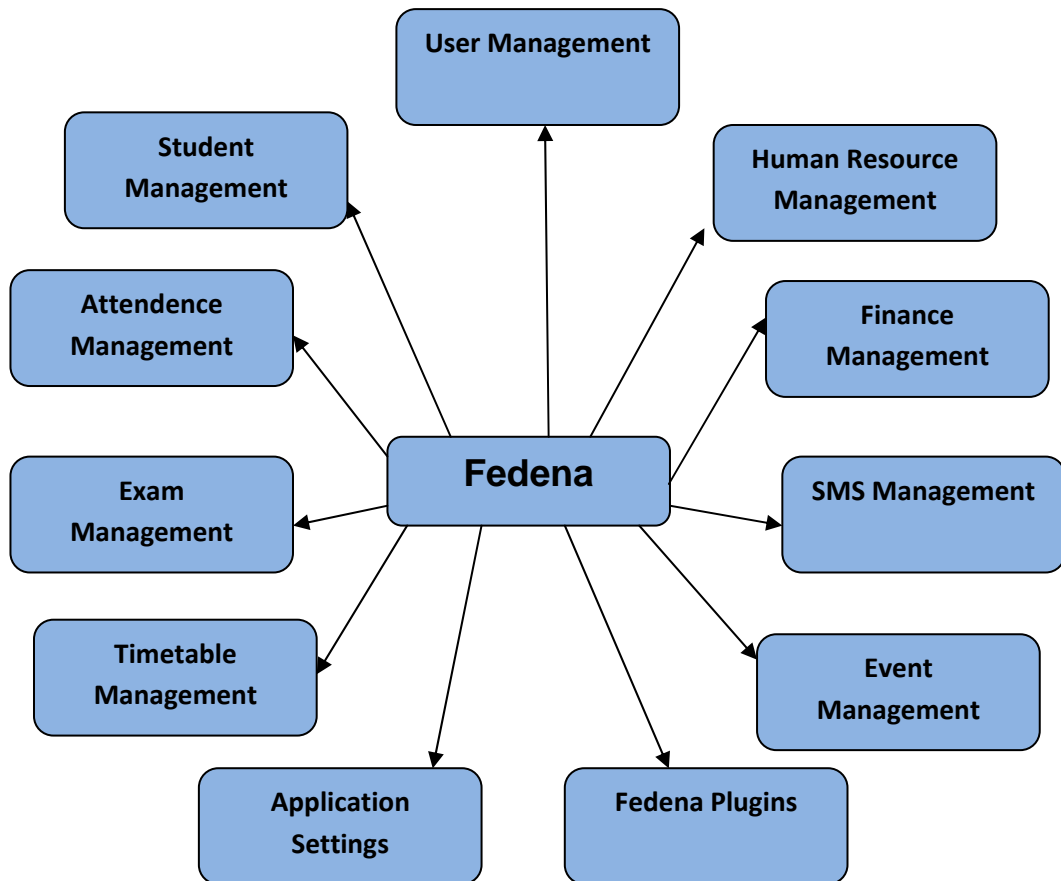


Project Fedena Technical Documentation



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

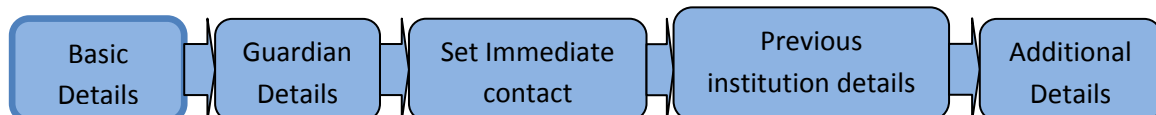
change_password Action for initiating the password change of user
choose_layout Method for choosing action to login page / forgot password page / dashboard according to user request
create Method to create User Object
current_user Method to return the User Object of current logged in user
dashboard Shows 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.

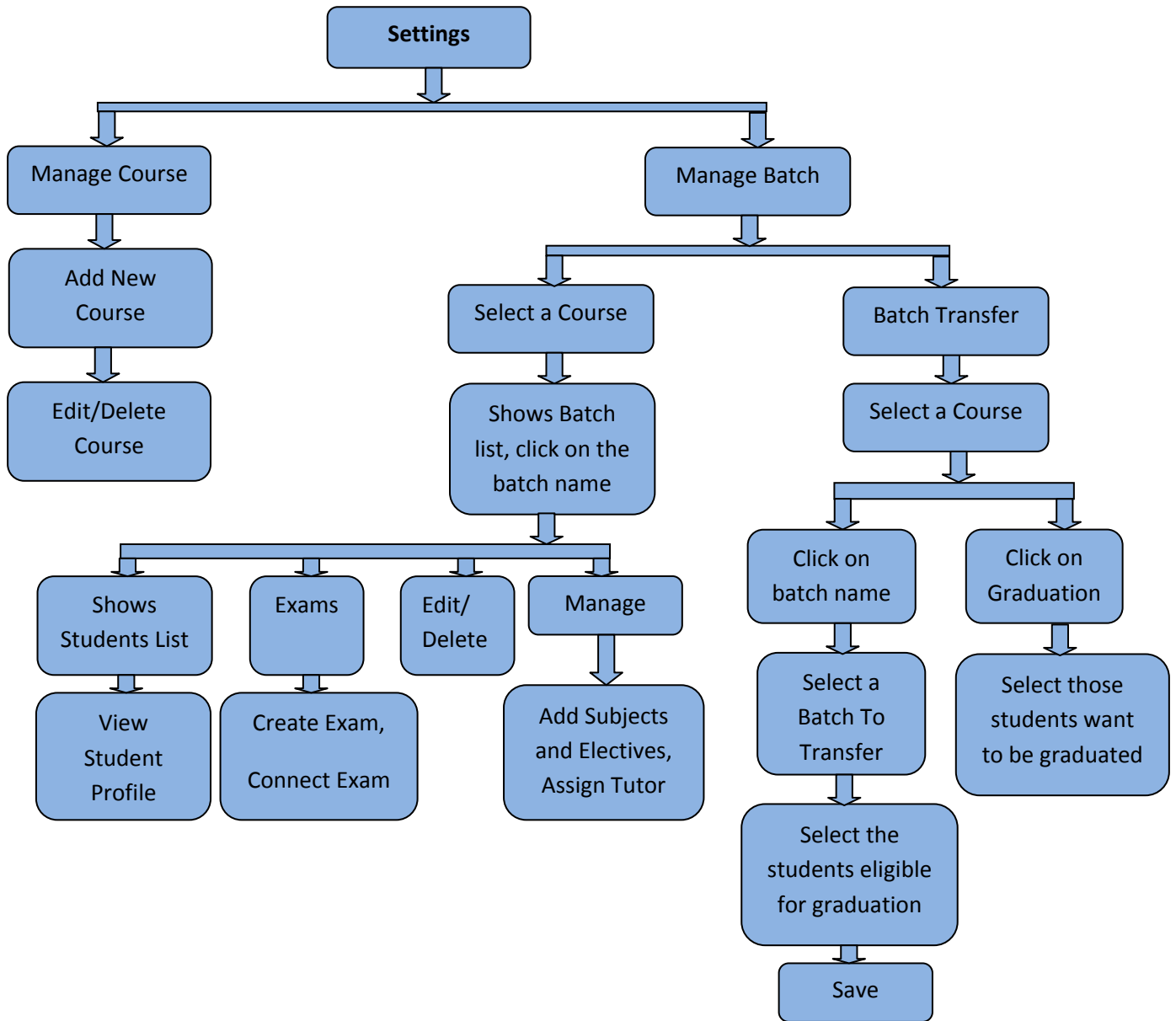
edit additional details : Edit the additional details entered for a student.

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

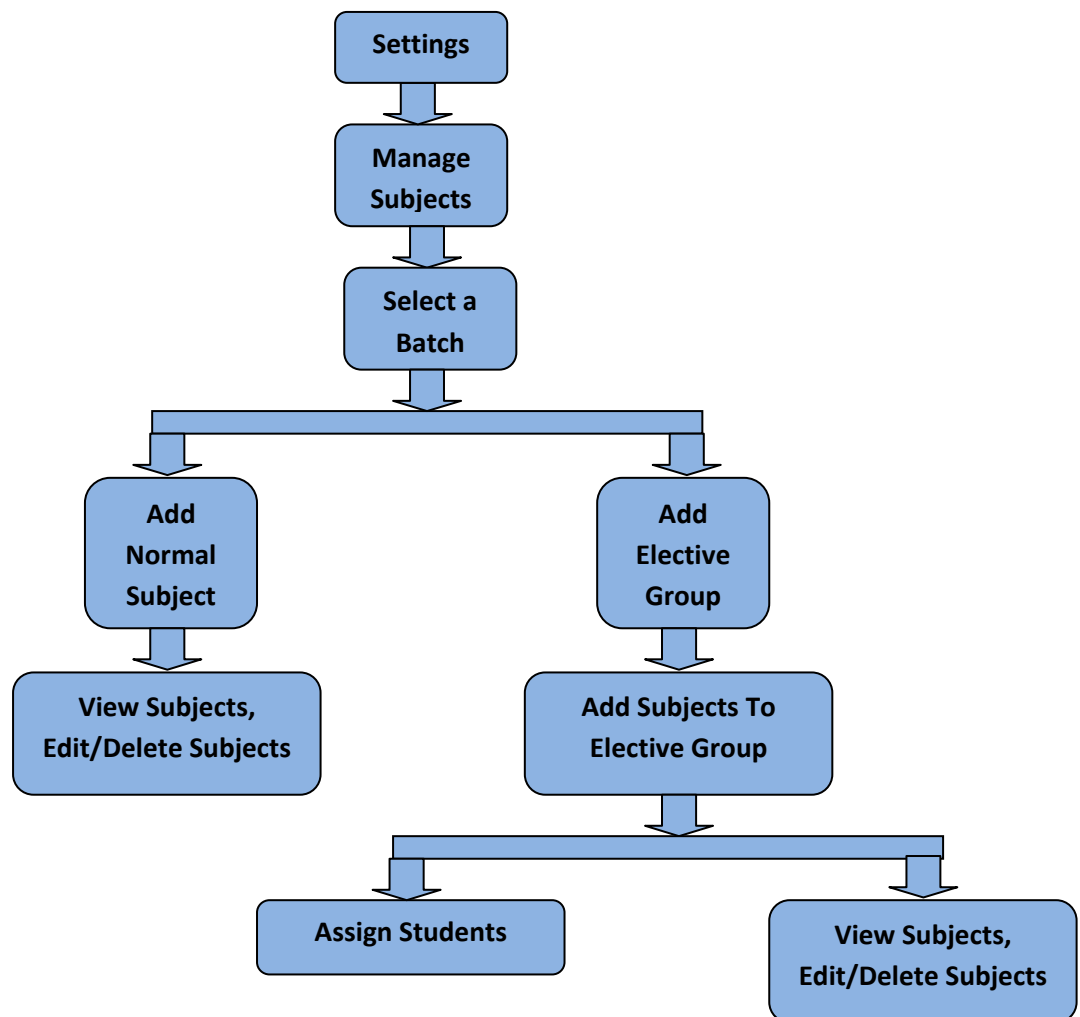
index *To find the batch*
new *To create the new batch*
create *To create the new batch*
update *To update the batch*
show *To show the batch*
destroy *To destroy the batch*
assign_tutor *To assign the tutor*
update_employees *To update the employees*

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*

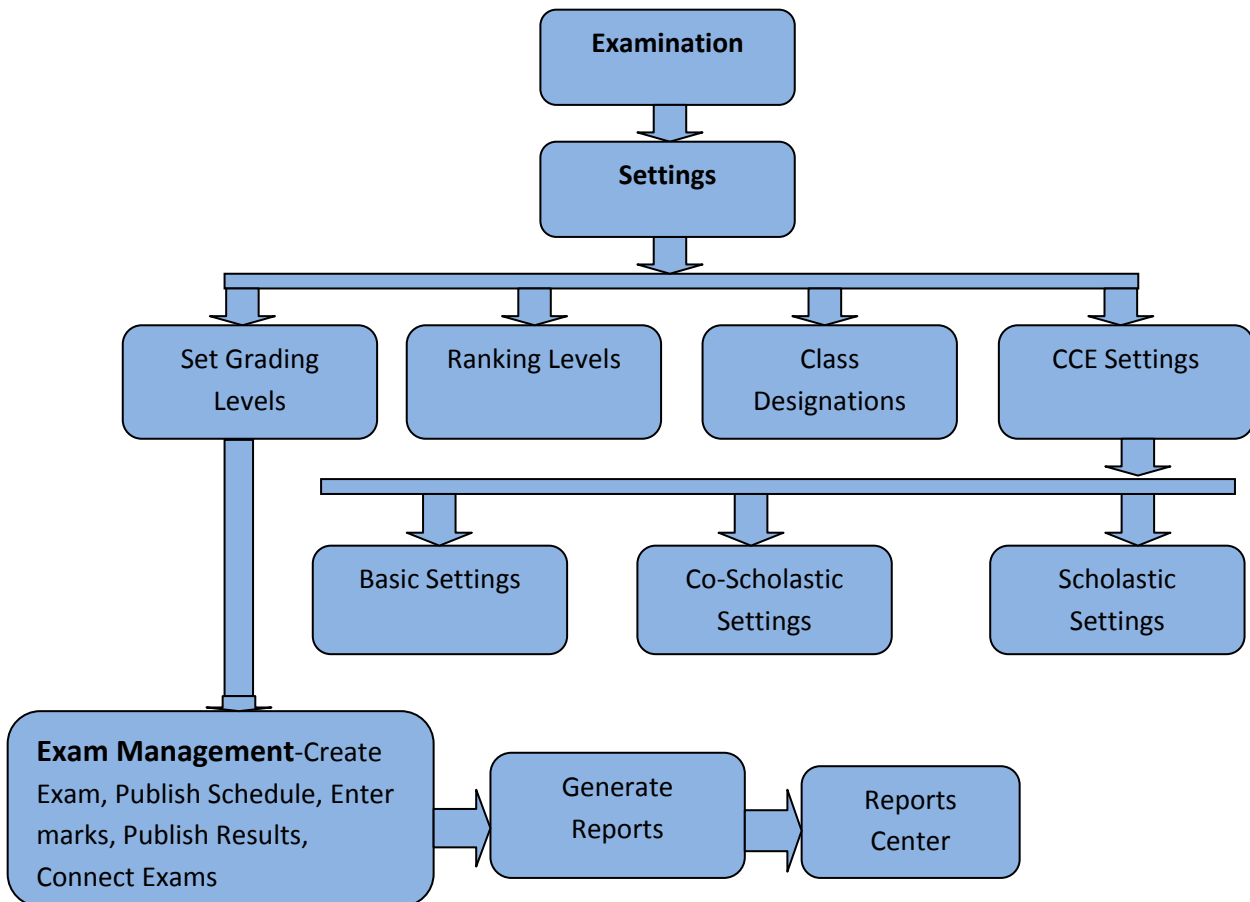
create To create the new elective group
edit To edit the elective group
update To update the elective group
destroy To delete the elective group
pre_load_objects It is a private method

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*

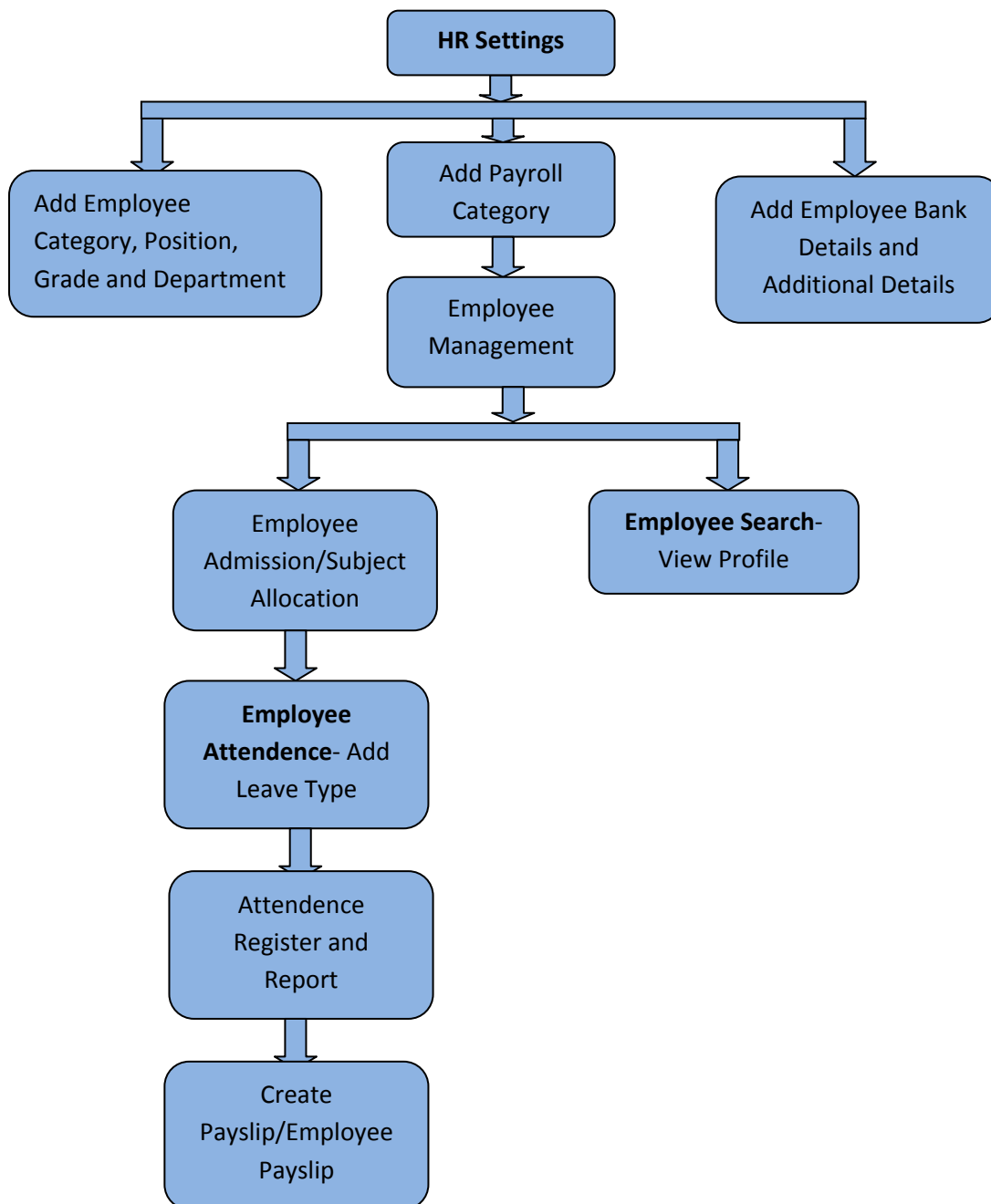
ObservationController

show *To show the observations*

RankingLevelsController

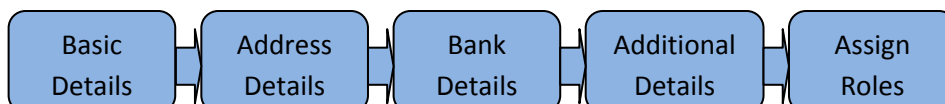
load_ranking_levels *To load ranking level form*
create_ranking_level *To create the ranking level*
edit_ranking_level *To edit the ranking level*
update_ranking_level *To update the ranking level*
delete_ranking_level *To delete the ranking level*
ranking_level_cancel *To cancel the ranking levels*
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

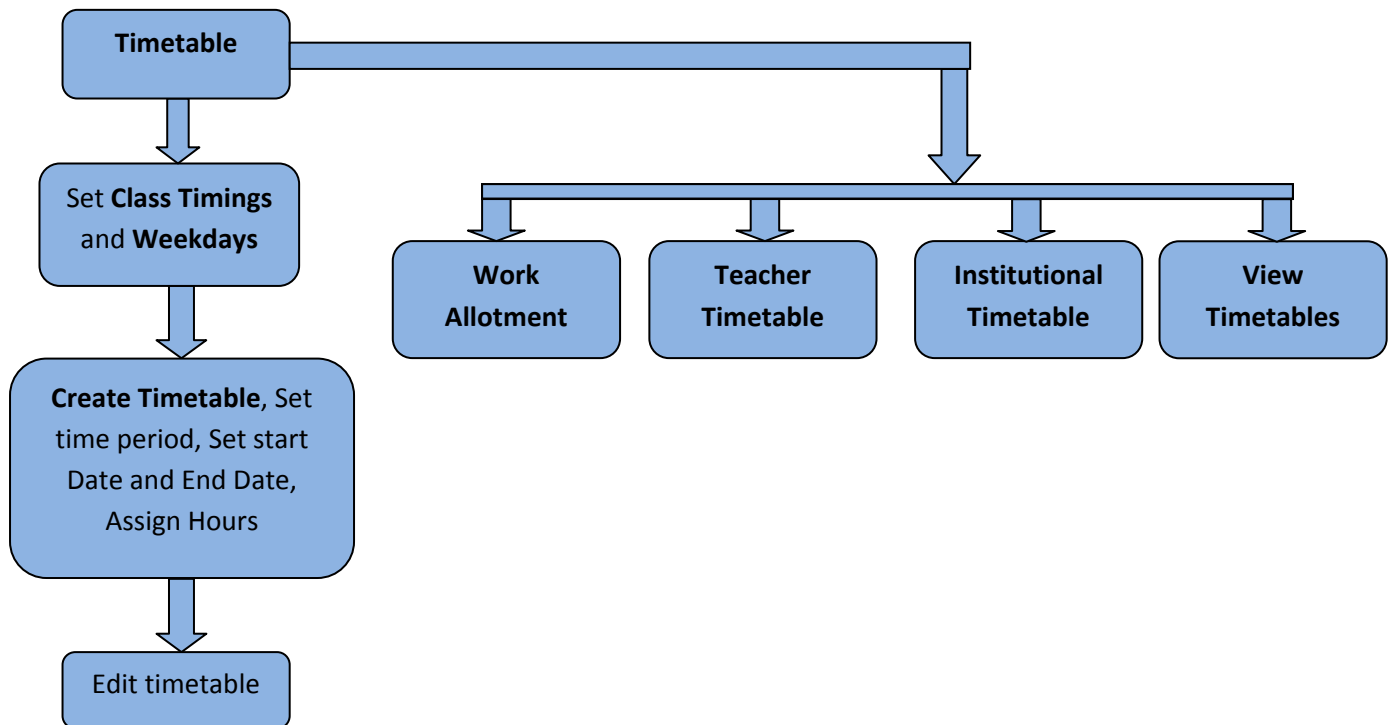
add_leave_types *To add employee leave types*
edit_leave_types *To edit the employee leave types*
delete_leave_types *To delete the employee leave types*
leave_reset_settings *To reset the leaves*
update_employee_leave_reset_all *To reset all employee leaves*
employee_leave_reset_by_department *To reset 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*

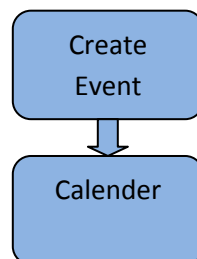
employee_timetable *To find the employee timetable*
update_employee_tt *To update the employee timetable*
students_view *Timetable view of students*
update_students_tt *To update student timetable*
weekdays *To find the weekdays*
timetable_pdf *To take the timetable pdf*
work_allotment *To allot the 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*

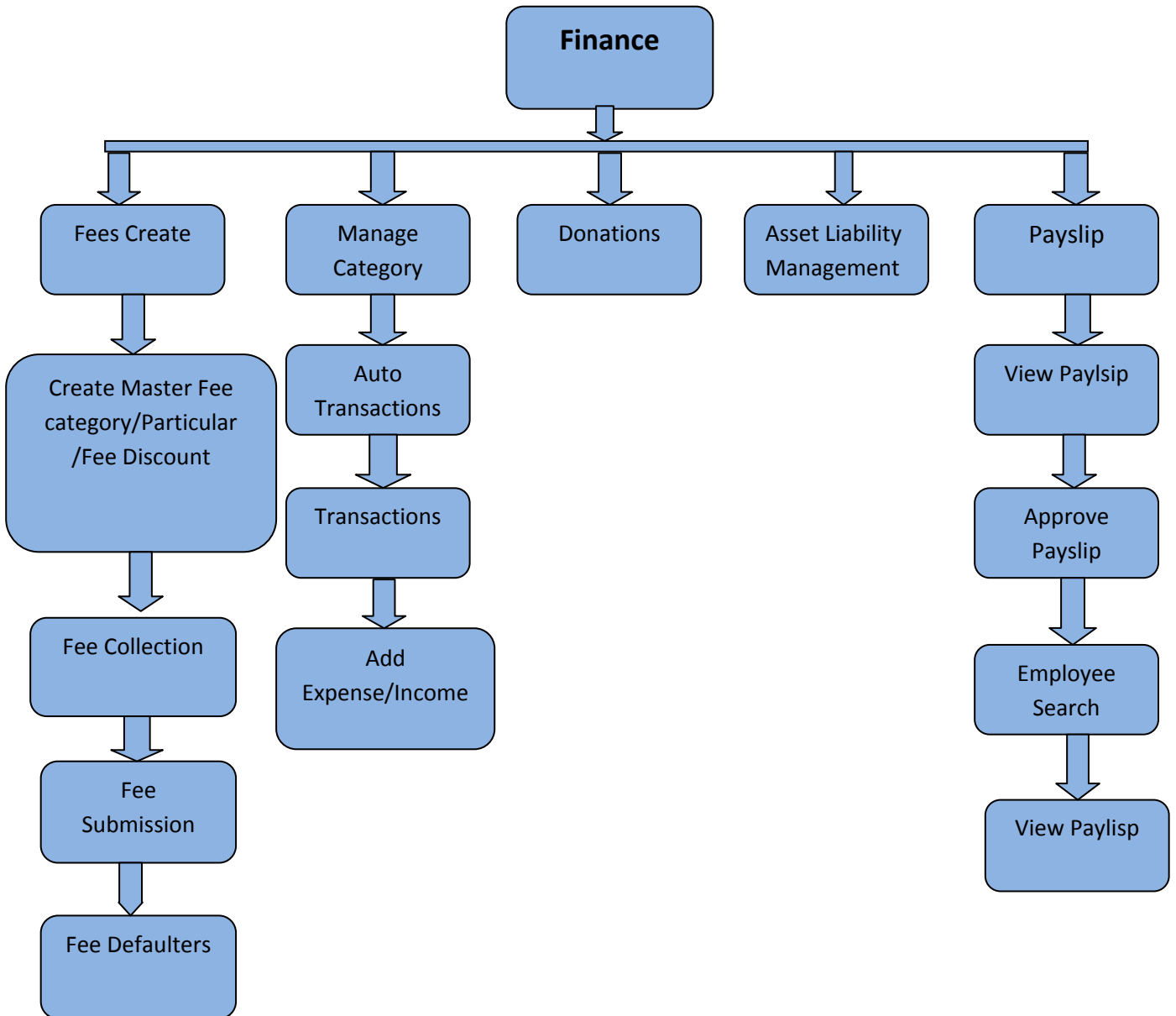
IncludeGradingSystem *CCE/GPA/CWA*

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

StudentCategoryFeeCollectionDiscount

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.