

## Team Report Iteration 2

### - README -

# E-mentor scheme web application

A web application written in Ruby which allows mentors and mentees to contact each other and exchange information.

## To Start The Web App

Visit <https://codio.com> and log into your account, or register first.

Click on 'My Projects' and then on 'New Project'.

Select Ruby + Rails as the starting point and give your project a name, then create it.

git clone the project

Open the terminal by clicking "Tools", then "Terminal"

Type the following commands in the terminal and press enter after each:

```
'cd project'
```

```
'bundle install'
```

```
'ruby app.rb'
```

Finally, go to the URL of **your app** ('This sinatra application is available at: ') in a new tab.

If you wish to stop the web app press, Control + C in the terminal located in the other tab, the codio box.

## Make an account

**Register** by visiting the login page, where you'll be asked if you have an account, and enter your details to make one. Select your role based on what you would like to be: a mentor or a mentee.

## Contact a mentor

Mentees can use the search and filter functions to look for a suitable mentor. You can enter an introductory email that will be sent to the mentor. As a mentee-driven scheme, it is your responsibility to initiate and maintain contact with your mentor.

## Mentor Acceptance

As a mentor, you can go to your dashboard and view the mentees who have expressed their interest in a mentorship. You can click on 'view more' to see a description of the mentee and accept the invitation.

## Mentorship

Once the mentee and mentor have started the mentorship, they will be able to see each others' emails to start a correspondence.

## Change account credentials

Use the **Profile** menu on the top toolbar to change your credentials.

Simply enter your new credentials and the old ones for verification purposes and your details will be updated when you click Change Details.

## Administration

The **administrator** role is a special role with elevated privileges, such as suspending users, changing account details and viewing and dealing with reports. The initial administrator (the 'founder') can also create other administrator users.

## The report system

Users can **submit reports** to administrators in case something inappropriate happens during the mentoring process. The administrators are then able to view the reports.

### Example User Accounts you can use

<u>Email</u>	<u>Password</u>	<u>User Account type</u>
Mentee1@gmail.ac.uk	Password1	Mentee
Mentee2@gmail.ac.uk	Password1	Mentee
Mentor1@gmail.com	Password1	Mentor
Mentor2@gmail.com	Password1	Mentor
Founder1@gmail.com	Password1	Founder/Admin

## Stories

Our stories changed throughout the lifecycle of our project due to various reasons, and below we will describe some of the main changes made to the original stories to bring us where we are now. Note: we changed how most acceptance criteria is worded as during our first submission we wrote these in first person which turned out to be the incorrect - they were reworded into the correct format, and we will not include this in the acceptance criteria changes below as most would have to be crossed out and changed.

KEY: Changes - ~~changes~~ , Additions - ~~additions~~.

**Story 1:** we changed the story description so that it was more clear what the aim was, and we also removed Acceptance Criteria B as we came to an agreement that it was not required. Acceptance criteria D was a new requirement specified by our client when we showed them our system after the first iteration.

No	Story	Acceptance Criteria	Client Priority	Estimated effort
1	As a mentee, I want to create a mentee account <del>with email and password</del> , so that <del>I can be part of the system</del> <del>so that I can find a mentor.</del>	<p>A. When the <del>create account</del> <del>register</del> button is pressed, an email address and password can be entered.</p> <p><del>B. When the 'submit' button is pressed, an email will be sent to the mentee that contains a confirmation link to confirm the creation of their account.</del></p> <p>C. If no email or password is entered, the 'create account' button will be unavailable.</p> <p>D. <del>The email entered must contain a valid university suffix, or else the user will have to enter another email.</del></p>	Must do	5

**Story 2:** we again changed the wording of the story so that it was clearer, and also re-worded the acceptance criteria to be more specific on how the mentee can search for a mentor. We also removed the clear button acceptance criteria as our system doesn't need one with the way it operates - the mentee can instantly search with different filters, and they do not stack up on each other so there is no need for a 'clear filter' button.

No.	Story	Acceptance Criteria	Client Priority	Estimated effort
2	As a mentee, I want to be able to filter <del>by interests and hobbies</del> mentors, so that I can choose a suitable mentor from a list <del>can request a suitable one.</del>	<p>A. When the 'filter' button is pressed, a list of <del>interests and hobbies</del> industry sectors, subjects, and job titles will appear.</p> <p>B. When specific filters are chosen from a menu, the list of relevant mentors to be updated accordingly.</p> <p>C. <del>When the 'clear filter' button is pressed, all filters will be cleared and the list of mentors will be updated accordingly.</del></p>	Could do	5

**Story 3:** no changes had to be made. The mentee/mentor confirmation email system was implemented as described below.

No.	Story	Acceptance Criteria	Client Priority	Estimated effort
3	As a mentee, I want to be able to see a confirmation email, so that I know that I have a mentor.	<p>A. When a mentor approves a mentee's request, they will receive an email informing them of their acceptance.</p> <p>B. When the invitation has been accepted, the mentor is able to see the mentee's preferred communication method.</p>	Must do	5

**Story 4:** we reworded the story as the system is supposed to be mentee driven, and mentors should not contact the mentees themselves. Rewording of the acceptance criteria is also shown below.

No.	Story	Acceptance Criteria	Client Priority	Estimated effort
4	As a mentee, I want to add a degree and topics of interest to my profile, <del>so that a potential mentor can contact me.</del>	<p>A. When the 'my profile' button is pressed, the user will be directed to profile settings.</p> <p>B. When on 'my profile', a mentee can add the degree they are studying.</p> <p>C. When on 'my profile', a mentee can add their topics of interest to it.</p> <p>D. When the 'submit' button is pressed, the updated details will be saved to their profile.</p>	Should do	8

**Story 5:** reworded 'communicate' to 'arrange a meeting' to make the story description more accurate.

No.	Story	Acceptance Criteria	Client Priority	Estimated effort
5	As a mentee, I want to send a personalised email to mentors, so that I can <del>communicate</del> <b>arrange</b> a meeting with them.	<p>A. When a mentee finds a mentor a suitable mentor, they will be able to send them a mentee invitation.</p> <p>B. When the 'send invitation' button is pressed, the user is directed to a 'send email' page.</p> <p>C. When on the send email page, a mentee will be able to personalise their invitation email.</p> <p>D. When on the send email page, a mentee is able to press the 'send' button to send the personalised email to a mentor.</p>	Should do	2

**Story 6:** changed story description slightly to make it more concise. Rewording of acceptance criteria also can be seen.

No	Story	Acceptance Criteria	Client Priority	Estimated effort
6	As a mentor, I want to create a mentor account <del>with email and password</del> , so that prospective mentees can contact me.	<p>A. When the 'create account' button is pressed, an email address and password can be entered.</p> <p>B. When the 'submit' button is pressed, a confirmation email link will be sent to confirm the creation of an account.</p> <p>C. If no email or password is entered, the 'create account' button will <del>be unavailable</del> <b>not work and the user will see an error message.</b></p> <p>D. When an email entered has already been used, it will display an error message.</p> <p>E. When an invalid email is entered, pressing the 'create account' button will display an error message.</p>	Must do	5

**Story 7:** we reworded the story description to make it clear what the requirement is - previously it was badly worded and now it clearly describes what we need to implement.

No	Story	Acceptance Criteria	Client Priority	Estimated effort
7	As a mentor, I want to <del>add a degree and topics of interest to</del> <b>have the option to update</b> my profile <b>with my job title and industry sector.</b> <del>so that a potential mentee can contact me.</del>	<p>A. The 'my profile' button will take the mentor to profile settings.</p> <p>B. The mentor should be able to change their details in the profile settings.</p> <p>C. The mentor should be able to change the industry sector they are currently working within..</p> <p>D. When the 'submit' button is pressed, the updated details will be saved to the mentors profile.</p>	Should do	3

**Story 8:** we changed the story description to show that the mentors can accept or decline a mentee request.

No	Story	Acceptance Criteria	Client Priority	Estimated effort
8	As a mentor, I want to <del>accept a mentee invitation</del> <b>receive an email from a mentee and have the choice to accept a mentee invitation</b> , so that I can become their mentor.	A. When a mentee sends an invitation to a mentor, the mentor will receive a notification email. B. Mentors should have the option to accept or decline the mentee invitations. C. If the invitation is accepted, the system will send a notification email to the mentee to notify them that their request has been accepted.	Must do	2

**Story 9:** with this story, we initially changed our first idea but ended up going back to it again. This was due to us re-evaluating this story at different times in the project's lifecycle and deciding that ultimately our first concept will be more suitable for our system. We decided that it would be easier to create a new account than promoting an existing one, and also decided that it would be very unlikely that a mentee or mentor would actually get upgraded to an admin on our system.

No	Story	Acceptance Criteria	Client Priority	Estimated effort
9	<del>As the initial administrator, I want to be able to promote accounts, so that there can be other admins apart from me.</del> <b>As the initial administrator, I want to be able to create admin accounts, so that there can be other admins apart from me.</b>	A. <del>When the 'promote to admin' button is pressed, an admin will be able to promote user accounts to admin accounts with elevated privileges.</del> <b>When the 'Admin Creation' page is chosen, the admin can fill in details to create a new admin account which can then be used to log in as an admin on the system.</b>	Should do	8

**Story 10:** in correspondence to our august semester feedback, we reworded the story description in order to make it more concise. We have decided not to implement acceptance criteria A and B as our detail changing system is located on one page where the user can see their current details and can change the ones they wish to change - this is the same on the admin side, as the admin can change the specific fields for any user if required.

No.	Story	Acceptance Criteria	Client Priority	Estimated effort
10	As an administrator, I want to change the clients' registration details, so that mentors and mentees can have their details changed if a problem arises so that the accounts are more secure.	<p>A. <del>When an admin clicks on the "Change Password" button, they are able to enter their new password into the confirm password tab.</del></p> <p>B. <del>When an admin clicks on the "Change Email" button, they are able to enter a user's new email into the confirm email tab.</del></p> <p>C. When an admin has entered their confirmed registration detail, they will then have access to click the "Save Changes" button.</p> <p>D. When an admin clicks on the "Save Changes" button, their new registration details will be saved.</p>	Must do	13

**Story 11:** No changes were made to this story. The administrator can suspend and restore accounts whenever that would be necessary.

No.	Story	Acceptance Criteria	Client Priority	Estimated effort
11	As an administrator, I want to be able to suspend and restore user accounts, so that I can moderate the platform well.	<p>A. When the 'suspend account' button on a user's profile is pressed, the account is suspended.</p> <p>B. When the 'restore account' button on a user's profile is pressed, the account is restored.</p> <p>C. Suspended accounts cannot be used until they are restored.</p>	Should do	13



**Story 12:** no changes were made to this story. The administrator can search for users and their details will be displayed accordingly.

No.	Story	Acceptance Criteria	Client Priority	Estimated effort
12	As an administrator, I want to be able to see a list of all the users and their details.	A. When the 'user accounts' button is clicked by an administrator, more information about the selected users is displayed. B. When a user is selected by an administrator, a list of their account information will be displayed.	Must do	8

**Story 13:** this was the only story that was fully scrapped. We decided that the priority was low and, with the remaining time we had left, it would be more useful to focus on the administration stories in the second iteration. This was one of the stories which we were going to implement in the second iteration - if we were given more time, we would have reconsidered this story.

No.	Story	Acceptance Criteria	Client Priority	Estimated effort
13	As a guest, I want to be able to see the platform, so that I can be encouraged to register.	A. When the system is used as a guest, its features will be displayed normally, but with limited functionality.	Could do	3

**Story 14:** slight tweaks were made to this story within the acceptance criteria. This simply sums up what is expected from the system.

No.	Story	Acceptance Criteria	Client Priority	Estimated effort
14	As a user, I want to be able to change my email and password, so that my account will be secure.	<p>A. When a user tries to change their email or password. They must first input their original to confirm they are the actual user.</p> <p>B. When a new unique email address is entered, an error message is displayed if the new email address is the same as the original.</p> <p>C. When the 'save' button is clicked, the new email or password will be updated for the corresponding account.</p> <p>D. If both email and password are empty, then the 'save' button will be disabled.</p>	Should do	8

**Story 15:** the acceptance criteria was changed slightly as we decided that providing a form for the user to describe the issue would be sufficient for the system. The user can freely describe what has happened, and the administrators will be able to see whatever the users type into the input area. We decided to focus on different aspects of the project instead introducing more features into the report system as the client priority wasn't very high.

No.	Story	Acceptance Criteria	Client Priority	Estimated effort
15	As a user, I want to be able to report a problem to the administrator, so the problem can be resolved.	<p>A. When the 'report a problem' button is clicked, a form will be displayed, including <del>problem categories,</del> 'describe the details about the issue', <del>transcript of the invitation message sent if applicable, and etc.</del></p> <p>B. After writing the problem, the submit button can be clicked and a report will be sent to the administrator.</p>	Could do	5

**Story 16:** within this story we have reworded the first acceptance criteria to be more exact about what the administrator will see. We also removed acceptance criteria C because we thought that the additional features were too complex to implement when the administrator can already do all these things (i.e. suspend/restore accounts, communicating with users).

No.	Story	Acceptance Criteria	Client Priority	Estimated effort
16	As an administrator, I can view help requests submitted by users, so that I can resolve their issues.	<p>A. When the 'view all requests' button is clicked, <del>I will see all the reports sent by users</del> <b>an administrator will see all the help requests</b>.</p> <p>B. When the help requests have been read by the admin, they can be acted upon.</p> <p>C. <del>After selecting a help request, there will be several buttons with common actions such as suspending accounts, restoring accounts, or opening communications with another user.</del></p>	Could do	5

**Story 17:** no changes were made to this story, the user can log into the system as described below.

No.	Story	Acceptance Criteria	Client Priority	Estimated effort
17	As a user, I want to be able to enter a username and password, so that I can login.	<p>A. When the application is used, a username and password can be entered to log in.</p> <p>B. If username and password are empty, then the login button will be disabled.</p> <p>C. If an incorrect username or password is submitted, an incorrect details message will be displayed.</p> <p>D. When the password is forgotten, the user can click the 'forgot password' button to submit a report to the admin for password change.</p> <p>E. When a valid username and password is entered, the user can utilise the full functionality of the website.</p>	Must do	5

## Stories Breakdown

Below is a breakdown of how the stories were split over the two iterations, and a brief description of who worked on the corresponding stories. As you can see, we attempted to split the workload evenly throughout the two iterations to work within an agile methodology.

- **Iteration 1:**

**Proposed stories to implement** (1, 3, 4, 5, 6, 7, 8, 12, 17)

**Actual stories implemented** (1, 2, 3, 4, 5, 6, 7, 8, 14, 17)

Within the first iteration, we ended up completing stories 2 (**As a mentee, I want to be able to filter mentors, so that I can request a suitable one**) and 14 (**As a user, I want to be able to change my email and password, so that my account will be secure**) even though they were not initially in our plans as we realised they would be easy and more useful to implement within the first iteration, as we were already working on similar features at the time. We pushed story 12 (**As an administrator, I want to be able to see a list of all the users and their details**) to the second iteration as we decided that we would leave all the administrator stories for the second iteration.

**Frontend**(Jakub, Wenxiang) - formatted views, worked on the .css, and aimed to make the system look user friendly.

**Backend**(Ariful, Vlad ) - coded, implemented, and debugged the code to make the system work as described for the first iteration.

**Database**(Edmond, Josh) - Implemented a database to be used by the backend team throughout the lifecycle of our project. Tweaks made where necessary to ensure the databases linked where required.

- **Iteration 2:**

**Proposed stories to implement** (9, 10, 11, 12, 13, 15, 16.)

**Actual stories implemented** (9, 10, 11, 12, 15, 16.)

We were able to implement all these stories apart from story 13 (**As a guest, I want to be able to see the platform, so that I can be encouraged to register.**). As iteration time was smaller we did not have enough time to fully work out how the guest features would integrate into our system. However we were also able to fully refactor our database so that there would be no data replication and use standard database getters throughout the system.

**Frontend**(Jakub, Wenxiang) - formatted views, worked on the .css, for the new sections of the website. Also worked on improving the look and feel of the website.

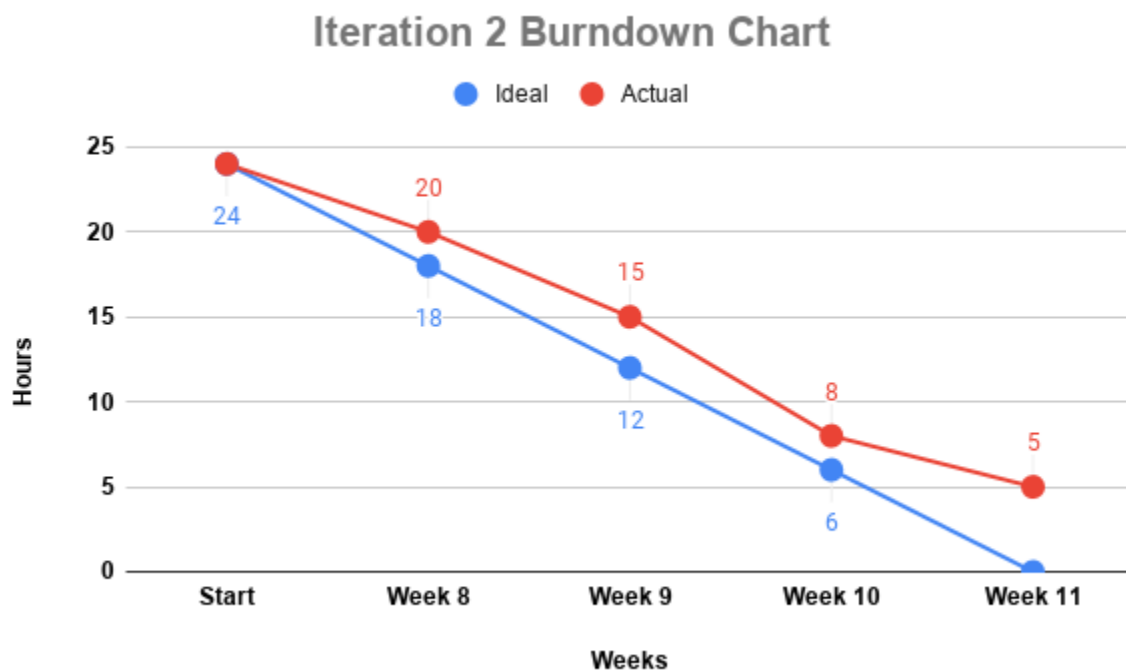
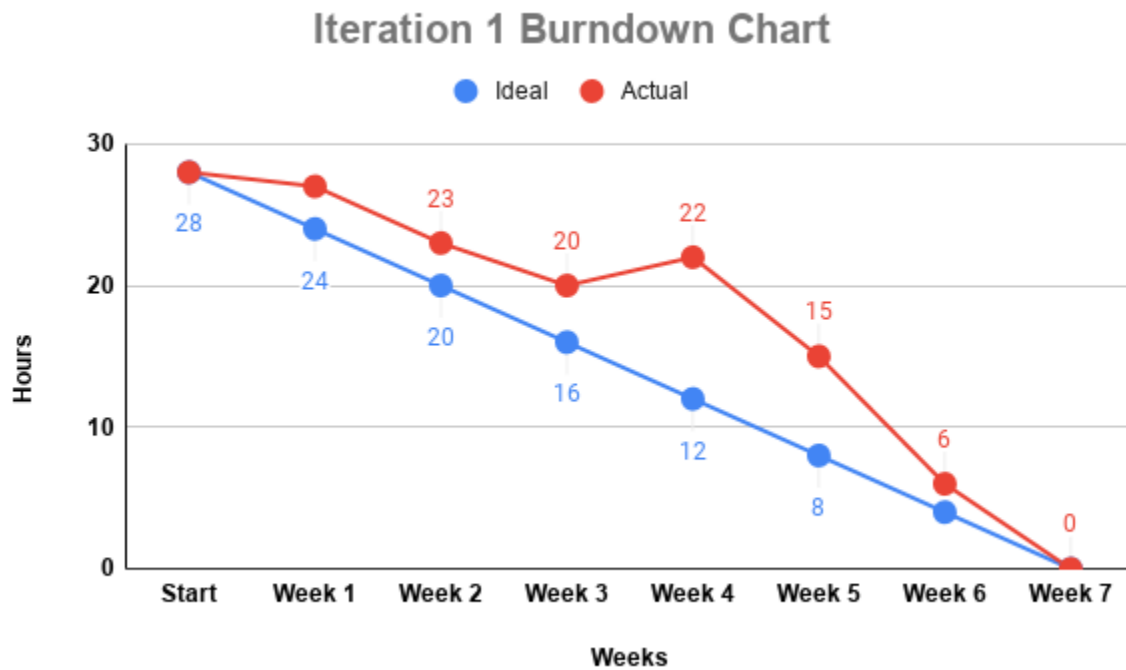
**Backend**(Ariful) - Worked on implementing the new admin features and fixed bugs found from new and old code.

**Testing**(Edmond, Vlad) - Worked on the unit and feature tests, making sure all the website features were working as expected throughout the implementation of new features

**Database**(Josh) - Worked on database refactor to make sure no replication within databases.

**Burndown Chart**

Below are the burndown charts for both iteration 1 and iteration 2, in correspondence with our weekly log:



## Testing and Coverage

Testing was done using both unit and feature tests. Each aspect of the application was taken into consideration when writing the test cases, so many scenarios were covered from the point of view of a user browsing the website. Therefore, a high percentage of coverage was achieved, ensuring that the application is thoroughly tested. Additionally, the testing code was refactored, so the repetitive lines of code from iteration 1 have been replaced with more efficient code that is easier to maintain and understand.

### Screenshots:

```
Finished in 21.81 seconds (files took 1.98 seconds to load)
175 examples, 0 failures

Coverage report generated for RSpec to /home/codio/workspace/project/coverage. 625 / 664 LOC (94.13%) covered.
```

### How to run the tests:

- Open a codio terminal (see the 'To Start The Web App' section)
- make sure the gems are installed and you are in the project directory (by entering 'cd project' into the terminal, followed by 'bundle install' to install all the specified gems in the projects gem file).
- Type 'rspec' in order to run all the tests, both unit and feature
- the result will appear shortly

**Exemplar Code**

```
188 ▸ post "/suspension" do
189   @id = params[:id]
190   @user = User.first(id: @id)
191   date_time = Time.new
192   email = @user.email
193   case @user.suspend
194
195   # Constructs restoration notification email to send to user
196 ▸   when 1
197     @user.suspend = 0
198     subject = "You were unsuspended by an admin on the eMentori
199     body = "You unsuspended at #{date_time.strftime('%R')} on #
200
201   # Constructs suspension email to send to user
202 ▸   when 0
203     subject = "You were suspended by an admin on the eMentoring
204     body = "You suspended at #{date_time.strftime('%R')} on #{c
205     @user.suspend = 1
206   end
207
208   # Sends the constructed email and redirect to dashboard
209   puts "Sending email..."
210 ▸   if send_mail(email, subject, body)
211     puts "Email Sent Ok."
212 ▸   else
213     puts "Sending failed."
214   end
215
216   @user.save_changes
217   redirect "/dashboard"
218 end
```

**Listing A - controllers/controllers\_admin.rb (188 - 228)**

```
220 ▸ get "/view-reports" do
221   @id = request.cookies.fetch("id", "")
222   redirect "/login" if @id == ""
223
224   @user = User.first(id: @id)
225   @founder = true if @user.get_privileges == "Founder"
226   @reports = Report.all
227   erb :view_reports
228 end
229
230 ▸ get "/view-report-detail" do
231   @id = request.cookies.fetch("id", "")
232   redirect "/login" if @id == ""
233
234   report_id = params.fetch("id", "")
235   @report = Report.first(id: report_id)
236   erb :view_report_detail
237 end
238
```

**Listing B - controllers/controllers\_admin.rb (220 - 237)**

```

43      <% if @table_show%>
44      <% if @user_list.count > 0 %>
45      <table class = "centerTable" border="1">
46      <tr id = "mentorTable">
47      <th>Name</th>
48      <th>Email</th>
49      <th>Privilege</th>
50      <th>Action</th>
51      </tr>
52      <% @user_list.each do |user| %>
53      <tr>
54      <td><%= h user.name %></td>
55      <td><%= h user.email %></td>
56      <td><%= h user.get_privileges %></td>
57      <td><a href="/view-user?id=<%= h user.id %>">View
58      </td>
59      <% end %>
60      </tr>
61      </table>
62      <%else %>
63      <p class = "error">
64      Sorry, no users found.
65      </p>
66      <% end %>

```

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

```

225 post "/post-make-report" do
226   id = request.cookies.fetch("id", "0")
227   puts params
228   description = Description.new
229   description.load(params)
230   description.save_changes
231   identifier = params.fetch("identifier", "0")
232   puts identifier
233   time = Time.new.to_s
234   report = Report.new
235   report.load(id.to_i, identifier, description.id, time)
236   description.save_changes
237   report.save_changes
238   redirect "/dashboard"
239 end

```

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



```
82 # Time class works in seconds. 86400 is one day in seconds.
83 # We store the time since they last sent an invite in seconds
84 # We check that the time time in last_Send and current time is
85 # than one day. If so we can do the actual invite/etc
86 # If there is no last_Send we can assume they never sent an i
87 # then we do the invite process.
88 time_now = Time.new
89 if @last_send.nil?
90   @mentor_id = params.fetch("mentor_Id")
91   invitation_email
92 else
93   time_last_send = Time.at(@last_send.to_i)
94   @mentor_id = params.fetch("mentor_Id")
95
96   # sends the invitation email if more than 1 day has passed :
97   # invite was sent. If less than 1 day, redirect to view-ment
98   if time_now - time_last_send >= 86_400
99     invitation_email
100   else
101     redirect "/view-mentor?id=#{@mentor_id}&error=1"
102   end
103 end
104 end
105 end
106
107 # A small method that constructs and sends an invitation email
108 def invitation_email
109   @id = request.cookies.fetch("id")
110   @user = User.first(id: @id)
111   @user.has_mentor = @mentor_id
112   time_now = Time.new
113   @user.last_send = time_now.to_i
114   @user.save_changes
115
116   mentor = User.first(id: @mentor_id)
117   email = mentor.email
118
119   # Constructs an invitation email to send to a mentor
120   subject = "You have been invited to a mentorship!"
121   body = "This mentorship is by #{@user.name}. Below is their i
122   puts "Sending email..."
123
124   # Sends the invitation email and redirect to mentee page
125   if send_mail(email, subject, body)
126     puts "Email Sent Ok."
127   else
128     puts "Sending failed."
129   end
130
131   redirect "/mentee"
132 end
```

**Listing E - controllers/controllers\_mentee.rb (82 - 132)**

```

30 describe "#load_profile" do
31   it "used to load the updated account information of the user" do
32     params = { "first_name" => "Alan", "surname" => "Smith", "email" => "AlanS@gmail.com", "password" => "Password1",
33               "confirmpassword" => "Password1", "privilege" => "Mentee" }
34     updatedInfo = { "first_name" => "Gordon", "surname" => "Ransay", "email" => "GR11@gmail.com" }
35     user.load(params)
36     user.load_profile(updatedInfo)
37     expect(user.first_name).to eq("Gordon")
38     expect(user.surname).to eq("Ransay")
39     expect(user.email).to eq("GR11@gmail.com")
40     expect(user.password).to eq("Password1")
41     expect(user.privilege).to eq(2)
42   end
43 end

```

**Listing F - spec/unit/users\_spec.rb (30 - 43)**

```

17 it "allows the mentor to reject the request" do
18   mentor_login
19   click_link "View More"
20   expect(page).to have_content "Choose what you want to do"
21   choose("rejectDecision")
22   click_button "Submit"
23   expect(page).to have_content "Currently you have no requests"
24 end
25
26 it "allows the mentor to accept the request" do
27   reset_mentee
28   mentee_login
29   mentor_filter
30   send_invitation
31   click_link "Logout"
32   mentor_login
33   click_link "View More"
34   choose("acceptDecision")
35   click_button "Submit"
36   expect(page).to have_content "Menteel@gmail.ac.uk"
37 end
38
39 it "allows the mentee to see the mentor's contact details" do
40   mentee_login
41   expect(page).to have_content "You have been matched with a mentor"
42   expect(page).to have_content "Mentor1 TestDudet"
43   expect(page).to have_content "Mentor1@gmail.com"
44   reset_mentee
45   reset_mentor
46 end

```

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