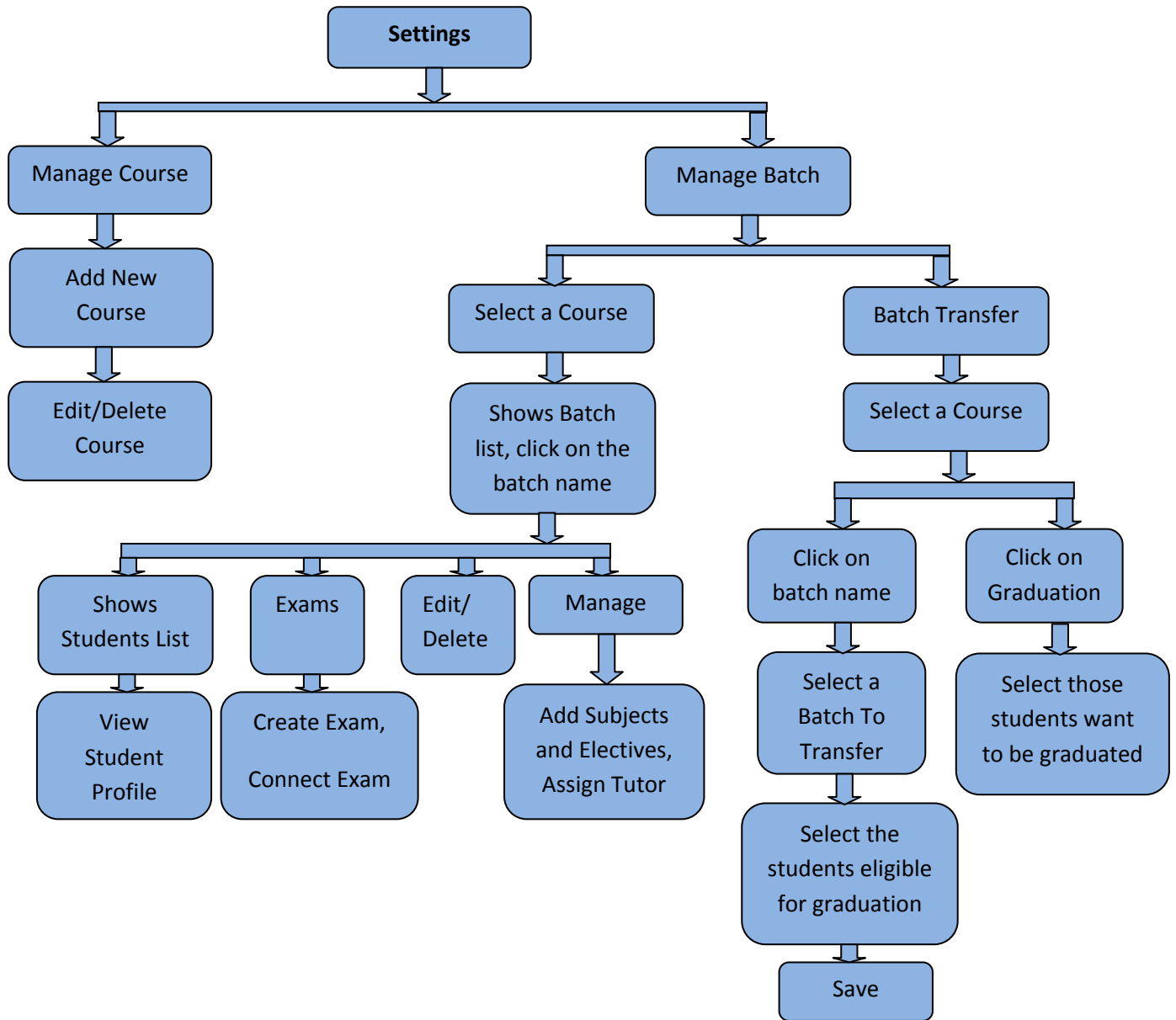
**edit additional details** : Edit the additional details entered for a student.

**delete additional details** : Delete additional details of a student  
**change to former** : Archive a Student while graduation / releaving from the institution  
**generate tc pdf** : Generate the TC for a student  
**generate all tc pdf** : Generate TC for all the students of the batch.  
**destroy** : Destroy a student record  
**edit** : Edit a Student  
**edit guardian** : Edit the guardian details  
**email** : To send the email  
**exam report** : Generate report of an examination  
**update student result for examtype** : Update exam result for a particular exam.  
**previous year marks overview** : Overview of previous marks  
**reports**: To take the exam report  
**search ajax** :Ajax search for students  
**student annual overview** : Daw open flash chart for the entire year  
**subject wise report** : Daw open flash chart for the entire year  
**add guardian** : Add a Guardian student  
**list students by course**: List the students by course  
**profile**: Fetch all details to display the previous details  
**profile pdf** : Generate the PDF report of the student profile  
**show previous details** : Show previous institution details  
**show**: Method for fetching image data from database  
**guardians** View Guardians of a students  
**del guardian** Delete a Guardian of a student  
**academic pdf** Generate the PDF report for examination records of a batch for a student  
**categories** Add/View student categories  
**category delete** Delete a student category  
**category edit** Edit a student category  
**category update** Update a student category  
**view all** Fetch all active batches  
**advanced search** Student advanced search  
**advanced search pdf** PDF for the advanced search results  
**list batches** List all batches of a course  
**list\_doa\_year** List the date of admission of students  
**list\_dob\_year** List the date of birth of students  
**doa equal to update** Method for checking the date of admission entered  
**doa greater than update** Method for checking the date of admission entered  
**doa less than update** Method for checking the date of admission entered  
**dob equal to update** Method for checking the date of birth entered  
**dob greater than update** Method for checking the date of birth entered  
**dob less than update** Method for checking the date of birth entered  
**electives** : To find the elective group  
**assign all students**: Assign all students to a particular elective  
**assign students** :Assign students one by one to an elective.  
**unassign all students**: Unassign all students to a particular elective  
**unassign students**: Unassign students one by one to an elective.  
  
**fees** : To update the fees status of student table  
**fees details** : To fetch the student fees details  
**find students**: It is defined under private method ,it is applicable

**Once an Student is releaved from the institution his/her details are moved into the ArchivedStudnets table with same structre.Other details are archived in a similar fashion.**



## Course and Batch Management



## **Models and Databases**

### **Course**

**id**(Integer)  
**course\_name**(String)  
**code**(String)  
**section\_name**(String)  
**is\_deleted**(Boolean)  
**created\_at**(Datetime)  
**updated\_at**(Datetime)  
**grading\_type**(String)

### **Batch**

**id**(Integer)  
**name**(String)  
**course\_id**(Integer)  
**start\_date**(Date)  
**end\_date**(Date)  
**is\_deleted**(Boolean)  
**employee\_id**(Integer)

## **Controllers**

### **CoursesController**

**index** *To find the active courses*  
**new** *To create the new course*  
**manage\_course** *To manage the course*  
**grouped\_batches** *To get the exam report of more than one batch group batch is using.*  
**create\_batch\_group** *To create the batch group*  
**edit\_batch\_group** *To edit the batch group*  
**update\_batch\_group** *To update the batch group*  
**delete\_batch\_group** *To delete the batch group*  
**update\_batch** *To update the batch*  
**create** *To create the new course*  
**edit** *To edit the course*  
**update** *To update the course*  
**destroy** *To delete the course*  
**show** *To view the active batches*

### **Batchescontroller**

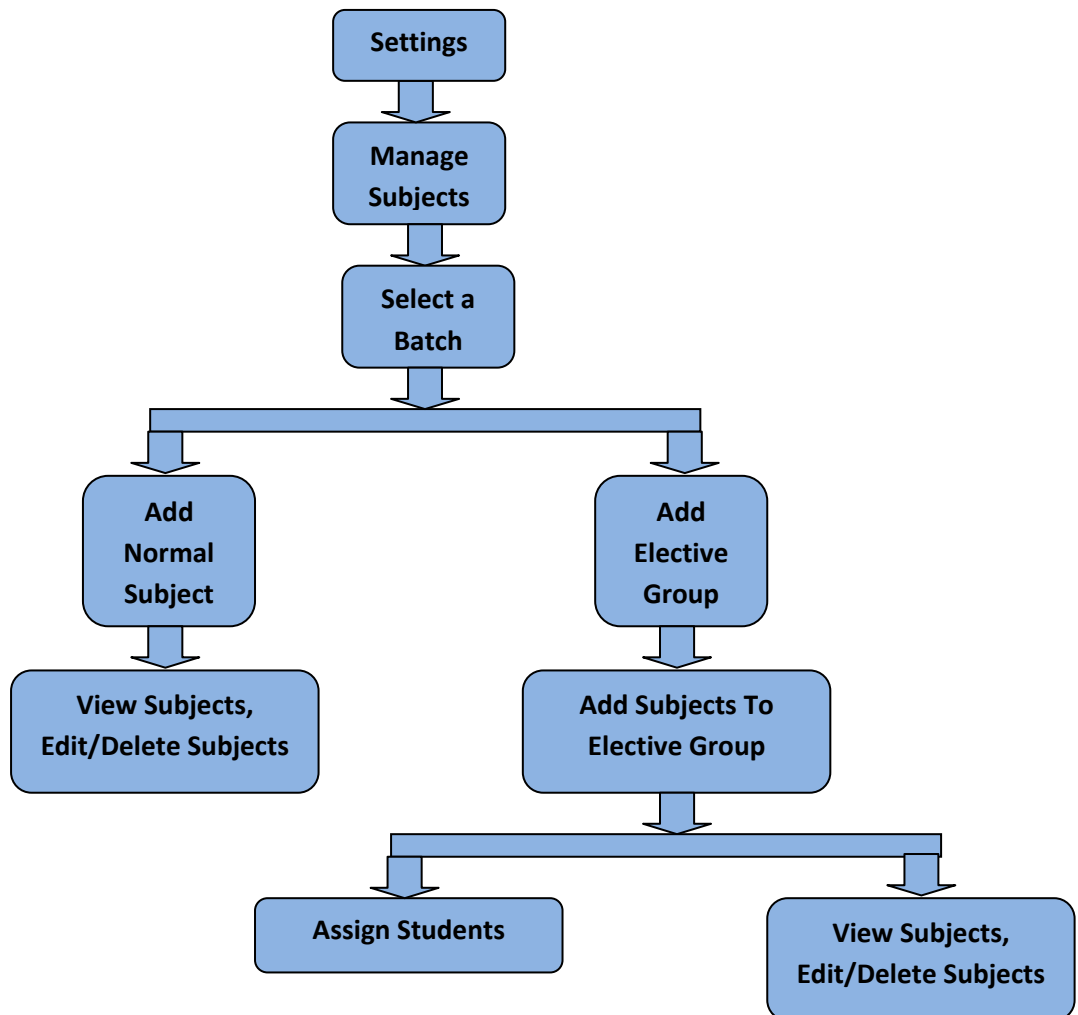
**index** *To find the batch*  
**new** *To create the new batch*  
**create** *To create the new batch*  
**update** *To update the batch*  
**show** *To show the batch*  
**destroy** *To destroy the batch*  
**assign\_tutor** *To assign the tutor*  
**update\_employees** *To update the employees*

**assign\_employee** *To assign employees*  
**remove\_employee** *To remove employee*  
**batches\_ajax** *To find the active batches*

### **BatchTransferController**

**index** *To find the active batches*  
**show** *To view the batches*  
**transfer** *To transfer the students*  
**graduation** *The students to be graduated to another batch*  
**subject\_transfer** *Transfer subjects to another batch*  
**get\_previous\_batch\_subjects** *To get the previous batch subjects*  
**update\_batch** *To update the batch*  
**assign\_previous\_batch\_subject** *To assign previous batch subject*  
**assign\_all\_previous\_batch\_subjects** *To get all previous batch subjects*  
**new\_subject** *To create the new subject*  
**create\_subject** *To create the subject*

### **Subject Management**



## **Models and Databases**

### **Subject**

**id**(Integer)  
**name**(String)  
**code**(String)  
**batch\_id**(Integer)  
**no\_exams**(Boolean)  
**max\_weekly\_classes**(Integer)  
**elective\_group\_id**(Integer)  
**is\_deleted**(Boolean)  
**created\_at**(Datetime)  
**updated\_at**(Datetime)  
**credit\_hours**(Decimal)  
**prefer\_consecutive**(Boolean)  
**amount**(Decimal)

### **StudentSubject**

**id**(Integer)  
**Subject\_id**(Integer)  
**Student\_id**(Integer)  
**Batch\_id**(Integer)

### **StudentPreviousSubjectMark**

**id**(Integer)  
**student\_id**(Integer)  
**subject**(String)  
**mark**(String)

### **SubjectLeave**

**id**(Integer)  
**student\_id**(Integer)  
**month\_date**(Date)  
**subject\_id**(Integer)  
**employee\_id**(Integer)  
**class\_timing\_id**(Integer)  
**reason**(String)  
**created\_at**(Datetime)  
**updated\_at**(Datetime)  
**batch\_id**(Integer)

## **Controllers**

### **ElectiveGroupsController**

**index** *To find the elective groups for the batches*

**new** *To create the new elective group*

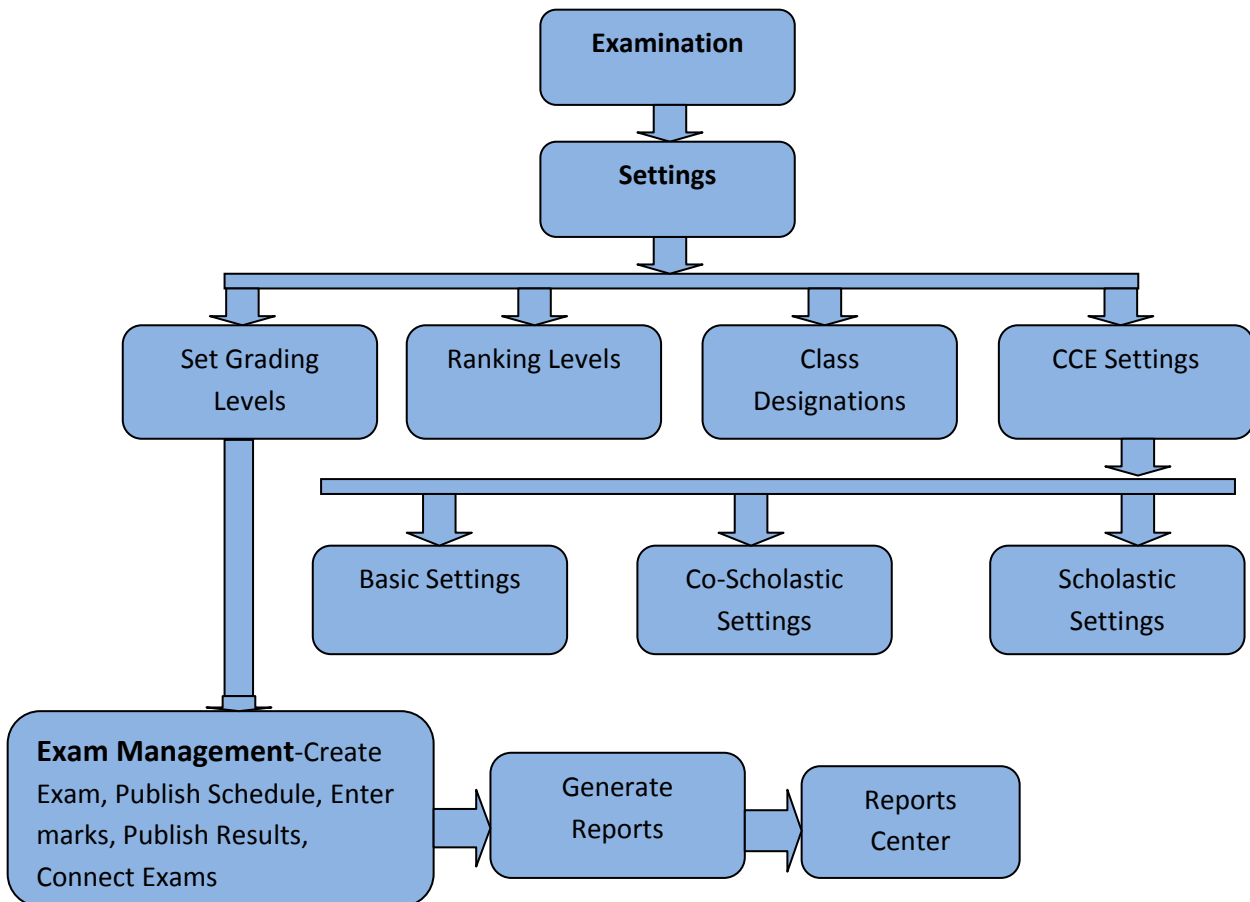
**create** To create the new elective group  
**edit** To edit the elective group  
**update** To update the elective group  
**destroy** To delete the elective group  
**pre\_load\_objects** It is a private method

**SubjectsController**

**index** To find the active batch  
**new** To create the new subject  
**create** To create the new subject  
**edit** To edit the subject  
**update** To updatet the subjects  
**destroy** To delete the subjects  
**show** To view the subjects

**Examination Management**

An exam group is common for all the exams during a term/set of exams. An exam group has an entry Exam for each of the subjects in exam table. Exam score for each student in each exam goes into details go into exam score table.



## **Models and Databases**

### **Examinations**

**Exam**  
**ExamGroup**  
**ExamScore**  
**GradingLevel**  
**RankingLevel**  
**ClassDesignation**  
**CceExamCategory**  
**CceGrade**  
**CceGradeSet**  
**CceReport**  
**CceWeightage**  
**FaCriteria**  
**FaGroup**  
**Observation**  
**ObservationGroup**  
**AssesmentScore**  
**DescriptiveIndicator**

### **Exam**

**id**( Integer )  
**exam\_group\_id**( Integer )  
**subject\_id**( Integer )  
**start\_time**( Datetime )  
**end\_time**( Datetime )  
**maximum\_marks**( Integer )  
**minimum\_marks**( Integer )  
**grading\_level\_id**( Integer )  
**weightage**( Integer )  
**event\_id**( Integer )  
**created\_at**( Datetime )  
**updated\_at**( Datetime )

### **ExamGroup**

**id**( Integer )  
**name**( String )  
**batch\_id**( Integer )  
**exam\_type**( String )  
**is\_published**( Boolean )  
**result\_published**( Boolean )  
**exam\_date**(date)  
**is\_final\_exam**(Boolean)  
**cce\_exam\_category**(Integer)

### **ExamScore**

**id**( Integer )  
**student\_id**( Integer )

**exam\_id**( Integer )  
**marks**( Integer )  
**grading\_level\_id**( Integer )  
**remarks**( String )  
**is\_failed**( Boolean )  
**created\_at**( Datetime )  
**updated\_at**( Datetime )

### **GradingLevel**

**id**( Integer )  
**name**( String )  
**batch\_id**( Integer )  
**min\_score**( Integer )  
**order**( Integer )  
**is\_deleted**( Boolean )  
**created\_at**( Datetime )  
**updated\_at**( Datetime )  
**credited\_points**(Decimal)  
**description**(string)

### **RankingLevel**

**id**(Integer)  
**name**(String)  
**gpa**(Decimal)  
**marks**(Decimal)  
**subject\_count**(Integer)  
**priority**(Integer)  
**created\_at**(Datetime)  
**updated\_at**(Datetime)  
**full\_course**(Boolean)  
**course\_id**(Integer)  
**subject\_limit\_type**(String)  
**marks\_limit\_type**(String)

### **ClassDesignation**

**id**(Integer)  
**name**(String)  
**cgpa**(Decimal)  
**created\_at**(Datetime)  
**updated\_at**(Datetime)  
**marks**(Decimal)  
**course\_id**(Integer)

### **CceExamCategory**

**id**(Integer)  
**name**(String)  
**desc**(String)  
**created\_at**(Datetime)  
**updated\_at**(Datetime)

### **CceGrade**

**id**(Integer)  
**name**(String)  
**grade\_point**(Float)  
**cce\_grade\_set\_id**(Integer)  
**created\_at**(Datetime)  
**updated\_at**(Datetime)

### **CceGradeSet**

**id**(Integer)  
**name**(String)  
**created\_at**(Datetime)  
**updated\_at**(Datetime)

### **CceReports**

**id**(Integer)  
**observable\_id**(Integer)  
**observable\_type**(String)  
**student\_id**(Integer)  
**batch\_id**(Integer)  
**grade\_string**(String)  
**created\_at**(Datetime)  
**updated\_at**(Datetime)  
**exam\_id**(Integer)

### **CceWeightage**

**id**(Integer)  
**weightage**(Integer)  
**criteria\_type**(String)  
**cce\_exam\_category\_id**(Integer)  
**created\_at**(Datetime)  
**updated\_at**(Datetime)

### **CceWeightagesCourse**

**cce\_weightage\_id**(Integer)  
**course\_id**(Integer)

### **FaCriterias**

**id**(Integer)  
**fa\_name**(String)  
**desc**(String)  
**fa\_group\_id**(Integer)  
**created\_at**(Datetime)  
**updated\_at**(Datetime)  
**sort\_order**(Integer)  
**is\_deleted**(Boolean)



### **FaGroup**

**id**(Integer)  
**name**(String)  
**desc**(Text)  
**cce\_exam\_category\_id**(Integer)  
**created\_at**(Datetime)  
**updated\_at**(Datetime)  
**cce\_grade\_set\_id**(Integer)  
**max\_marks**(Float)  
**is\_deleted**(Boolean)

### **FaGroupsSubject**

**Subject\_id**(Integer)  
**fa\_group\_id**(Integer)

### **Observation**

**id**(Integer)  
**name**(String)  
**desc**(String)  
**is\_active**(Boolean)  
**observation\_group\_id**(Integer)  
**created\_at**(Datetime)  
**updated\_at**(Datetime)  
**sort\_order**(Integer)

### **ObservationGroup**

**id**(Integer)  
**name**(String)  
**header\_name**(String)  
**desc**(String)  
**cce\_grade\_set\_id**(Integer)  
**created\_at**(Datetime)  
**updated\_at**(Datetime)  
**observation\_kind**(String)  
**max\_marks**(Float)  
**is\_deleted**(Boolean)

### **AssessmentScore**

**id**(Integer)  
**sstudent\_id**(Integer)  
**grade\_points**(Integer)  
**created\_at**(Datetime)  
**updated\_at**(Datetime)  
**exam\_id**(Integer)  
**batch\_id**(Integer)  
**descriptive\_indicator\_id**(Integer)

## Action / Methods

### ExamGroupsController

**create** *Create an exam group*

**destroy** *To delete the exam group*

**edit** *To Edit the exam group*

**index** *To view the exams*

**new\_** *To create the exams*

**show**

**initial\_queries** *This is under a private method,we can use it to other actions.*

**protect\_other\_batch\_exams** *This is under a private method,we can use it to other actions.*

### ExamsController

**new\_** *Create the new exams*

**create\_** *Create the new exams*

**edit** *Edit the exams*

**update** *Update the exams*

**show** *Show the exams*

**destroy** *To delete the exams*

**save\_scores** *Save score for an exam for a student*

**query\_data** *Query details for the currently requested Exam Object ,it is a private method*

**restrict\_employees\_from\_exam\_edit** *Restrict the employees while editing the exams, it is a private method*

### ExamScoresController

**in\_place\_edit\_for** :exam\_score, :score

### ExamController

**update\_exam\_form** *To create the new exam group*

**publish** *To publish the exam schedule*

**grouping** **Grouping** *the exams to connect*

**list\_batch\_groups** *List the batch groups*

**generate\_previous\_reports** *To generate the previous reports*

**select\_inactive\_batches** *To select the inactive batches*

**generate\_reports** *Generate the report of the particular batch*

**exam\_wise\_report** *To get the exam wise rpeort*

**list\_exam\_types** *List all exam types*

**generated\_report** *To generate the exam report*

**generated\_report\_pdf** *To take the exam report pdf*

**consolidated\_exam\_report** *To get consolidated exam report*

**consolidated\_exam\_report\_pdf** *To take the consolidated exam report pdf*

**subject\_rank** *Student ranking per subject*

**list\_batch\_subjects** *List the subjects based on the ranking*

**student\_subject\_rank** *To get the student ranking per subject*

**student\_subject\_rank\_pdf** *To get the pdf of student ranking per subject*

**subject\_wise\_report** *To take the subject wise report*

**list\_subjects** *To list the subjects*

**generated\_report2** *To get the generated report*

**generated\_report2\_pdf** *To get the generated report pdf*  
**student\_batch\_rank** *To get the student batch rank*  
**student\_batch\_rank\_pdf** *To get the student batch rank pdf*  
**course\_rank** *To find the course rank*  
**batch\_groups** *To find the batch groups*  
**student\_course\_rank** *To find the student course rank*  
**student\_course\_rank\_pdf** *To find the student course rank pdf*  
**student\_school\_rank** *To find the student school rank*  
**student\_school\_rank\_pdf** *To find the student school rank pdf*  
**student\_attendance\_rank** *To find the student attendance rank*  
**student\_attendance\_rank\_pdf** *To find the student attendance rank pdf*  
**ranking\_level\_report** *To take the ranking level report*  
**select\_mode** *To select the modes*  
**select\_batch\_group** *To select the batch groups*  
**select\_type** *To select the report type*  
**student\_ranking\_level\_report** *To find the student ranking level report*  
**student\_ranking\_level\_report\_pdf** *To find the student ranking level report pdf*  
**transcript** *To find the transcript pdf*  
**Student\_transcript** *To find the student transcript*  
**Student\_transcript\_pdf** *To find the student transcript pdf*  
**load\_batch\_students** *To load the batch students*  
**combined\_report** *to find the combined report of the students*  
**load\_levels** *To load the ranking levels*  
**student\_combined\_report** *To find the combined report of the student*  
**student\_combined\_report\_pdf** *To find the combined report of the student pdf*  
**select\_report\_type** *To select the exam report type*  
**generated\_report3** *To find the academic report*  
**final\_report\_type** *To find the final exam report type*  
**generated\_report4** *To find the grouped exam type*  
**generated\_report4\_pdf** *To find the grouped exam type pdf*  
**combined\_grouped\_exam\_report\_pdf** *To find the combined grouped exam report pdf*  
**previous\_years\_marks\_overview** *To find previous year marks*  
**previous\_years\_marks\_overview\_pdf** *To find previous year marks pdf*  
**academic\_report** *To find the academic report*  
**previous\_batch\_exams** *To find the previous batch exams*  
**list\_inactive\_batches** *To find the list of inactive batches*  
**list\_inactive\_exam\_groups** *To find the list of inactive exam groups*  
**previous\_exam\_marks** *To find the previous exam marks*  
**edit\_previous\_marks** *To edit the previous marks*  
**update\_previous\_marks** *To update the previous marks*  
**create\_exam** *To create the exam*  
**update\_batch\_ex\_result** *To update the batch exam result*  
**update\_batch** *To update the batch*  
**graph\_for\_generated\_report** *To find the graph of the generated exam report*  
**graph\_for\_generated\_report3** *To find the graph of the exam report*  
**graph\_for\_previous\_marks\_overview** *Graph of the previous marks*

### **CceExamCategoriesController**

**index** *To find the cce exam category*  
**new** *To create the cce exam category*  
**create** *To create the cce exam category*  
**edit** *To edit the cce exam category*  
**update** *To update the cce exam category*

**destroy** *To delete the cce exam category*

### **CceGradeSetsController**

**index** *To find the cce grade set*  
**new** *To create the cce grade set*  
**create** *To create the cce grade set*  
**edit** *To edit the cce grade set*  
**update** *To update the cce grade set*  
**show** *To show the cce grade set*  
**destroy** *To delete the cce grade set*  
**new\_grade** *To create the cce grade*  
**create\_grade** *To create the cce grade*  
**edit\_grade** *To edit the cce grade*  
**destroy\_grade** *To delete the cce grade*

### **CceReportsController**

**index**  
**create\_reports** *To generate cce report*  
**student\_wise\_report** *To find the student wise report*  
**student\_report** *To find the student report*  
**student\_report\_pdf** *To find the student report pdf*  
**student\_transcript** *To find the student transcript report*  
**fetch\_report** *To fetch the report*

### **CceWeightagesController**

**index** *To find the cce weightages*  
**new** *To create the cce weightage*  
**create** *To create the cce weightage*  
**show** *To show the cce weightage*  
**edit** *To edit the cce weightage*  
**update** *To update the cce weightage*  
**destroy** *To delete the cce weightage*  
**assign\_courses** *To assign course*  
**assign\_weightages** *To assign cce weightages*  
**select\_weightages** *To select the cce weightage*  
**update\_course\_weightages** *To update the course weightages*

### **ClassDesignationsController**

**load\_class\_designations** *To load class designation form*  
**create\_class\_designation** *To create the class designation*  
**edit\_class\_designation** *To edit the class designation*  
**update\_class\_designation** *To update the class designation*  
**delete\_class\_designation** *To delete the class designation*

### **FaCriteriasController**

**index** *To find the fa criterias*  
**show** *To show the fa criterias*

### **FaGroupsController**

**index** *To find the active fa groups*  
**new** *To create the fa group*  
**create** *To create the fa group*  
**show** *To show the fa group*  
**edit** *To edit the fa group*  
**update** *To update the fa group*  
**destroy** *To delete the fa group*  
**assign\_fa\_groups** *To assign fa group*  
**select\_fa\_groups** *To select fa group*  
**new\_fa\_criteria** *To create the new fa criteria*  
**create\_fa\_criteria** *To create the new fa criteria*  
**edit\_fa\_criteria** *To edit the fa criteria*  
**update\_fa\_criteria** *To update the fa criteria*  
**destroy\_fa\_criteria** *To delete the fa criteria*  
**reorder** *Reorder the fa criteria*

### **ObservationGroupsController**

**index** *To find the observation groups*  
**new** *To create the new observation groups*  
**create** *To create the new observation groups*  
**show** *To show the observation groups*  
**edit** *To edit the observation groups*  
**update** *To update the observation groups*  
**destroy** *To delete the observation groups*  
**new\_observation** *To create the new observation*  
**create\_observation** *To create the new observation*  
**edit\_observation** *To edit observation*  
**update\_observation** *To update the observation*  
**assign\_courses** *To assign courses*  
**update\_course\_obs\_group** *To update the course observation groups*  
**destroy\_observation** *To destroy observation*  
**reorder** *To reorder the observation*

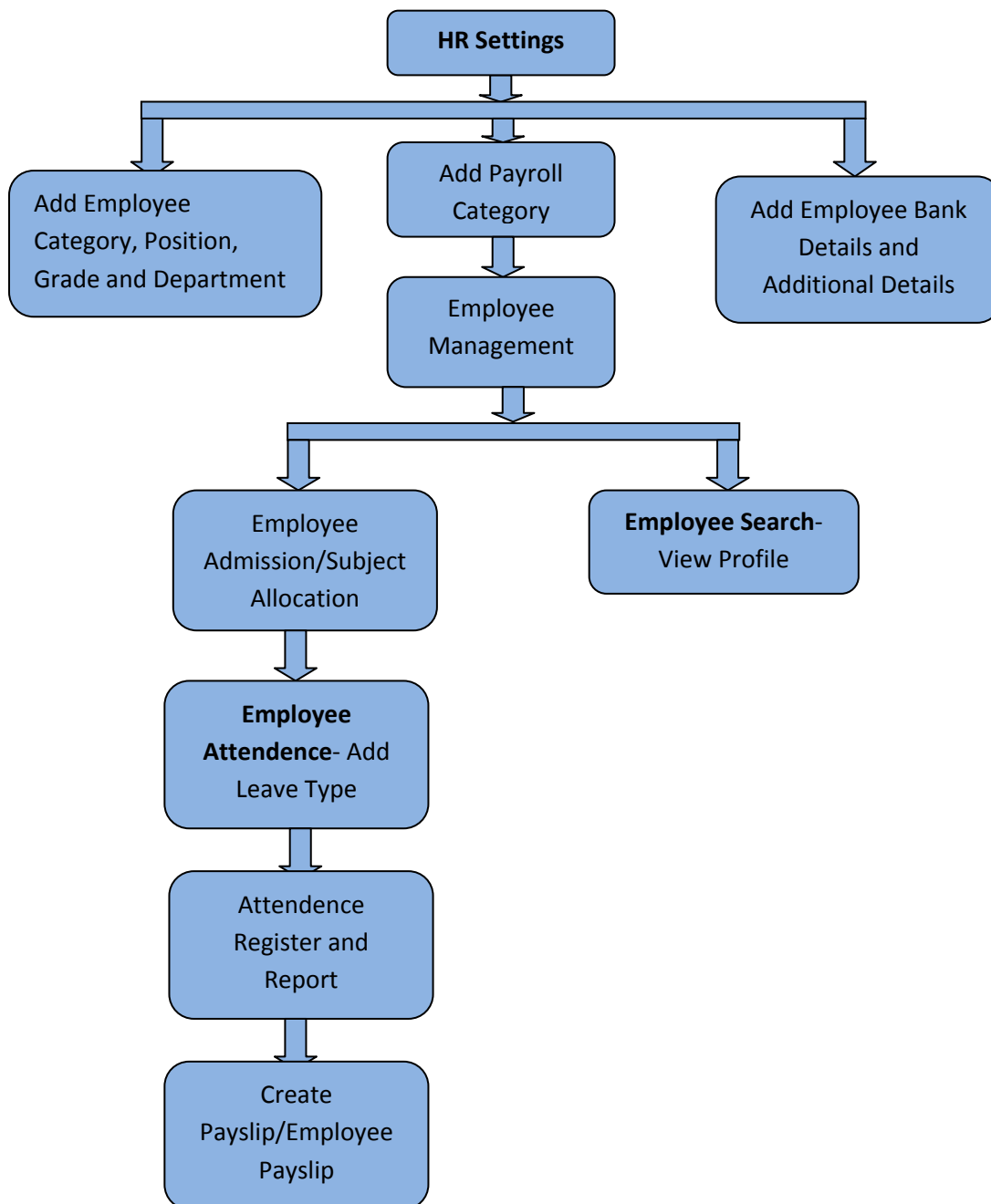
### **ObservationController**

**show** *To show the observations*

### **RankingLevelsController**

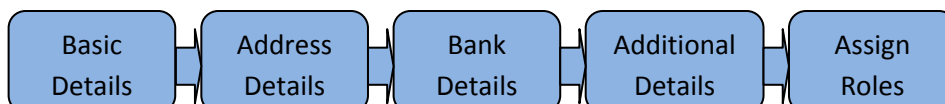
**load\_ranking\_levels** *To load ranking level form*  
**create\_ranking\_level** *To create the ranking level*  
**edit\_ranking\_level** *To edit the ranking level*  
**update\_ranking\_level** *To update the ranking level*  
**delete\_ranking\_level** *To delete the ranking level*  
**ranking\_level\_cancel** *To cancel the ranking levels*  
**change\_priority** *To change the priority*

## Human Resource module



Human Resource module is used to manage Employees, Payroll & Payslip, Employee subject Association, Employee Attendance

### Steps in Employee Admission



## **Models/Database Structure**

**Employee**  
**EmployeeAdditionalDetail**  
**EmployeeAttendance**  
**EmployeeBankDetail**  
**EmployeeCategory**  
**EmployeeDepartment**  
**EmployeeDepartmentEvent**  
**EmployeeGrade**  
**EmployeeLeave**  
**EmployeeLeaveType**  
**EmployeePosition**  
**EmployeeSalaryStructure**  
**EmployeesSubject**  
**IndividualPayslipCategory**  
**MonthlyPayslip**  
**PayrollCategory**  
**BankField**  
**ApplyLeave**

### **Employee**

**id**(Integer)  
**employee\_category\_id**(Integer)  
**employee\_number**(String)  
**joining\_date**: date  
**first\_name**(String)  
**middle\_name**(String)  
**last\_name**(String)  
**gender**: boolean  
**job\_title**(String)  
**employee\_position\_id**(Integer)  
**employee\_department\_id**(Integer)  
**reporting\_manager\_id**(Integer)  
**employee\_grade\_id**(Integer)  
**qualification**(String)  
**experience\_detail**: text  
**experience\_year**(Integer)  
**experience\_month**(Integer)  
**status**: boolean  
**status\_description**(String)  
**date\_of\_birth**: date  
**marital\_status**(String)  
**children\_count**(Integer)  
**father\_name**(String)  
**mother\_name**(String)  
**husband\_name**(String)  
**blood\_group**(String)  
**nationality\_id**(Integer)  
**home\_address\_line1**(String)  
**home\_address\_line2**(String)  
**home\_city**(String)

**home\_state**(String)  
**home\_country\_id**(Integer)  
**home\_pin\_code**(String)  
**office\_address\_line1**(String)  
**office\_address\_line2**(String)  
**office\_city**(String)  
**office\_state**(String)  
**office\_country\_id**(Integer)  
**office\_pin\_code**(String)  
**office\_phone1**(String)  
**office\_phone2**(String)  
**mobile\_phone**(String)  
**home\_phone**(String)  
**email**(String)  
**fax**(String)  
**photo\_filename**(String)  
**photo\_content\_type**(String)  
**photo\_data**( Binary )  
**created\_at**( DateTime )  
**updated\_at**( DateTime )  
**photo\_file\_size**(Integer)  
**user\_id**(Integer)

### **EmployeeAdditionalDetail**

**id**: integer  
**employee\_id**: integer  
**additional\_field\_id**: integer  
**additional\_info**: string The information of the employee ( employee\_id ) is stored here corresponding to the additional field (additional\_field\_id)

### **EmployeeBankDetail**

**id**: integer  
**employee\_id**: integer  
**bank\_field\_id**: integer  
**bank\_info**: string

### **EmployeeSalaryStructure**

**id**: integer  
**employee\_id**: integer  
**payroll\_category\_id**: integer  
**amount**: string

### **EmployeesSubject**

**id**: integer  
**employee\_id**: integer  
**subject\_id**(Integer)

### **EmployeeAttendances**

**Id**(Integer)



**attendance\_date**(Date)  
**employee\_id**(Integer)  
**employee\_leave\_type\_id**(Integer)  
**reason**(String)  
**is\_half\_day**(Boolean)

### **EmployeeCategory**

**id**(Integer)  
**name**(String)  
**prefix**(String)  
**status**(Boolean)

### **EmployeeDepartment**

**id**(Integer)  
**code**(String)  
**names**(String)  
**status**(Boolean)

### **EmployeeDepartmentEvent**

**id**(Integer)  
**event\_id**(Integer)  
**employee\_department\_id**(Integer)  
**created\_at**(Datetime)  
**updated\_at**(Datetime)

### **EmployeeLeave**

**id**(Integer)  
**employee\_id**(Integer)  
**employee\_leave\_type\_id**(Integer)  
**leave\_count**(Decimal)  
**leave\_taken**(Decimal)  
**reset\_date**(Date)  
**created\_at**(Datetime)  
**updated\_at**(Datetime)

### **EmployeeLeaveType**

**id**(Integer)  
**name**(String)  
**code**(String)  
**status**(Boolean)  
**max\_leave\_count**(String)  
**carry\_forward**(Boolean)

### **IndividualPayslipCategory**

**id**(Integer)  
**employee\_id**(Integer)  
**salary\_date**(Date)  
**name**(String)  
**amount**(String)

is\_deduction(Boolean)  
include\_every\_month(Boolean)

### **MonthlyPayslip**

id(Integer)  
salary\_date(Date)  
employee\_id(Integer)  
payroll\_category\_id(Integer)  
amount(String)  
is\_approved(Boolean)  
approver\_id(Integer)  
is\_rejected(Boolean)  
rejected\_id(Integer)  
reason(String)  
remark(String)

### **PayrollCategory**

id(Integer)  
name(String)  
percentage(Float)  
payroll\_category\_id(Integer)  
is\_deduction(Boolean)  
status(Boolean)

### **BankField**

id(Integer)  
name(String)  
status(Boolean)

### **ApplyLeave**

id(Integer)  
employee\_id(Integer)  
employee\_leave\_types\_id(Integer)  
is\_half\_day(Boolean)  
start\_date(Date)  
end\_date(Date)  
reason(String)  
approved(Boolean)  
viewed\_by\_manager(Boolean)  
manager\_remark(String)

***Once an Employee is relieved from the institution his/her details are moved into the ArchivedEmployees table with same structure. Other details are archived in a similar fashion, for example his details from EmployeeBankDetail are moved into ArchivedEmployeeBankDetailTable.***

## **Methods and actions**

### **EmployeeController**

**add\_catgeory** *To create the employee category*  
**edit\_category** *To edit the employee category*  
**delete\_catgeory** *To delete the employee category*  
**add\_position** *To add employee position*  
**edit\_position** *To edit the employee position*  
**delete\_position** *To delete the employe position*  
**add\_department** *To add employee department*  
**edit\_department** *To edit the employee department*  
**delete\_department** *To delete the employee department*  
**add\_grade** *To add the employee grade*  
**edit\_grade** *To edit the employee grade*  
**delete\_grade** *To delete the employee grade*  
**add\_bank\_details** *To add the bank details*  
**edit\_bank\_details** *To edit the bank details*  
**delete\_bank\_details** *To delete the bank details*  
**add\_additional\_details** *To add the additional details*  
**edit\_additional\_details** *To edit the additional details*  
**delete\_additional\_details** *To delete the additional details*  
**admission1** *First form of employee admission,contain the basic details*  
**update\_positions** *To update the employee positions*  
**edit1** *To edit the employee basic details*  
**edit\_personal** *To edit the employee personal details*  
**admission2** *Second form of employee admission*  
**edit2** *To edit the employee contact details*  
**edit\_contact** *To edit the employee contact details*  
**admission3** *Third form of employee admission*  
**edit3** *To edit the employee bank details*  
**admission3\_1** *To add the employee additional details*  
**edit\_privilege** *To edit the employee privilege*  
**edit3\_1** *To edi the employee additional details*  
**admission4** *The form to assign employee manager*  
**view\_rep\_manager** *To view the employee manager*  
**change\_reporting\_manager** *To change the employee manager*  
**update\_reporting\_manager\_name** *To update the employee manager name*  
**search** *To search the employees*  
**search\_ajax** *To search the employees*  
**profile** *To view the employee profile*  
**profile\_general** *To view the employee general details*  
**profile\_personal** *To view employee personal details*  
**profile\_address** *To view employee address*  
**profile\_contact** *To view employee contact details*  
**profile\_bank\_details** *To view employee bank details*  
**profile\_additional\_details** *To view employee additional details*  
**profile\_payroll\_details** *To view employee payroll details*  
**profile\_pdf** *To take the employee profile pdf*  
**view\_all** *To view all employees*  
**employee\_list** *To show employee list*  
**show** *To show employees*  
**add\_payslip\_category** *To add employee payslip category*

**create\_payslip\_category** *To create employee payslip category*  
**remove\_new\_paylist\_catgeory** *To remove the employee payroll category*  
**create\_monthly\_payslip** *To create the employee monthly payslip*  
**view\_payslip** *To view the employee payslip*  
**update\_monthly\_payslip** *To update the employee payslip*  
**delete\_payslip** *To delete the employee payslip*  
**view\_attendance** *To view employee attendance*  
**employee\_leave\_count\_edit** *To edit the employee leave count*  
**employee\_leave\_count\_update** *To update the employee leave count*  
**subject\_assignment** *To assign the subject to the employees*  
**update\_subjects** *To update the subjects*  
**select\_department** *To select the employee departments*  
**update\_employees** *To update the employees*  
**assign\_employees** *To assign the employees*  
**remove\_employees** *To remove the employees*  
**timetable** *To view the employee timetable*  
**timetable\_pdf** *To take the employee timetable pdf*  
**hr** *To get the employee dashboard*  
**select\_employee\_department** *Select the employees from the departments*  
**rejected\_payslip** *To reject the employee payslip*  
**update\_rejected\_employee\_paylsip** *To update the rejected employee pasylip*  
**edit\_rejected\_payslip** *To edit the rejected payslip*  
**update\_rejected\_payslip** *To update the rejected payslip*  
**view\_rejected\_payslip** *To view the rejected payslip*  
**update\_employee\_select\_list** *To update the employees*  
**payslip\_date\_select** *To select the payslip date*  
**one\_click\_payslip\_generation** *To generate the payslip on one click*  
**payslip\_revert\_date\_select** *To select the payslip revert date*  
**one\_click\_payslip\_revert** *To revert the payslip in one click*  
**leave\_management** *To manage the employee leaves*  
**all\_employee\_leave\_applications** *Employee leave applications*  
**update\_employees\_select** *To update the employees*  
**leave\_list** *To list the leaves*  
**department\_payslip** *To find the department payslip*  
**view\_employee\_payslip** *To view the employee payslip*  
**view\_employee\_pasylip\_pdf** *To take the employee payslip pdf*  
**individual\_payslip\_pdf** *To take the individual employee payslip*  
**employee\_individual\_payslip\_pdf** *To take the pdf payslip of individual employee*  
**advanced\_search** *To search the employees*  
**list\_doj\_year** *To list the joining dates*  
**doj\_equal\_to\_update** *To select the joining date equals to*  
**doj\_less\_than\_update** *To select the joining date less than*  
**doj\_greater\_than\_update** *To select the joining date greater than*  
**list\_dob\_year** *To list the date of birth*  
**dob\_equal\_to\_update** *To select the date of birth equals to*  
**dob\_less\_than\_update** *To select the date of birth less than*  
**dob\_greater\_than\_update** *To select the date of birth greater than*  
**remove** *To remove the employee*  
**remove\_subordinate\_employee** *To remove the subordinate employee*  
**change\_to\_former** *To change the employee to former*  
**delete** *To delete the employee*  
**advanced\_search\_pdf** *To take the advanced search pdf*  
**payslip\_approve** *To approve the employee payslip*  
**one\_click\_approve** *To approve the employee\_paysli p*  
**one\_click\_approve\_submit** *To submit one click approve*

## **EmployeeAttendanceController**

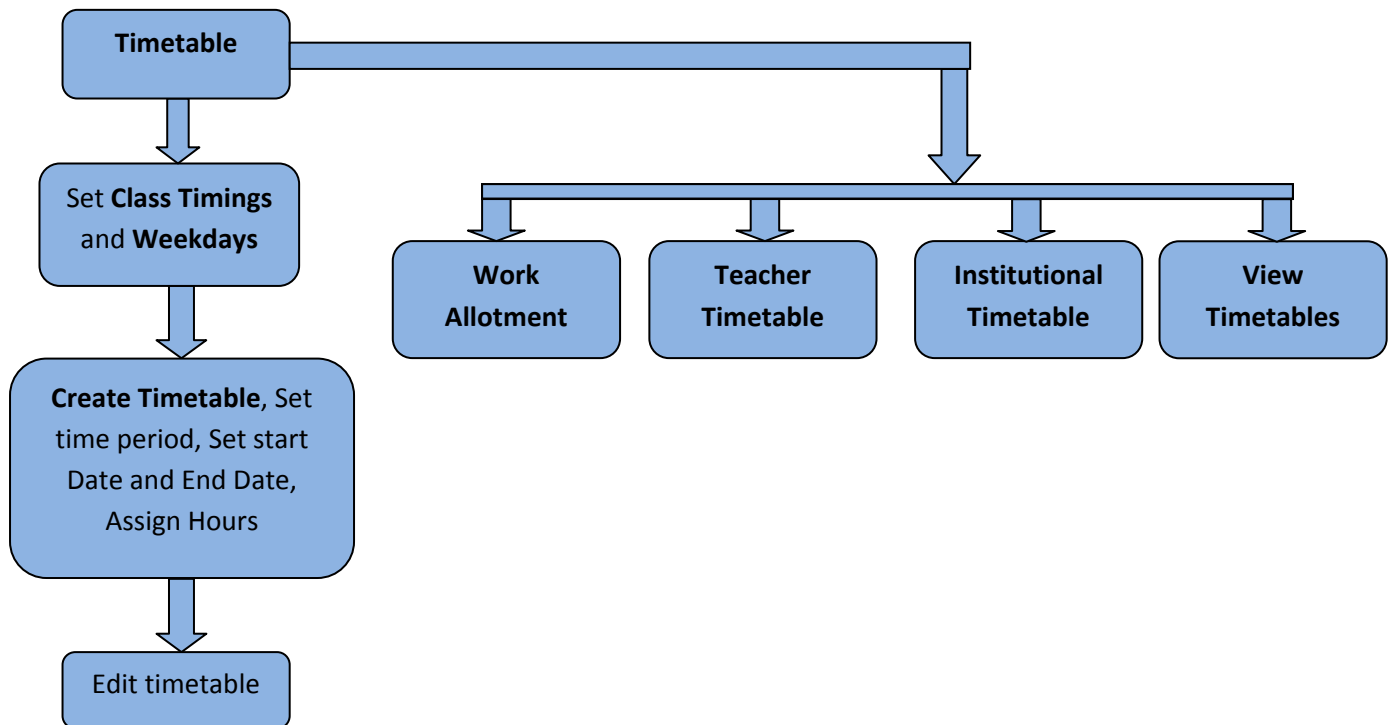
**add\_leave\_types** *To add employee leave types*  
**edit\_leave\_types** *To edit the employee leave types*  
**delete\_leave\_types** *To delete the employee leave types*  
**leave\_reset\_settings** *To reset the leaves*  
**update\_employee\_leave\_reset\_all** *To reset all employee leaves*  
**employee\_leave\_reset\_by\_department** *To reset the employee leave by department wise*  
**list\_department\_leave\_reset** *To list the employee department leave reset*  
**update\_department\_leave\_reset** *To update the department leave reset*  
**employee\_leave\_reset\_by\_employee** *Reset the employee leaves*  
**employee\_search\_ajax** *To search the employees*  
**employee\_view\_all** *To view all the employees*  
**employee\_list** *To list the employees*  
**employee\_leave\_details** *To find the employee leave details*  
**employee\_wise\_leave\_reset** *To reset the leave employee wise*  
**register** *To create the employee attendance*  
**update\_attendance\_form** *To update the attendance form*  
**report** *To get the attendance report*  
**update\_attendance\_report** *To update the attendance report*  
**emp\_attendance** *To find the employee attendance*  
**leave\_history** *To find the leave history*  
**update\_leave\_history** *To update the leave history*  
**leaves** *To find the leaves*  
**leave\_application** *To apply the leave*  
**leave\_app** *To apply the leave*  
**approve\_remarks** *To approve remarks*  
**deny\_remarks** *To deny the remarks*  
**approve\_leave** *To approve the leave*  
**deny\_leave** *To deny the leave*  
**cancel** *To cancel the leave*  
**new\_leave\_applications** *To create the nbew leave application*  
**all\_employee\_new\_leave\_applications** *To create the leave application of all employees*  
**all\_leave\_applications** *To find all leave applications*  
**individual\_leave\_applications** *To find the individual employee leaves*  
**own\_leave\_application** *To find the applied leave*  
**cancel\_application** *To cancel the leave application*  
**update\_all\_application\_view** *To update all employee leave application*  
**employee\_attendance\_pdf** *To take the employee attendance pdf*

## **EmployeeAttendencesController**

**Index** *To find the employee attendeces*  
**show** *To show the employee attendences*  
**new** *To create the employee attendance*  
**create** *To create the employee attendance*  
**edit** *To edit the employee attendance*  
**update** *To update the employee attendece*  
**destroy** *To delete the employee leaves*

## TimeTable Management

In timetable module weekdays and period timings are set at the beginning. Period timing and Weekdays can be set in common for all the batches in the institution. This *will be overridden* for a *batch* upon setting period or week day for that batch



### Associated models

***TimetableEntry***

***ClassTiming***

***Timetable***

***Weekdays***

### ***TimetableEntry***

**id:** integer

**batch\_id:** integer

**week\_day\_id:** integer

**class\_timing\_id:** integer

**subject\_id:** integer

**employee\_id:** integer

timetable\_id(Integer)

### ***ClassTiming***

**id:** integer  
**batch\_id:** integer  
**name:** string  
**start\_time:** time  
**end\_time:** time  
**is\_break:** boolean  
**is\_deleted:**(Boolean)

### **Weeday**

**Id**(Integer)  
**name**(String)  
**weekday**(String)  
**batch\_id**(Integer)  
**sort\_order**(Integer)  
**day\_of\_week**(Integer)  
**is\_deleted**(Boolean)

### **Timetable**

**id**(Integer)  
**start\_date**(Date)  
**end\_date**(Date)  
**is\_active**(Boolean)  
**created\_at**(Datetime)  
**updated\_at**(Datetime)

## **Method And Actions**

### **TimetableController**

**new\_timetable** *To create the new timetable*  
**update\_timetable** *To update the timetable*  
**view** *To view the timetable*  
**edit\_master** *To edit the timetable*  
**teachers\_timetable** *To find the teachers timetable*  
**update\_teachers\_tt** *To update the teachers timetable*  
**update\_timetable\_view** *To update the timetable view*  
**destroy** *To delete the timetable*

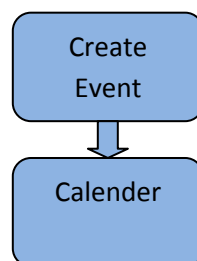
**employee\_timetable** *To find the employee timetable*  
**update\_employee\_tt** *To update the employee timetable*  
**students\_view** *Timetable view of students*  
**update\_students\_tt** *To update student timetable*  
**weekdays** *To find the weekdays*  
**timetable\_pdf** *To take the timetable pdf*  
**work\_allotment** *To allot the work to employees*  
**timetable** *To find the timetable*  
**delete\_blank** *It is inside the hash class*

### **TimetableEntriesController**

**new** *To create the new timetable entries*  
**select\_batch** *To select the batch*  
**new\_entry** *To create the new entry*  
**update\_employees** *To update the employees*  
**delete\_employee2** *To delete the employee*  
**update\_multiple\_timetable\_entries2** *To update the timetable entries*  
**tt\_entry\_update2** *To update the entries*  
**tt\_entry\_noupdate2** *To update the page*  
**tte\_from\_batch\_and\_tt** *It is a private method*

## **Events Module**

Events are created for various notifications and calendar entries . A new *Event* Object is created while a new exam is created or a fee is due.



### **Database Structure**

#### **Event**

**id:** integer  
**title:** string  
**description:** string  
**start\_date:** datetime  
**end\_date:** datetime  
**is\_common:** boolean  
**is\_holiday:** boolean  
**is\_exam:** boolean  
**created\_at:** datetime



**updated\_at:** datetime  
**is\_due:** Boolean  
**origin\_id**(Integer)  
**origin\_type**(String)

## **Method And Actions**

### **EventController**

**index** *To create the new event*  
**event\_group** *To find the event group*  
**select\_course** *To select the course*  
**course\_event** *To find the course event*  
**remove\_batch** *To remove batch*  
**select\_employee\_department** *To find the employee department*  
**department\_event** *To find the department event*  
**remove\_department** *To remove the department*  
**show** *To show the events*  
**confirm\_event** *To confirm the event*  
**cancel\_event** *To cancel event*  
**edit\_event** *To edit the event*

### **CalenderController**

**index** *To view all the events*  
**new\_calender** *To create the new calendar event*  
**show\_event\_tooltip** *To view the events*  
**show\_holiday\_event\_tooltip** *To view the holiday events*  
**show\_exam\_event\_tooltip** *To view the exam event*  
**show\_due\_tooltip** *To show the finance dues*  
**event\_delete** *To delete the event*  
**build\_common\_events\_hash** *It is a private method to find the common events*  
**build\_student\_events\_hash** *It is a private method method to find the students events*  
**build\_employee\_events\_hash** *It is a private method to find the employee events*  
**load\_notification** *To load notifications*

## **News management**

### **Models and databases**

#### **Associated models**

**News**  
**News\_comments**

#### **Database structure:**

##### **News**

**Id**(Integer)  
**title**(string)  
**author\_id**(integer)  
**content**(text)  
**created\_at**(date\_time)

updated\_at(date\_time)

### ***NewsComments***

id(Integer)

content(text)

news\_id(integer)

author\_id(integer)

created\_at(date\_time)

updated\_at(date\_time)

is\_approved(Boolean)

### **Controllers and views**

The following controllers are associated with the News management module:

#### **NewsController**

##### ***Views/methods associated with NewsController***

**index** index page for news with various options

**add** Publish school news

**add\_comment** : news/add\_comment -> adds comment to th published news

**all** shows all published news

**delete:** delete a news

**edit** edits a news

**search:** search for news

**view:**

### **Application Settings**

All the general settings related to fedena is stored in the table configuration

#### **Configuration**

**id:** integer

**config\_key:** string

**config\_value:** string

Following are the valid settings for the configuration table

**InstitutionName** *Name of the institution*

**InstitutionAddress** *Address Line of the institution*

**InstitutionPhoneNo** *Phone number of the institution*

**StudentAttendanceType** *Daily / SubjectWise*

**CurrencyType** *Symbol for currency*

**Locale** *en*

**AdmissionNumberAutoIncrement** *If set Increments the admission number automatically for each admission 1 /0*

**EmployeeNumberAutoIncrement** *If set Increments the admission number automatically for each admission 1/0*

**TotalSmsCount** *Read only The total number of smses sent from the application is shown here*

**NetworkState** *online/offline*

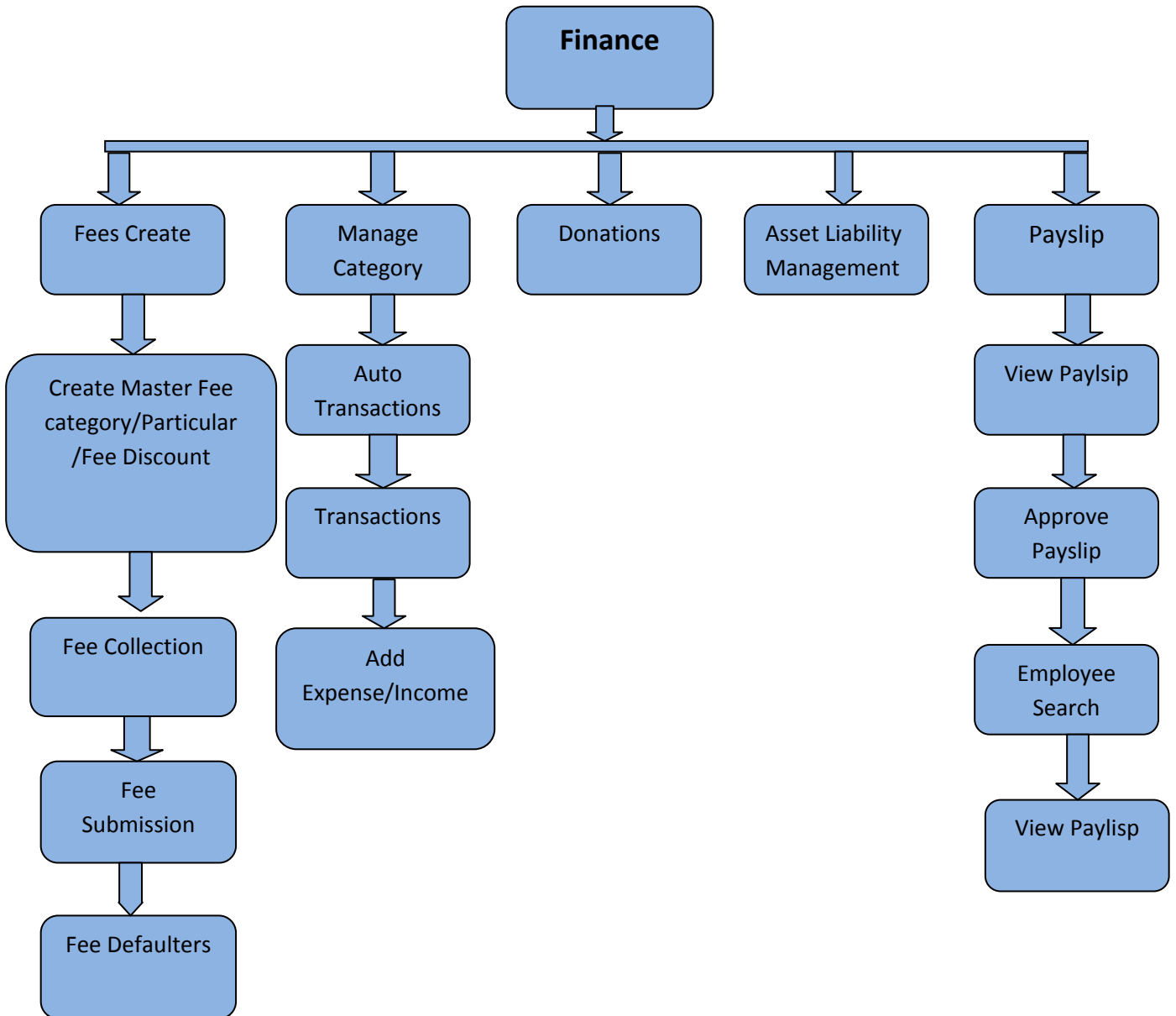
**IncludeGradingSystem** *CCE/GPA/CWA*

**EnableNewsCommentModeration**

AvailableModules" "HR" This row would enable the HR module

AvailableModules" "Finance" This row would enable the Finance module

## Finance Management



## AssociatedModels

**FinanceFee**

*FinanceFeeCategory*

**FinanceFeeCollection**

**FinanceFeeParticular**

**FinanceFeeStructureElement**

**FinanceTransaction**

*FinanceTransactionCategory*

*FinanceTransactionTrigger*

**Liability**

*SudentCatgeoryFeeCollectionDiscount*

*StudentCategoryDiscount*

**StudentFeeCollectionDiscount**  
**StudentFeeDiscount**  
**FinanceDonation**  
**FeeDiscount**  
**FeeCollectionParticular**  
**FeeCollectionDiscount**  
**BatchFeeDiscount**  
**BatchFeeCollectionDiscount**  
**Asset**

## **Database Structure**

### **FinanceFee**

**id**(Integer)  
**fee\_collection\_id**(Integer)  
**transaction\_id**(Integer)  
**student\_id**(Integer)  
**is\_paid**(Boolean)

### **FinanceFeeCategory**

**id**(Integer)  
**name**(String)  
**description**(Text)  
**batch\_id**(Integer)  
**is\_deleted**(Boolean)  
**is\_master**(Boolean)  
**created\_at**(Datetime)  
**updated\_at**(Datetime)

### **FinanceFeeCollection**

**id**(Integer)  
**name**(String)  
**start\_date**(Date)  
**end\_date**(Date)  
**fee\_category\_id**(Integer)  
**batch\_id**(Integer)  
**is\_deleted**(Boolean)

### **FinanceFeeParticular**

**id**(Integer)  
**name**(String)  
**description**(Text)  
**amount**(Decimal)  
**finance\_fee\_category\_id**(Integer)  
**student\_category\_id**(Integer)  
**admission\_no**(String)  
**student\_id**(Integer)  
**is\_deleted**(Boolean)  
**created\_at**(Datetime)

updated\_at(Datetime)

### **FinanceFeeStructureElement**

id(Integer)  
amount(Decimal)  
label(String)  
batch\_id(Integer)  
student\_category\_id(Integer)  
student\_id(Integer)  
parent\_id(Integer)  
fee\_collection\_id(Integer)  
deleted(Boolean)

### **FinanceTransaction**

id(Integer)  
title(String)  
description(String)  
amount(Decimal)  
fine\_included(Boolean)  
category\_id(Integer)  
student\_id(Integer)  
finance\_fees\_id(Integer)  
created\_at(Datetime)  
updated\_at(Datetime)  
transaction\_date(Date)  
fine\_amount(Decimal)  
master\_transaction\_id(Integer)  
finance\_id(Integer)  
finance\_type(String)  
payee\_id(Integer)  
payee\_type(String)  
receipt\_no(String)  
voucher\_no(String)

### **FinanceTransactionCategory**

id(Integer)  
name(String)  
description(String)  
is\_income(Boolean)  
deleted(Boolean)

### **FinanceTransactionTriggers**

id(Integer)  
finance\_category\_id(Integer)  
percentage(Decimal)  
title(String)  
description(String)

### **FinanceDonation**

**id**(Integer)  
**donor**(String)  
**description**(String)  
**amount**(Decimal)  
**transaction\_id**(Integer)  
**created\_at**(Datetime)  
**updated\_at**(Datetime)  
**transaction\_date**(Date)

### **FeeDiscount**

**id**(Integer)  
**type**(String)  
**name**(String)  
**receiver\_id**(Integer)  
**finance\_fee\_category\_id**(Integer)  
**discount**(Decimal)  
**is\_amount**(Boolean)

### **FeeCollectionParticular**

**id**(Integer)  
**name**(String)  
**description**(Text)  
**amount**(Decimal)  
**finance\_fee\_collection\_id**(Integer)  
**student\_category\_id**(Integer)  
**admission\_no**(String)  
**student\_id**(Integer)  
**is\_deleted**(Boolean)  
**created\_at**(Datetime)  
**updated\_at**(Datetime)

### **FeeCollectionDiscount**

**id**(Integer)  
**type**(String)  
**name**(String)  
**receiver\_id**(Integer)  
**finance\_fee\_collection\_id**(Integer)  
**discount**(Decimal)  
**is\_amount**(Boolean)  
**created\_at**(Datetime)  
**updated\_at**(Datetime)

### **Asset**

**id**(Integer)  
**title**(String)  
**description**(String)  
**amount**(Integer)  
**is\_inactive**(Boolean)  
**is\_deleted**(Boolean)

**created\_at**(Datetime)  
**updated\_at**(Datetime)

### **Liability**

**id**(Integer)  
**title**(String)  
**description**(String)  
**amount**(Integer)  
**is\_solved**(Boolean)  
**is\_deleted**(Boolean)  
**created\_at**(Datetime)  
**updated\_at**(Datetime)

### **Finance Controller**

**index** Here we can see the finance index  
**automatic\_transactions** To find the automatic transaction  
**donation** To create the donation  
**donation\_receipt** To take printout of donantion receipt  
**donation\_edit** To edit the donation  
**donation\_delete** To delete the donation  
**donation\_receipt\_pdf** To take printout of donation receipt  
**expense\_create** To create the expense  
**expense\_edit** To edit the expense  
**expense\_list\_update** To update the expense list  
**expense\_list\_pdf** To Take the expense list pdf  
**income\_create** To create the income  
**income\_edit** To edit the income  
**delete\_transaction** To delete the transaction  
**income\_list\_update** To update the income list  
**income\_details** To find the income details  
**income\_list\_pdf** To take the income list pdf  
**income\_details\_pdf** To take income details pdf  
**categories** To find the categories  
**category\_new** To create the category  
**category\_create** To create the category  
**category\_delete** To delete the category  
**category\_edit** To edit the category  
**category\_update** To update the category  
**transaction\_trigger\_create** To create the transaction trigger  
**transaction\_trigger\_edit** To edit the transaction trigger  
**transaction\_trigger\_update** To update the transaction trigger  
**transaction\_trigger\_delete** To delete the transaction trigger  
**update\_monthly\_report** To View the transaction report  
**transaction\_pdf** To take the transaction pdf  
**salary\_department** To find the employee department  
**salary\_employee** To find the salary of the employees  
**employee\_payslip\_monthly\_report** To get the employee payslip  
**donation\_report** To find donation report  
**fees\_report** To find the fees of students  
**batch\_fees\_report** To find the fees of batch  
**student\_fees\_structure** To find the student fees structure  
**approve\_monthly\_payslip** To approve the monthly payslip

**one\_click\_approve** *To approve the payslip by a single click*  
**one\_click\_approve\_submit** *For submitting the payslip*  
**employee\_payslip\_approve** *To approve employee payslip*  
**employee\_payslip\_reject** *To reject the employee payslip*  
**employee\_payslip\_accept\_form** *The employee payslip accept form action*  
**employee\_payslip\_reject\_form** *The employee payslip reject form action*  
**view\_monthly\_payslip** *To view the monthly payslip*  
**view\_employee\_payslip** *To view employee payslip*  
**search\_ajax** *To search the employees*  
**create\_liability** *To create the liability*  
**update\_liability** *To update the liability*  
**view\_liability** *To view the liability*  
**liability\_pdf** *To take the pdf of the liability*  
**delete\_liability** *To delete the liability*  
**each\_liability\_view** *To view the liabilities*  
**create\_asset** *To create the asset*  
**view\_asset** *To view the asset*  
**asset\_pdf** *To take the asset pdf*  
**edit\_asset** *To edit asset*  
**update\_asset** *To update the asset*  
**delete\_asset** *To delete the asset*  
**each\_asset\_view** *To view all assets*  
**master\_fees** *To create the master fees*  
**master\_category\_new** *To create the master category*  
**master\_category\_create** *To create the master category*  
**master\_category\_edit** *To edit the master category*  
**master\_category\_update** *To update the master category*  
**master\_category\_particulars** *To find the master finance particulars*  
**master\_category\_particulars\_update** *To update the master category particulars*  
**master\_category\_particulars\_delete** *To delete the master category particular*  
**master\_category\_delete** *To delete the master category*  
**show\_master\_categories\_list** *To view the master category list*  
**fees\_particulars\_new** *To create the fees particular*  
**fees\_particular\_create** *To create the fees particular create*  
**fees\_particulars\_new2** *To create the fees particular*  
**fees\_particular\_create2** *To create the fees particular create*  
**add\_particulars** *To add particulars*  
**add\_particulars\_new** *To create the new particulars*  
**add\_particulars\_create** *To create the particulars*  
**student\_or\_student\_category** *To find the student category*  
**add\_particulars\_edit** *To edit the particulars*  
**add\_particulars\_update** *To update the particulars*  
**add\_particulars\_delete** *To delete the particulars*  
**fee\_collection\_batch\_update** *To update the fee collection*  
**fee\_collection\_new** *To create the fee collection*  
**fee\_collection\_create** *To create the fee collection*  
**fee\_collection\_view** *To view the fee collection*  
**fee\_collection\_dates\_batch** *To find the finance fee collections*  
**fee\_collection\_edit** *To edit the fee collection*  
**fee\_collection\_update** *To update the fee collection update*  
**fee\_collection\_delete** *To delete the fee collection*  
**fee\_submission\_batch** *To find the active batches*  
**update\_fee\_collection\_dates** *To find the fee collection dates*  
**load\_fee\_submission\_batch** *To load the fee submissions*  
**update\_ajax** *To find the fees*



**student fee receipt pdf** To take the student receipt pdf  
**update fine ajax** To update the fine  
**search logic** To search the fees  
**fees student dates** To find student fee collections  
**fees submission student** To find the student fees submission  
**update student fine ajax** To update the student fees  
**fees submission save** To save the fees  
**fees student structure search logic** Student fee structure search  
**fees structure dates** To find fees structure  
**fees structure for student** To find the student fee structure  
**fee defaulters** To find the fee defaulters  
**update batches** To find the batches  
**update fee collection dates defaulters** To update fee collection defaulters  
**fees defaulters students** To find student fee defaulters  
**fee defaulters pdf** To take the fee defaulters pdf  
**pay fee defaulters** To pay fees of defaulters  
**update defaulters fine ajax** To update the defaulters fees  
**report compare** To compare the reports  
**month date** To find the dates  
**partial payment** To pay partial fees  
**pdf fees structure** The fee structure pdf  
**graph for update monthly report** To get the graph of monthly report  
**graphp for transaction comparison** To take the graph of the report compare  
**fee discounts** To find the fee discounts  
**fee discount new** To create the fee discount  
**load discount create form** Discount create form  
**load discount batch** Batchwise discount  
**load batch fee category** To find batch wise fee category  
**batch wise discount create** To create batch wise discount  
**category wise fee discount create** To create the category wise discount  
**student wise fee discount create** Create student wise discount  
**update master fee category list** To update master fee category  
**show fee discounts** To view the fee discounts  
**edit fee discount** To edit the fee discount  
**update fee discount** To update fee discount  
**delete fee discount** To delete the fee discount  
**collection details view** To view the fee collections  
**fixed category name** To find the finance transaction categories

## **SMS MANAGEMENT**

### **AssociatedModels**

**SmsLog**  
**SmsSetting**  
**SmsMessage**

### **DataBase Structure**

**SmsLog**

*id(Integer)*  
*mobile(String)*  
*gateway\_response(String)*  
*sms\_message\_id(Integer)*  
*created\_at(Datetime)*  
*updated\_at(Datetime)*

### **SmsSetting**

*id(Integer)*  
*settings\_key(String)*  
*is\_enabled(Boolean)*

### **SmsMessage**

*id(Integer)*  
*body(String)*  
*created\_at(Datetime)*  
*updated\_at(Datetime)*

## **Method And Actions**

### **SmsController**

**index** *To set the new smssetting*  
**settings** *Find the smssettings using settings key*  
**update general sms settings** *To update the general sms settings*  
**students** *To send sms to students*  
**list students** *To list the students*  
**batches** *To find the active batches*  
**sms all** *To send sms to all*  
**employees** *To send sms to employees*  
**list employees** *To list the employees*  
**departments** *To send sms to employee departments*  
**show sms messages** *To show the messages*  
**show sms logs** *To show the sms logs*

### **SmsSettingsController**

**index** *To find the enabled application*  
**update general sms settings** *To update the general sms settings*

## **Attendance Management**

### **Associated Models**

*Attendance*  
*Student*  
*Batch*

### **DateBase Structure**

## **Attendance**

**id**(Integer)  
**student\_id**(Integer)  
**period\_table\_entry\_id**(Integer)  
**forenoon**(Boolean)  
**afternoon**(Boolean)  
**reason**(String)  
**month\_date**(Date)  
**batch\_id**(Integer)

## **Method And Actions**

### **attendances controller**

**index** To show the attendance of the students  
**list subject** To list the subjects  
**show** To show the attendance  
**subject wise register** Find the attendance register by subject wise  
**daily register** Find the attendance by daily wise  
**new** To create the new attendance  
**create** To create the new attendance  
**edit** To edit the attendance  
**update** To update the attendance  
**destroy** To delete the attendance  
**only privileged employee allowed** Give permission to privileged employees

### **AttendanceReportsController**

**index** To find the attendance of the students  
**subject** To find the subjects  
**mode** To select the mode  
**show** To show the attendance by selecting mode  
**year** To get the attendance report year wise  
**report2** To get the attendance report  
**report** To take the attendance report  
**student details** To get the student details  
**filter** To filter the attendance  
**fliter2** To filter the attendance  
**advance search** To search the attendance  
**report pdf** To take the pdf of attendance report pdf  
**filter report pdf** To filter the pdf of attendance report

## **Fedena Plugins**

The fedena plugins are different types based on the functionality. These plugin modules are using in the fedena core .

Example: Library, Hostel, Transport etc.