

Test plans (Sprint 2)

Common Test Case:

At no time should the web site display a Django error frame or blank screen.

Automation tests are found in sample/tests.py file. These are tests that check whether the correct information for doctors, field workers, and patients is being passed. (And maybe also check whether the edit personal info works)

To run tests, type “python manage.py test sample”

User Story #3: Search box for admin

Task 1: Make the search box on the admin page

- 1) Login to the website as an admin (username: admin, password: admin)
- 2) Click on any of the links (for example: Doctors), there should be a search box on each sub-link.

Task 2: Search the database for the searched term.

- 1) Login to the website as a doctor (username: doctor1, password: doctor).
(To test for some other sub-link, create several instances of that sub link by clicking on the “Add new ..” button at the top-right of the sub-link page.)
- 2) On the top right corner, click the button, click on profile to view the profile
- 3) Click on edit profile to add first name and last name if not already there.
- 4) Click Submit and login to the website as an admin (username: admin, password: admin)
- 5) Click on Doctors, and search for part of, or the full first name, last name or specialties.
- 6) Click the search button, and the page is updated to display only the rows with the searched term.
- 7) Now search for a term which does not appear on the page. We should see 0 results displayed for this search.

User Story #4: table to display the search results for admin

The steps for Task 1 and Task 2 are the same as the steps from US#3 Task 2 since these user stories are connected, and US#3 redirects to US #4.

Task 1: Make a search result page with clickable links to patient/doctor/field worker etc.

Task 2: Redirect US#3 Task 2 to this page

User Story #5: admin sort entries on the search result.

Task 1: Ability to sort the results.

Follow the same steps as US #3, Task 2

- 8) After doing steps 1-7, click on the top bar “First Name” or “Last Name” to sort it by ascending/descending order. Click on it again to toggle sort.

User Story #6: admin edit button beside each entry of the search result

Task 1: Click on any of the links for the results from the search page to edit information

Follow the same steps as US #3, Task 2

8) Click on the first name, which will redirect you to the information page for the sub-link.

Here you can edit the information, and then click "Save" to update it.

User Story #8: doctor can view patient scans

Task 1: Modify case page (and maybe patient page as well) to be able to view uploaded images.

The first 4 steps are for creating a case if there isn't one.

1) Login as a worker (User: worker1, Password: worker)

2) Click "Add new patient" to create a new patient

3) Go back, and click "Add new case" to create a new case.

4) Click on "Browse" to browse an image from your computer, click submit to upload the image. Finish creating the case.

5) Login as a doctor (Username: doctor1, Password: doctor).

6) Click on a case, and then click on a link inside the image sub-field.

7) Opening this link should display a patient image.

User Story #11: doctor can add comments on patient's case

Task 1: Design patient case page

Automated tests for this task are found in tests.py

Task 2: Complete the frontend and backend of "Add new case" page, so that it updates the patient page, and the field worker page.

Follow the same 5 steps from US #8. After logging in as a doctor, the case should show up in the cases table. Similarly, the worker page should also be updated.

Task 3: Make a "Add comments section", which updates the comment section, and sorts by latest comment. There should be a separate section of comments for each doctor.

Follow the same 5 steps from US #8.

6) Click on a case, and then click on "Add comment"

7) If you click submit on an empty comment, it should redirect you to the same page, meaning the comment will not be submitted.

8) Enter some comment, and click submit. The submitted comment should be at the top (if there are multiple comments on the case).

9) Logout from doctor, and login as worker. Add a comment, and once again, the submitted comment should be at the top.

Task 4: Show message to make sure they want to add this comment, because they will not be able to edit or delete this comment

Follow the same steps from previous task. It should prompt you if you are sure you want

to comment, click yes.

User Story #12: doctor can add sub commenting a patient's case

Task 1: Add a sub comment button, that will add "update", or link, a comment made by the doctor, but not edit it.

Follow steps 1-8 from US #11 Task 4 to login as a doctor and add a comment.

1) Next to every comment, there should be a "Add sub-comment" button, click one of them to add a sub-comment

2) The sub-comment should show up indented at the bottom of the comment.

User Story #13: a link to the doctor's profile that contains a history of their comments.

Task 1: Update the doctor's page whenever they add a comment/annotation (we cannot do the annotation part this sprint). Also, next to the displayed comment, show information such as patient name, case ID etc.

Follow steps 1-8 from US #11 Task 4 to login as a doctor and add a comment.

1) The added comment should show up in the History table at the bottom of the doctor's page (If it's hidden, click Show to show the history table)

Follow steps 1-2 from US #12 Task 1 to add a sub-comment. The sub-comment should also show up in the history table.

User Story #14: doctors edit the priority of the patient.

Task 1: Include the ability for doctors to change priority, and update the respective pages.

Automated tests for this task are found in tests.py

User Story #15: doctors and field workers have search bar in home page

Task 1: Make the search box on the doctor's/field workers page

Nothing to test here.

Task 2: Search the database for the searched term.

1) Login as a doctor

2) Search nothing. You should be redirected to the same page.

3) Search for a name, which will take redirect to the search result page. This table should display all the names from the search term. You should be able to click on a name to be redirected to that person's page.

4) Search for a comment by typing a partial comment from one of the comments. This table should display all the comments that contain that sub-comment, as well as a link to the case which the comment is from.

5) Search something that you know for sure isn't on any of the pages. The search result page should contain no results.

Task 3: Redirect to search result page.

Same as previous task

Task 4: Ability to sort the results.

Same as previous task.

Click on any of the top bars to sort it by ascending/descending order. Click on it again to toggle sort.

User Story #20: upload images(field worker)

Task 1: Send data from the “Add new case” page to the server using MMS/SMS

- 1) Send a picture via the Android phone’s MMS app.
- 2) Check if the the picture uploaded to the server or not.

User Story #22: assigning the priority to the patients(field worker)

Task 1: Ability to modify priority. Update respective pages/tables.

Automated tests for this task are found in tests.py

User Story #23: android app for field workers

Task 1: Connect the Django server.

- 1) Login the worker account on the app
- 2) Check if there is a correct message responded form the server.

Task 2: Create an app interface for field works to add new patient.

- 1) Run the Android app emulator, wait for about 10mins to load the app.
- 2) Check if the interface layout is properly or not.

Task 3:Build an app interface for patient’s page.

- 1) Run the Android app emulator.
- 2) Search a patient, then check if the correct list of patient loaded from the server.
- 3) Click on one patient from the patient table
- 4) Check if the patient information loaded correctly with the server data.

Task 4: Organize the patients’ data for the field worker.

- 1) Run the Android app emulator.
- 2) Sign in as a field worker
- 3) Click the save button
- 4) Shut down the WiFi connection Internet on the mobile.
- 5) Check if the patient’s data is loaded.

User Story #24: privilege to add comments to each patient(field worker).

Task 1: Update the “Add comments section” so there is a clear (color) distinction between comments added by field workers and doctors.

Follow the steps from US #11 Task 4 to add a case, and add a comment.

Adding a comment as a doctor should make the background of the comment blue, while adding a comment as a worker should make the background of the comment yellow.