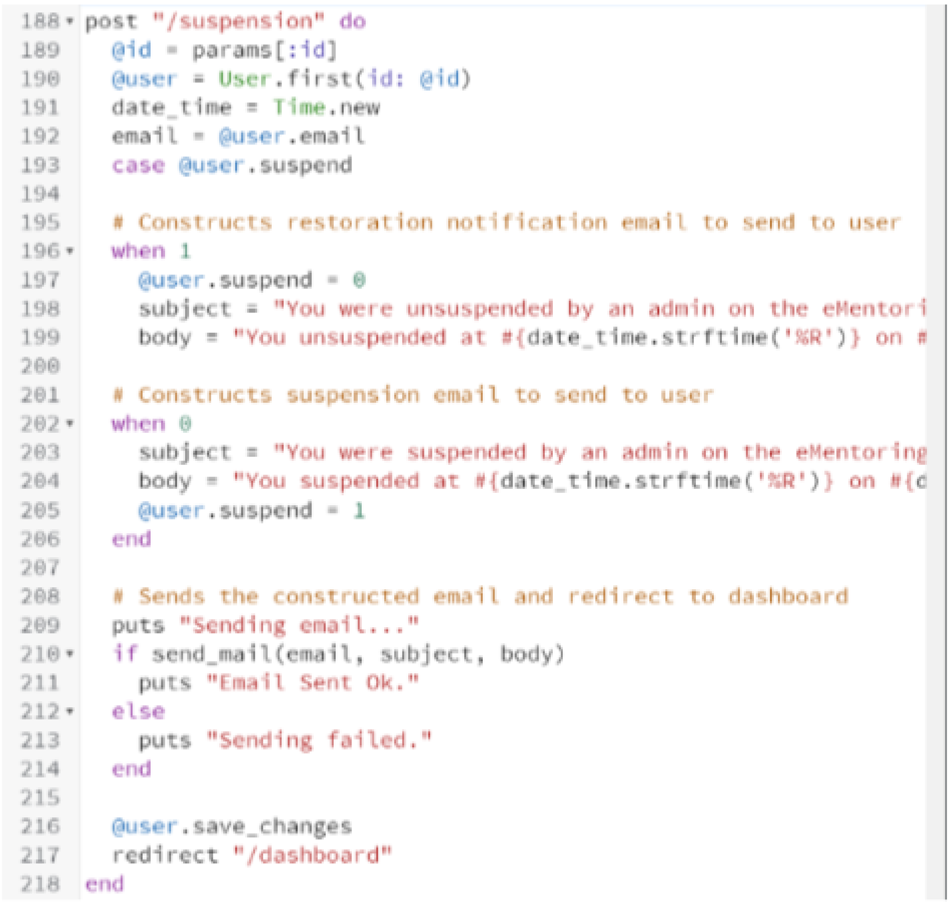
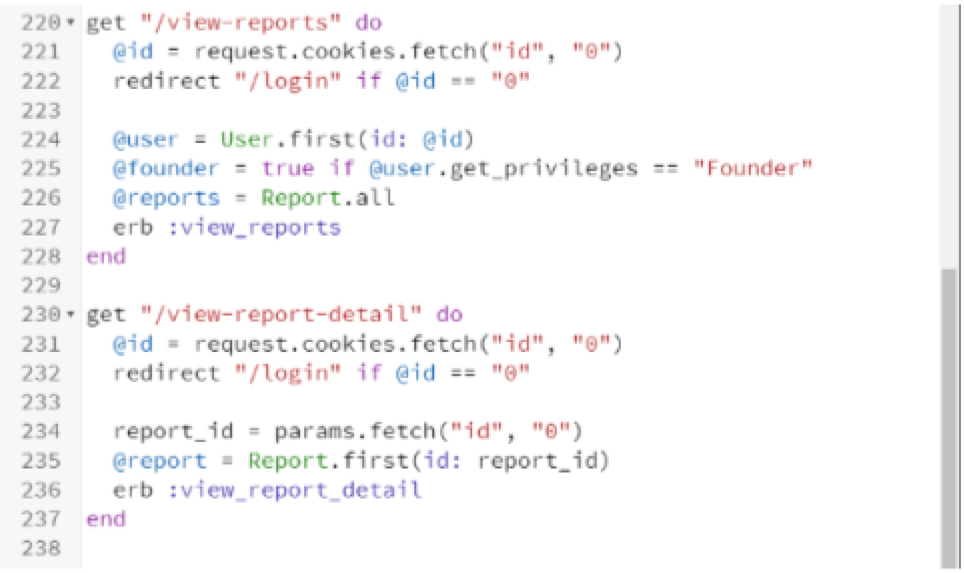
Exemplar Code



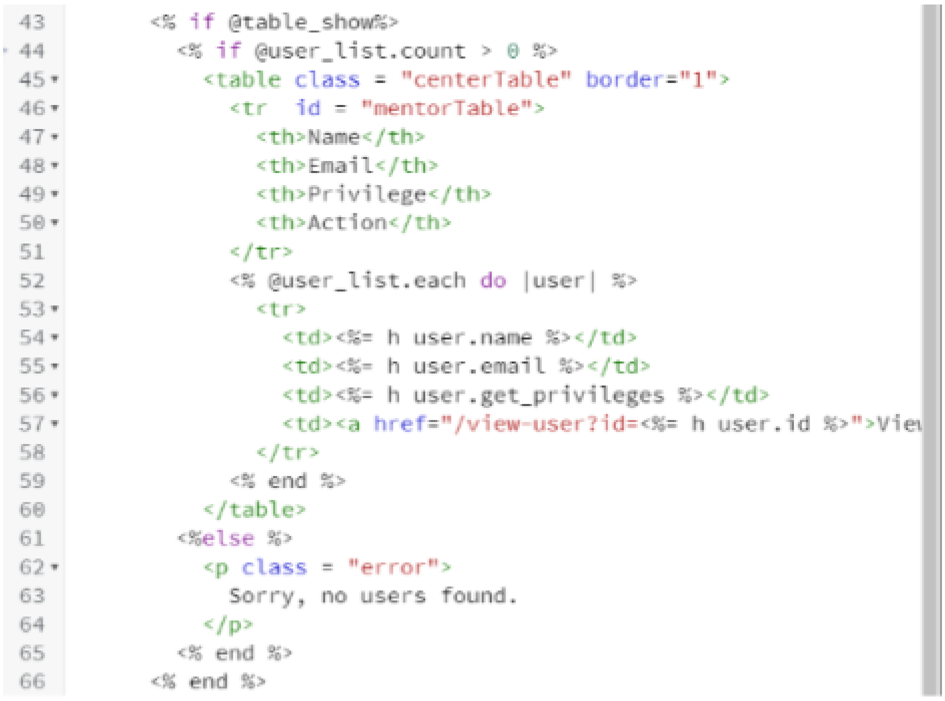
```ruby
post "/suspension" do
  @id = params[:id]
  @user = User.first(id: @id)
  date_time = Time.new
  email = @user.email
  case @user.suspend

  # Constructs restoration notification email to send to user
  when 1
    @user.suspend = 0
    subject = "You were unsuspended by an admin on the eMentori
    body = "You unsuspended at #{date_time.strftime('%R')} on #

  # Constructs suspension email to send to user
  when 0
    subject = "You were suspended by an admin on the eMentoring
    body = "You suspended at #{date_time.strftime('%R')} on #{d
    @user.suspend = 1
  end

  # Sends the constructed email and redirect to dashboard
  puts "Sending email..."
  if send_mail(email, subject, body)
    puts "Email Sent Ok."
  else
    puts "Sending failed."
  end

  @user.save_changes
  redirect "/dashboard"
end
```

Listing A - controllers/controllers_admin.rb (188 - 228)



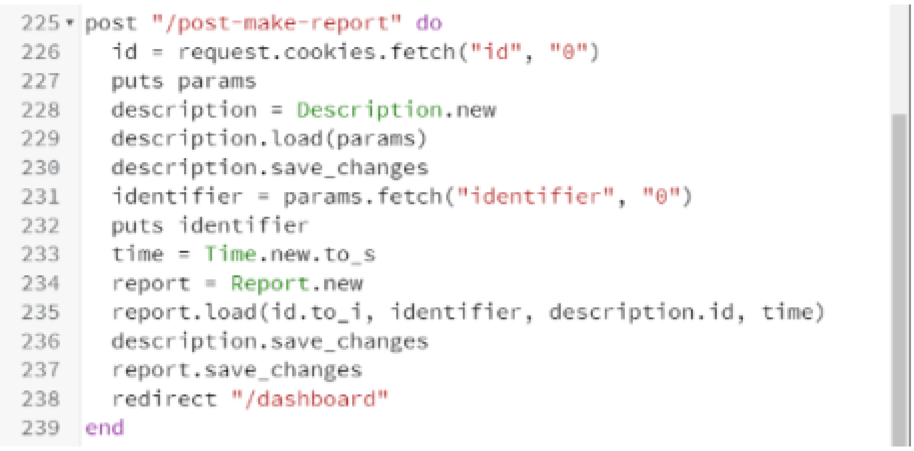
```ruby
get "/view-reports" do
  @id = request.cookies.fetch("id", "0")
  redirect "/login" if @id == "0"

  @user = User.first(id: @id)
  @founder = true if @user.get_privileges == "Founder"
  @reports = Report.all
  erb :view_reports
end

get "/view-report-detail" do
  @id = request.cookies.fetch("id", "0")
  redirect "/login" if @id == "0"

  report_id = params.fetch("id", "0")
  @report = Report.first(id: report_id)
  erb :view_report_detail
end
```

Listing B - controllers/controllers_admin.rb (220 - 237)



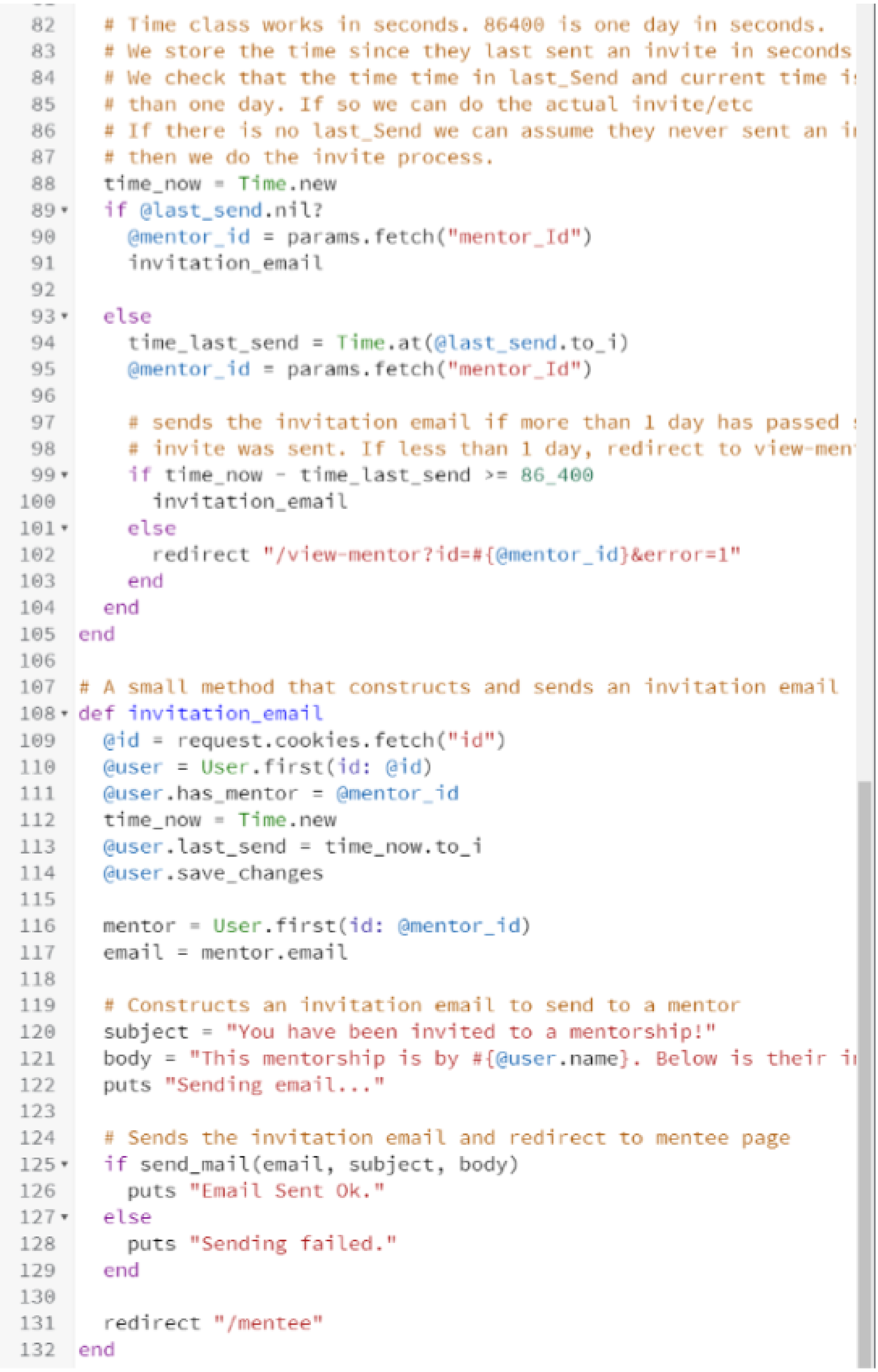
```
    <% if @table_show%>
      <% if @user_list.count > 0 %>
        <table class = "centerTable" border="1">
          <tr  id = "mentorTable">
            <th>Name</th>
            <th>Email</th>
            <th>Privilege</th>
            <th>Action</th>
          </tr>
          <% @user_list.each do |user| %>
            <tr>
              <td><%= h user.name %></td>
              <td><%= h user.email %></td>
              <td><%= h user.get_privileges %></td>
              <td><a href="/view-user?id=<%= h user.id %>">Vie
            </tr>
          <% end %>
        </table>
      <%else %>
        <p class = "error">
          Sorry, no users found.
        </p>
      <% end %>
    <% end %>
```

Listing C - views/admin.erb (43 - 66)



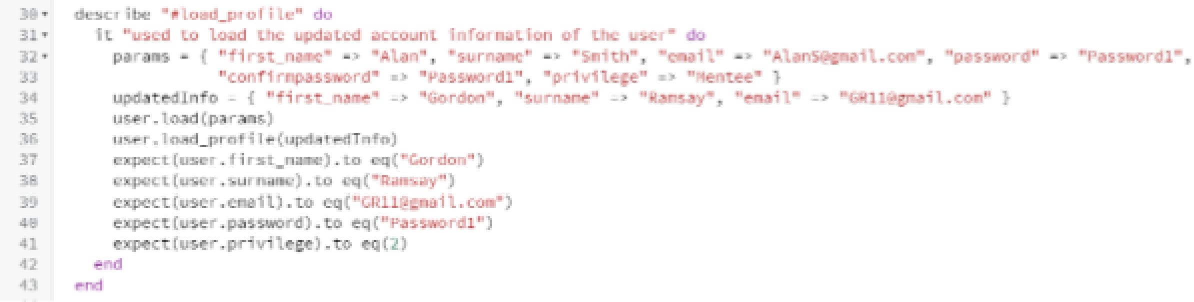
```
post "/post-make-report" do
  id = request.cookies.fetch("id", "0")
  puts params
  description = Description.new
  description.load(params)
  description.save_changes
  identifier = params.fetch("identifier", "0")
  puts identifier
  time = Time.new.to_s
  report = Report.new
  report.load(id.to_i, identifier, description.id, time)
  description.save_changes
  report.save_changes
  redirect "/dashboard"
end
```

Listing D - controllers/controllers.rb (225 - 239)



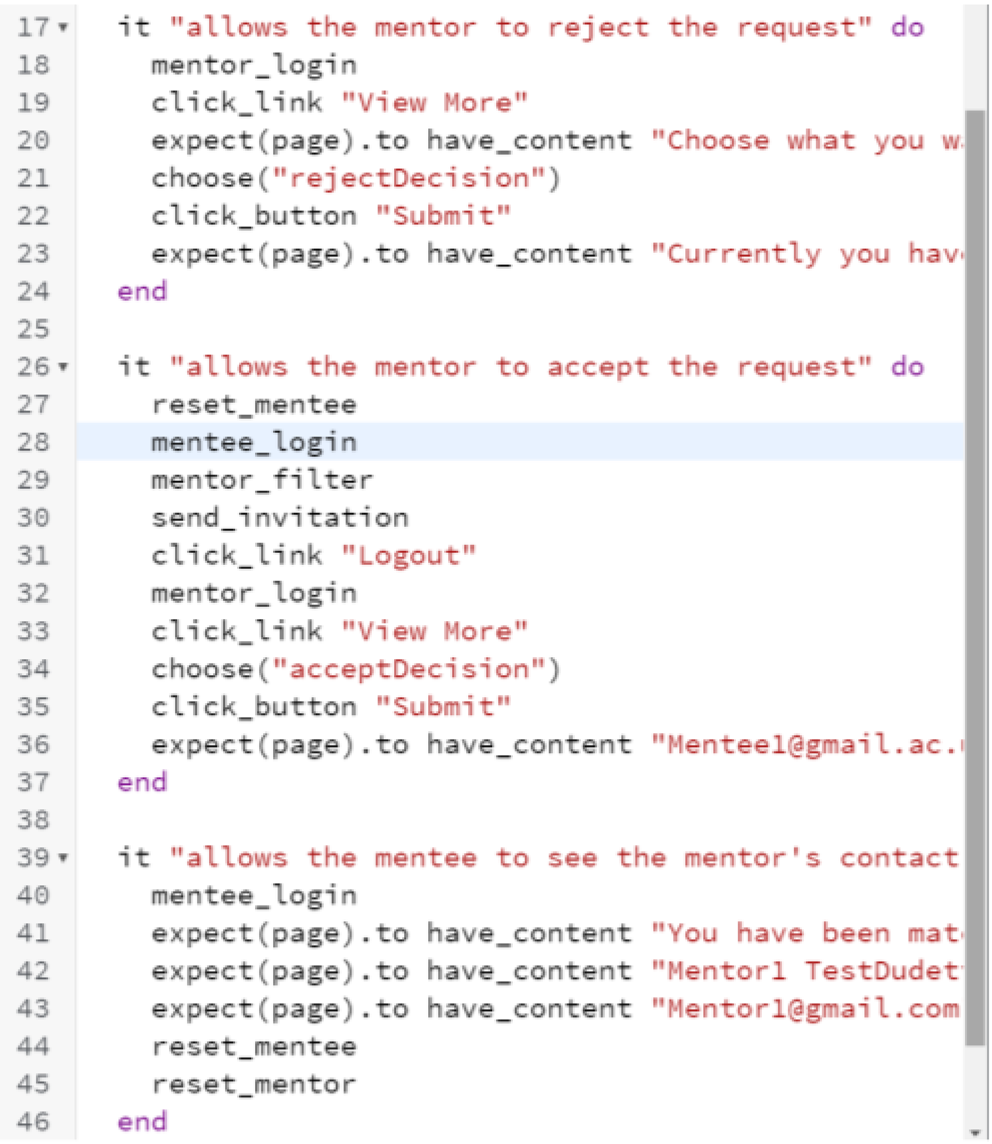
```
  # Time class works in seconds. 86400 is one day in seconds.
  # We store the time since they last sent an invite in seconds
  # We check that the time time in last_Send and current time i
  # than one day. If so we can do the actual invite/etc
  # If there is no last_Send we can assume they never sent an i
  # then we do the invite process.
  time_now = Time.new
  if @last_send.nil?
    @mentor_id = params.fetch("mentor_Id")
    invitation_email

  else
    time_last_send = Time.at(@last_send.to_i)
    @mentor_id = params.fetch("mentor_Id")

    # sends the invitation email if more than 1 day has passed
    # invite was sent. If less than 1 day, redirect to view-men
    if time_now - time_last_send >= 86_400
      invitation_email
    else
      redirect "/view-mentor?id=#{@mentor_id}&error=1"
    end
  end
end

# A small method that constructs and sends an invitation email
def invitation_email
  @id = request.cookies.fetch("id")
  @user = User.first(id: @id)
  @user.has_mentor = @mentor_id
  time_now = Time.new
  @user.last_send = time_now.to_i
  @user.save_changes

  mentor = User.first(id: @mentor_id)
  email = mentor.email

  # Constructs an invitation email to send to a mentor
  subject = "You have been invited to a mentorship!"
  body = "This mentorship is by #{@user.name}. Below is their i
  puts "Sending email..."

  # Sends the invitation email and redirect to mentee page
  if send_mail(email, subject, body)
    puts "Email Sent Ok."
  else
    puts "Sending failed."
  end

  redirect "/mentee"
end
```

Listing E - controllers/controllers_mentee.rb (82 - 132)



```
describe "#load_profile" do
  it "used to load the updated account information of the user" do
    params = { "first_name" => "Alan", "surname" => "Smith", "email" => "AlanS@gmail.com", "password" => "Password1",
               "confirmpassword" => "Password1", "privilege" => "Mentee" }
    updatedInfo = { "first_name" => "Gordon", "surname" => "Ramsay", "email" => "GR11@gmail.com" }
    user.load(params)
    user.load_profile(updatedInfo)
    expect(user.first_name).to eq("Gordon")
    expect(user.surname).to eq("Ramsay")
    expect(user.email).to eq("GR11@gmail.com")
    expect(user.password).to eq("Password1")
    expect(user.privilege).to eq(2)
  end
end
```

Listing F - spec/unit/users_spec.rb (30 - 43)



```
it "allows the mentor to reject the request" do
  mentor_login
  click_link "View More"
  expect(page).to have_content "Choose what you w
  choose("rejectDecision")
  click_button "Submit"
  expect(page).to have_content "Currently you hav
end

it "allows the mentor to accept the request" do
  reset_mentee
  mentee_login
  mentor_filter
  send_invitation
  click_link "Logout"
  mentor_login
  click_link "View More"
  choose("acceptDecision")
  click_button "Submit"
  expect(page).to have_content "Mentee1@gmail.ac.
end

it "allows the mentee to see the mentor's contact
  mentee_login
  expect(page).to have_content "You have been mat
  expect(page).to have_content "Mentor1 TestDudet
  expect(page).to have_content "Mentor1@gmail.com
  reset_mentee
  reset_mentor
end
```

Listing G - spec/features/mentorship.rb (17 - 46)