Software Architecture Design SAD :(

AT70.18

Asian Institute of Technology
School of Engineering and Technology

MVC-Architecture Web Application

Submitted to:
Dr. Chaklam Silpasuwanchai

Submitted by:
Abhishek Koirala (st120199)
Bibhuti regmi (st120320)
Shikshya Dahal (st120186)

Submitted Date: 4th February 2019

Role of each member:

All of us first discussed the concept of the project we were going to implement. We created the project and connected it with a repository in Shikshya's github account. Each of us imported the project and then worked on the same repository. Shikshya was responsible for generating and implementing MVC architecture and the rest of us were responsible for breaking down our concept into MVC logic and editing the view.

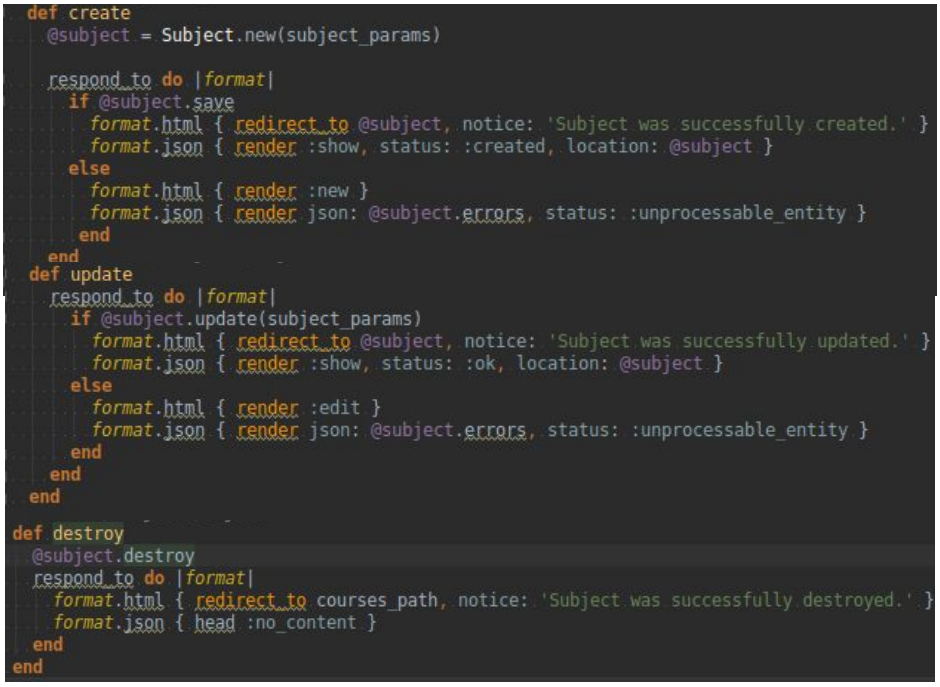
MVC

MVC (Model View Pattern) is an architectural pattern which is solely responsible for separation of concern and removing dependencies among different components of an entire project. While Model may consists of something like a POJO class which is responsible for communicating with database, the controller remains in between view and model and receives as well as process data to and from view and model and vice versa.

Our project:
This project is a web application which allows the user to add courses and inside each course, add subjects. To accomplish it we have implemented MVC pattern. There is an interface to add a course and also subjects within the course.

Major features are:

- CRUD operation for course and Subject

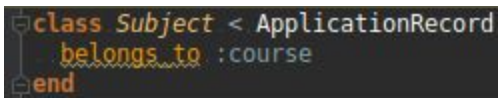


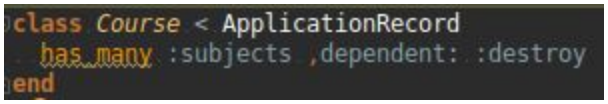
```
def create
  @subject = Subject.new(subject_params)

  respond_to do |format|
    if @subject.save
      format.html { redirect_to @subject, notice: 'Subject was successfully created.' }
      format.json { render :show, status: :created, location: @subject }
    else
      format.html { render :new }
      format.json { render json: @subject.errors, status: :unprocessable_entity }
    end
  end
end
def update
  respond_to do |format|
    if @subject.update(subject_params)
      format.html { redirect_to @subject, notice: 'Subject was successfully updated.' }
      format.json { render :show, status: :ok, location: @subject }
    else
      format.html { render :edit }
      format.json { render json: @subject.errors, status: :unprocessable_entity }
    end
  end
end

def destroy
  @subject.destroy
  respond_to do |format|
    format.html { redirect_to courses_path, notice: 'Subject was successfully destroyed.' }
    format.json { head :no_content }
  end
end
```

- One to Many relationship between course and subjects





What did we learn from this project?

We learnt the practical implementation of MVC design pattern from this project. The separated structure of model, view and controller in rails project helped us to separate our logic. We also learnt to work in team and utilize git to synchronize our code.