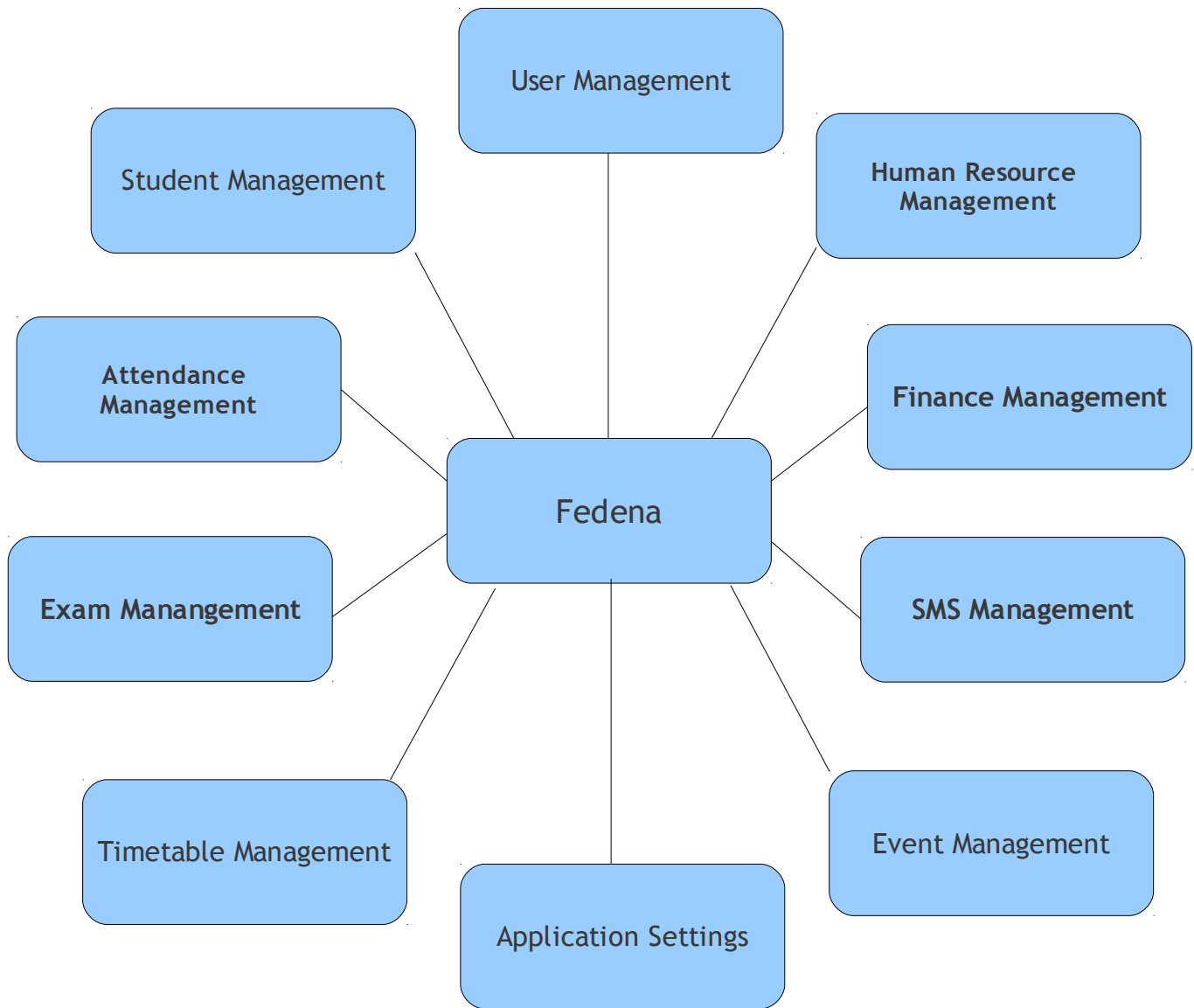


Project Fedena Technical Documentation



BASIC SYSTEM ARCHITECTURE



User Module

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

Steps in students admission



Models and Tables

Associated models

User

Student

StudentSubject

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)

student(Boolean) Set for administrators

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)

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_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` initiate change password action for a user

Student management

Manages students , guardians and former students.

Models and Tables

Associated models

Student

Guardian

ArchivedStudent

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)

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)

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)

Controllers

StudentController

Actions / Methods

academic_pdf Generate the PDF report for examination records of a batch for a student
academic_report Academic report of the student
academic_report_all Academic report for all the students
add_additional_details Add Additional details for a student.
add_guardian Add a *Guardian* student
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
admission4 Form to add Additional Details for the student
advanced_search Student advanced search
advanced_search_pdf PDF for the advanced search results
assign_all_students Assign all students to a particular elective
assign_students Assign students one by one to an elective.
Categories Add/View student categories

category_delete Delete a student category
category_edit Edit a student category
category_update Update a student category
change_to_former Archive a *Student* while graduation / releasing from the institution
del_guardian Delete a *Guardian* of a student
delete_additional_details Delete additional details of a student
destroy Destroy a student record
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
edit Edit a *Student*
edit_additional_details Edit the additional details entered for a student.
edit_admission4 Edit the previous details
edit_guardian Edit the guardian details
electives
email
exam_report Generate report of an examination
generate_all_tc_pdf Generate TC for all the students of the batch.
generate_tc_pdf Generate the TC for a student
guardians View Guardians of a students
list_batches List all batches of a course
list_doa_year
list_dob_year
list_students_by_course
previous_data Fetch details of previous institution .
previous_subject Fetch details of subjects from previous institution
previous_years_marks_overview
profile Fetch all details to display the previous details
profile_pdf Generate the PDF report of the student profile
save_previous_subject Save subject details of previous details
search_ajax Ajax search for students

show Method for fetching image data from database
show_previous_details Show previous institution details
student_annual_overview Daw open flash chart for the entire year
subject_wise_report Daw open flash chart for the entire year
unassign_all_students
unassign_students
update_student_result_for_examtype Update exam result for a particular exam.
view_all Fetch all active batches

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.

Additional examinations are similar to Examination except that each additional exam group has a field *students_list* showing the id's of students writing the exam.

Models and Databases

Examinations

- Exam
- Exam Group
- Exam Score
- Gradinglevel

Additional Examinations

- Additional Exam
- Additional Exam Grou
- Additional Exam Score

Grouping Examinations

- Exam group

Database Structure

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

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)

GroupedExam

id(Integer)
exam_group_id(Integer)
batch_id(Integer)

AdditionalExam

id(Integer)
additional_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)

AdditionalExamGroup

id(Integer)
name(String)
batch_id(Integer)
exam_type(String)
is_published(Boolean)
result_published(Boolean)
students_list(String)
exam_date: date

AdditionalExamScore

id(Integer)
student_id(Integer)
additional_exam_id(Integer)
marks: decimal
grading_level_id(Integer)
remarks(String)
is_failed(Boolean)
created_at(Datetime)
updated_at(Datetime)

Action / Methods

ExamGroupsController

Create Create an exam group
destroy
edit
index
new
set_exam_group_name
set_exam_maximum_marks
set_exam_minimum_marks
show
update

ExamsController

save_scores Save score for an exam for a student

Create
destroy
edit
index
new

Publish Send sms alerts and create *events* for the particular examination

private method query_data Query details for the currently requested *Exam* Object

ExamScoresController

in_place_edit_for :exam_score, :score

Similar methods are in place for the AdditionalExams

Methods for exam scores

Class average marks(Exam)

Subject total (Student,Subject)

Exam Group total (ExamGroup Student)

Score_for(Student)

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

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)

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

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

ArchivedEmployeeBankDetailTable.

Methods and actions

New,Add,Edit,Delete,Methods for:

- AdditionalDetails
- BankDetails
- EmployeeGrade
- PayrollCategory
- Payslip
- Category
- position
- Department

assign_reporting_manager

change_reporting_manager

Ajax Search - Ajax Search for employees

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

TimetableEntry

id: integer

batch_id: integer

week_day_id: integer

class_timing_id: integer

subject_id: integer

employee_id: integer

ClassTiming

id: integer

batch_id: integer

name: string

start_time: time

end_time: time

s_break: boolean

Weekday

id: integer

batch_id: integer

weekday: string

Events Module

Event

Events are created for various notifications and calendar entries .Cr

A new *Event* Object is created while a new exam is created or a fee is due.

Database structre of event table

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: integer

News management

Models and databases

Associated models

News

News_comments

Database structure:

news

title(string)

author_id(integer)

content(text)

created_at(date_time)

updated_at(date_time)

news_comments

content(text)

news(integer)

author(integer)

created_at(date_time)

updated_at(date_time)

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*

"ExamResultType *Marks / Grades / MarksAndGrades*

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*

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

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