Sprint Planning for sprint 2

User Story #3: Search box for admin

Task 1: Make the search box on the admin page

Estimated hours: 2

Assigned to: Dhaval

Required work: Design django admin functionality to make a search box on every sub-link.

Task 2: Search the database for the searched term.

Estimated hours: 6

Assigned to: Dhaval

Required work: Design django admin functionality with certain attributes and keys for searching the database

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

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

Estimated hours: 5

Assigned to: Dhaval

Required work: Design django admin functionality to display the searched results.

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

Estimated hours: 3

Assigned to: Dhaval

Required work: Design django admin functionality to redirect to the search page when the search button is clicked.

Task 3: Ability to sort the results.

Estimated hours: 5

Assigned to: Dhaval, Andrew

Required work: Design django admin functionality to be able to sort the results by clicking on the top bar/column bar.

User Story #8: doctor can view patient scans

Task 1: Modify patient page to be able to view uploaded images.

Estimated hours: 4

Assigned to: Anar

Required work: Design HTML page which displays an image when the link is clicked. The image is full resolution.

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

Task 1: Design patient case page

Estimated hours: 2

Assigned to: Anar, Jarick

Required work: Create HTML functionality so that doctors can add comments

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

the patient page, doctor's page, and the field worker page.

Estimated hours: 10

Assigned to: Anar, Jarick, Andrew, Dhaval

Required work: Implement django/HTML/Javascript functionality so that you can add a new case and after submitting, the case is created and displayed.

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.

Estimated hours: 4

Assigned to: Anar

Required work: Implement django/Javascript functionality so that adding a comment updates the comment section and puts the comment to the top, sorting by date added.

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

Estimated hours: 1

Assigned to: Anar

Required work: Implement a alert box using HTML when a user clicks on Add Comment button.

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.

Estimated hours: 3

Assigned to: Anar

Required work: Implement django/Javascript functionality so that adding a sub-comment updates the comment section, and the sub comment shows up distinctive from the comment. Sub-comment is also sorted by date added.

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.

Estimated hours: 5

Assigned to: Anar

Required work: Add a button "History" to the drop down bar on the doctor's page. Implement Javascript/django functionality that redirects to a History page containing the doctor's comments.

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.

Estimated hours: 6

Assigned to: Andrew, Dhaval

Required work: Implement Javascript/django functionality to update the priority on the HTML page as well as the database.

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

Estimated hours: 3

Assigned to: Anar, Jarick

Required work: Update HTML page so there is a search box on the doctor's/field worker's page.

Task 2: Search the database for the searched term.

Estimated hours: 5

Assigned to: Anar

Required work: Implement django functionality to search for the typed term in the search box.

Task 3: Redirect to search result page.

Estimated hours: 5

Assigned to: Anar

Required work: Implement HTML/Javascript functionality to display the results from the database in an organized manner. If the search includes a name, it should be clickable and redirectable to the user (field worker/doctor) or patient. If the search includes a case, it should be clickable and redirected to the case page.

Task 4: Ability to sort the results.

Estimated hours: 4

Assigned to:Anar

Required work: Implement Javascript functionality to sort the results by clicking the top bar.

User Story #20: upload images(field worker)

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

Estimated hours: 20

Assigned to: Anar

Required work: Build Django server where it can receive SMS and MMS.

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

Task 1: Ability to modify priority on their own page. Update respective pages/tables.

Estimated hours: 4

Assigned to: Anar, Dhaval

Required work: Implement Javascript/django functionality so the priority gets updated in the database and the HTML page.

User Story #23: android app for field workers

Task 1: Connect the Django server.

Estimated hours: 4

Assigned to: Bao

Required work: Implement and Android Application where it can connect the Django Web server.

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

Estimated hours: 8

Assigned to: Milind

Required work: Implement an xml design for the app interface.

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

Estimated hours: 10

Assigned to: Bao, Milind

Required work: Implement and xml design for the patient's page.

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

Estimated hours: 8

Assigned to: Bao, Milind

Required work: Implement android java code that can organize the patients data in the app.

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.

Estimated hours: 2

Assigned to: Anar

Required work: Implement CSS/Bootstrap/Javascript functionality so that doctor's comments are a different color than worker's comments.