Assignment T5: Second Iteration

Unleash Shirish - Clinical TriAll

Roopa Bhat - rsb2178
Sarah Hancock - seh2209
Julian Goldberg - jg3796
Sophia Kolak - sdk2147

New features:

- Confirmation messages on UI for all actions: editing info, accepting/rejecting matches
- Emails sent for actions besides just matching with a trial: making an account, updating info, edit trial, new trial, changes in matches
- Added a news feed on the dashboards that shows a participant's recent activity (list of emails they've received corresponding to the actions they've taken on their account)
- Adding a map and calendar view for participants and researchers on the dashboard
    - Map allows for visualizations of trial locations
    - Calendar shows what trials are coming up

Part 1: User Stories

User Story 1

As a researcher, I want to find fitting participants for clinical trials so that I can get results.

My conditions of satisfaction are:

1. The participants I am matched with do not violate the criteria for my trial.
2. I am able to ask potential participants for further information if necessary.
3. I am able to communicate with potential participants.
4. I am able to fulfill my requirement for a number of participants.
5. I am able to create multiple trials.
6. I know which trial a participant matched to.

Test 1: A participant who matches a researcher's criteria is matched with that researcher.

- Create a participant with healthy medical data
- Create a trial looking for healthy people
- The test passes if the participant is notified of the match.
- The test fails if the participant is not notified of the match.

Result: The match does occur and the information is stored correctly in the database. We found bugs on the frontend that caused the trials to load incorrectly onto the page, but we were able to fix those.

Test 2: A researcher is running multiple trials. They know which participants are matched with a given trial.

- Create a new researcher, and add multiple trials to their account.
- Create a new user matching each trial's criteria.
- The test passes if the researcher is effectively informed of which trial each participant matches.
- The test fails if the researcher cannot distinguish which trial a participant has matched or an incorrect trial is indicated.

Result: The information for each trial is shown in the researcher's dashboard. Originally, the trial information was not being loaded on the dashboard, although the trial was being saved in the database. We were able to fix this bug by correcting how we sent our JSON object, so that each trial is properly rendered by the javascript. A researcher can view how many participants have accepted a match to their trial.

Test 3: A researcher does not upload the IRB approval for their clinical trial.

- Researcher tries to create a trial without adding the IRB number
- The test passes if the researcher is not able to create the corresponding clinical trial.
- The test fails if the researcher is able to create a trial without an IRB number.

Result: This test passes because the trial is NOT created when an IRB number is not put in the form. If any part of the form is not filled out, the user will get a warning and will not be able to submit it.

User Story 2

As a potential participant, I want to participate in clinical trials so that I can earn money.

My conditions of satisfaction are:

1. I will know the hourly rate associated with the trials I am matched to.
2. I have the ability to decline.
3. I can access information about the trial.
4. If I meet all criteria for a trial, I will be notified.
5. I am not matched with trials for which I violate screening criteria.

Test 1: A participant can accept the match by indicating that they would like to move further.

- Create a participant with criteria x.
- Create a trial looking for people with criteria x.
- The participant and the trial are matched.
- The test passes if the participant can accept the match and the researcher is emailed with the participant's contact information.
- The test fails if the participant is not able to accept the match.

Result: The test passes because the participant can click on accept match and can wait to get further correspondence from the researcher. Additionally, the researcher is notified once a new participant accepts and is able to contact a participant. The status of the match gets correctly stored in the database.

Test 2: A participant can reject the match.

- Create a participant with criteria x. Submit their information into the database.
- Create a trial looking for people with criteria x.
- The participant and the trial are matched.
- The test passes if the participant is able to reject the match by indicating their choice on the application and the match is removed.
- The test fails if the participant is unable to reject the match and the match is not removed.

Result: The test passes because the participant can click on reject match and the trial is removed from their dashboard. This is also reflected in the database (the match is marked as "rejected", not deleted to avoid re-matching). Also, the researcher does NOT receive a notification.

Test 3: A participant has not input their information so that they can be matched with clinical trial criteria.

- The test passes if the user is not able to become a part of the match database, and if there is a portion of the criteria form that is not filled out, the user will not be able to submit it.
- The test fails if the user does not input the necessary information and is entered into the database anyway.

Result: This test passes because if a participant does not input their information, they cannot sign up and thus cannot be matched without entering the information in the form.

<u>User Story 3</u>

As a potential participant, I want to be shown only clinical trials that I meet the criteria for (age, height, weight, etc.) so that I do not waste time on trials that I don't match.

My conditions of satisfaction are:

1. I will be matched only with trials that I meet the criteria for.

2. I will know the interventions associated with each trial.
3. All trials I am matched with have received IRB approval.

Test 1: A participant gets matched with a trial that they fit the criteria for

- Create a participant with criteria x.
- Create an IRB approved trial looking for people with criteria x.
- The test passes if both parties are notified of the match and the participant can see interventions.
- The test fails if either party is not notified.

Result: This test passes because participants were only matched to trials that they fit all the criteria for. We have implemented the criteria: age, height, weight, gender, nationality, and race. Whenever these fields were the same for both the participant and trial, the participant was notified of a match. We will add further criteria in future iterations.

Test 2: A participant does not fit the criteria for any trials.

- The test passes if a participant without any matches is directed to the "Edit Information" section to edit their personal information.
- The test fails if a participant without any matches is not able to change their personal information.

Result: A participant without any matches is directed to the "Edit Information" section to specify their personal information, so this test passes.

User Story 4

As a potential participant, I want to be able to view the distances of the trials I match with.

My conditions of satisfaction are:

1. When I log in, I will see a list of all trials that I have been matched to
2. If I have multiple trial matches, I will be shown them in order of ascending distance (closest trials first)

Test 1: A participant gets matched with two trials that each has a different location.

- Create a participant with criteria x at location y.
- Create two separate trials looking for people with criteria x at different distances from the participant, within their preferred range.
- Confirm the participant matches to both trials
- The test passes if the participant is notified of both matches and is shown the closer trial first.

- The test fails if the participant is not notified, or if the participant is not shown the trials in order of closest to furthest.

Result: The test passes because the trials on a participant's dashboard are ranked by distance and the one closest to them shows up first. Originally, the front end was not loading the trials in the correct order because of a minor error in our comparison class, which returns all of a user's trials in order of increasing distance. We fixed this bug by correcting our comparison class in the backend Java code, which fixed the frontend.

Test 2: A participant gets matched with two trials that are equidistant from them.

- Create a participant with criteria x at location y.
- Create two separate trials looking for people with criteria x at equal distances from the participant, within their preferred range.
- Confirm the participant matches to both trials
- The test passes if the participant is notified of both matches and is shown both matches in an arbitrary order.
- The test fails if the participant is not shown both of the trials in some order.

Result: The participant's dashboard does show both the trials that they matched with in an arbitrary order (because they're equidistant). The participant gets two separate notifications for each of the matches.

User Story 5

As a potential participant, I want to be able to receive a notification when I match a trial.

My conditions of satisfaction are:

1. I will receive an email notification with the trial I matched with
2. The email will redirect me to the trial information, where I will be able to accept or decline the match

Test 1: A participant gets matched with a trial and accepts the match.

- Create a participant with criteria x.
- Create a trial looking for people with criteria x.
- Confirm the participant matches the trial.
- The test passes if the participant receives an email notification indicating that they've matched with the trial and the researcher receives an email notification indicating that the participant has accepted a match.
- The test fails if either party is not notified.

Result: This test passes because the participant receives a notification when they have matched with a trial. The participant is able to accept it on the frontend by clicking the button. The researcher receives a notification when the participant accepts a match. The database is updated to match the status of the trial match.

Test 2: A participant gets matched with a trial and declines the match.

- Create a participant with criteria x.
- Create a trial looking for people with criteria x.
- Confirm the participant matches the trial.
- The test passes if the participant receives an email notification indicating that they've matched with the trial but the researcher does not receive a notification.
- The test fails if the participant does not receive an email notification or the researcher does receive an email notification.

Result: This test passes because the participant receives a notification when they have matched with a trial. The researcher does not receive a notification because the match gets deleted. The data was correctly stored in the database.

Test 3: A participant gets matched with a trial and ignores the match.

- Create a participant with criteria x.
- Create a trial looking for people with criteria x.
- Confirm the participant matches the trial.
- The test passes if the participant receives an email notification indicating that they've matched with the trial but the researcher does not receive a notification.
- The test fails if the participant does not receive an email notification or the researcher does receive an email notification.

Result: This test passes because the participant receives a notification when they have matched with a trial. The researcher does not receive a notification because the match was ignored. The data was correctly stored in the database.

Part 2: Test plan with equivalence partitions and boundary conditions

Link to the folder in the github repository that contains the automated test suite:

https://github.com/sophiakolak/4156-team-project/tree/master/triAll/src/test/java

Trial.java

Tests: goodTrial, testJustBelowTrial, testAboveTrial, wrongDateTrial, noNameTrial, noDescTrial, lonAboveTrial, noLocTrial, endAboveTrial, payBelowTrial, irbAboveTrial, partNeededBelowTrial, partConfirmedBelowTrial,

	Input	**Eq Class 1**	**Eq Class 2**	**Eq Class 3**
int id	Int >= 0	Int < 0		
String name	String name of trial	Empty string		
String desc	String of trial description	Empty string		
double lat	Double between -90 and +90	Double < -90	-90 < Double < +90	Double > +90
double lon	Double between -180 and +180	Double < -180	-180 < Double < +180	Double > +180
String location	String of location	Empty string		
String start	String of start date	Valid date	Invalid day (e.g. 12/40/20)	Indvalid month (13/2/20)
String end	String of end date	Valid date	Invalid day (e.g. 12/40/20)	Indvalid month (13/2/20)
double pay	Double between 0 and infinity	Double < 0		
int irb	Int with length 4-5	Int with length < 4	Int with length > 5	Int with length 4 or 5
int partNeeded	Int >= 0	Int < 0		
int partConfirmed	IInt >= 0	Int < 0		
int parent	Int >= 0	Int < 0		

Criteria.java

Tests: goodCriteria, testJustBelowCriteria, testJustAboveCriteria, badStringCriteria

	Input	**Eq Class 1**	**Eq Class 2**	**Eq Class 3**
int id	Int >= 0	Int < 0		
int extID	Int >= 0	Int < 0		
int minAge	Int >= 18	Int < 18	Int > 18	Int = 18
int maxAge	Int <= 120	Int < 120	Int > 120	Int = 120
double minHeight	double >= 0	double < 0		
double maxHeight	double >= 0	double < 0		
double minWeight	double >= 0	double < 0		
double maxWeight	double >= 0	double < 0		
String gender	String of gender = female or male	String = female or male	String = any other random string	Empty string
String race	String not empty	String not empty	String not empty	Empty string
String nationality	String not empty	String not empty	String not empty	Empty string

User.java

Tests: testGoodUser, testJustBelowUser, testJustAboveUser, badStringsUser,

	Input	**Eq Class 1**	**Eq Class 2**	**Eq Class 3**
int id	Int >= 0	Int < 0		
double lat	Double between -90 and +90	Double < -90	-90 < Double < +90	Double > +90
double lon	Double between -180 and +180	Double < -180	-180 < Double < +180	Double > +180
String location	String of location	Empty string		
String first	String of first name	Empty string	String that contains non-letters	
String last	String of last name	Empty string	String that contains non-letters	
String email	String of email	Empty string	String not in format of email (no @)	

Notification.java

Tests: goodTimeNotification, badIdMessageNotification, badTimeNotification

	Input	**Eq Class 1**	**Eq Class 2**	**Eq Class 3**
Int id	Int >=0	int < 0		
String time	String of time	Valid time	Invalid time (contains characters)	
String message	String of email	Empty string	String not in format of email (no @)	

Match.java

Tests: goodMatch, goodMatch1, badDistMatch, badDistCorrectStatusMatch, badStatusMatch, testMatches

	Input	**Eq Class 1**	**Eq Class 2**	**Eq Class 3**
int id	Int >= 0	Int < 0		
double distance	Double >= 0	Double < 0		
String status	String of status = pending, accepted, rejected	Empty string	String of one of those three options	String not of those options

Part 3: Branch Coverage

Link to the folder in the repository that contains the coverage test reports:

https://github.com/sophiakolak/4156-team-project/tree/master/triAll/reports

We achieved nearly 100% branch coverage in each of our model classes:

- Trial: 97%
- User: 97%
- Criteria: 96%
- Match: 100%
- Notification: 91% (only one line wasn't covered)

Some branches within these model classes were difficult to cover since they threw very specific exceptions.

Part 4: Continuous Integration

Link to the files in the repository that configure the CI tool:

https://github.com/sophiakolak/4156-team-project/blob/master/build.xml

Link to the CI report (also on our README):

https://travis-ci.com/github/sophiakolak/4156-team-project