Implementation and Testing Report

Log In Accounts
Server Website: http://ultra-remote-medicine.herokuapp.com/

UserName: admin; Password: admin
(Admin account at http://ultra-remote-medicine.herokuapp.com/admin/ only)

UserName doctor1; Password: doctor

UserName worker1; Password: worker

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

User Story 1:
Andrew Pelegris: I worked on models.py, which both controls the database schema and controls how the database is viewed and edited by an administrator. These changes, combined with Django's existing administration framework, allow an administrator to log in securely, change their own personal information (such as their user-name and password), and edit the now fully implemented database, with full control.

Task 1; Set up the Database:

1. Log in as an administrator (UserName: admin; Password: admin) to the website.
2. Open up the first table.
3. Try Creating 3 new elements for that table.
    a. Try leaving all fields blank. If there are any required fields, they should indicate that they are mandatory and it should not be possible to complete the submission process.
    b. Try filling in all fields. Test the limits of each field by using long strings where applicable, incorrect valued-data (such as strings where a number is expected), and contradictory date-time entries (such as a start-time that occurs later than an end-time in a schedule TimeSlot). Check that the appropriate error messages are delivered and that the form can't be submitted when there are violations.
    c. Try a normal, reasonable case.

Check that no fields are unnecessarily made mandatory either (mandatory fields have bolded title text) and that there are no spelling mistakes.

4. On the list of table entries, check that the string displayed there matches specifications.
5. Repeat for all tables in the database.

Task 2: Build a webpage for the administrator to change their information.

1. Open the Users table.
   Check: The user-name of the current user should be in the list of users.
2. Click on the link corresponding to the current user.
3. All of the current user's information should appear and be editable. The Username, Last Login, and Date Joined fields should all be mandatory.
4. Press "save" and then revisit the user editing changes. All changes made should be present.

Task 3: Build a webpage to add/modify doctor and field worker accounts.

Testing for this task is covered by the tests from Task 1.

User Story 7:

Task 1: Build a log-in page for Doctors.

1. A log-in page should be displayed as the first page when the site is visited. Check that the text and formatting match specifications.
2. Use the actual log-in credentials of a registered Doctor.
   Check that the tester is brought to the Doctor's personal page.
3. Return to the root web-page and try to log in with fake credentials. Check that a message to the effect that the credentials have been rejected comes up.
4. Try logging in with a Field Worker's credentials. Check that the Field Worker is not taken to a version of the Doctor's personal page.

Task 2: Build a page that allows Doctors to change their personal information.

1. Log in to the website as a Doctor (UserName doctor1; Password: doctor).
2. At the top of the Doctor's personal page there is a box containing their information with an edit button. Click the edit button.

3. Try making the following 3 changes.
    a. Blank all the fields. Check that all required fields indicate that they are mandatory and that it isn't possible to submit the changes.
    b. Try filling in all fields. Test the limits of each field by using long strings where applicable and incorrect valued-data (such as strings where a number is expected). Check that the appropriate error messages are delivered and that the form can't be submitted when there are such violations.
    c. Try submitting a reasonable and normal set of changes.

    Check that no fields are unnecessarily made mandatory either (mandatory fields have bolded title text) and that there are no spelling mistakes.

User Story 19:

Task 1: Build a page for adding new patients.

1. Log in as a field worker (UserName worker1; Password: worker).
2. In the field worker's personal page, click the button called "Add New Patient"
3. Try to create 3 new patients:
    a. Don't fill in any fields and then try to submit. Check that an error message comes up indicating that the name and picture identification fields are mandatory.
    b. Fill in every field with information. Test the limits of each field by using long strings. In the picture identification field, try uploading a file from an unsupported or non-image format. Check that the appropriate error messages are delivered and that the form can't be submitted when there are violations.
    c. Create a normal patient record, complete with reasonable information.

    Check that no fields are unnecessarily made mandatory either (mandatory fields have bolded title text) and that there are no spelling mistakes.

4. Check that after successfully creating a patient, the tester is taken to the newly created patient's page. All information entered in the create-patient form should have been carried over.

User Story 21:

Task 1: Build a log-in page for Field Workers.

1. A log-in page should be displayed as the first page when the site is visited.
   Check that the text and formatting match specifications.
2. Use the actual log-in credentials of a registered Worker.
   Check that the tester is brought to the Worker's personal page.
3. Return to the root web-page and try to log in with fake credentials. Check that a message to the effect that the credentials have been rejected comes up.
4. Try logging in with a field Doctor's credentials. Check that the Doctor is not taken to a version of the Worker's personal page.

Task 2: Build a page that allows Doctors to change their personal information.

1. Log in to the website as a Worker (UserName worker1; Password: worker).
2. At the top of the Worker's personal page there is a box containing their information with an edit button. Click the edit button.
3. Try making the following 3 changes.
   a. Blank all the fields. Check that all required fields indicate that they are mandatory and that it isn't possible to submit the changes.
   b. Try filling in all fields. Test the limits of each field by using long strings where applicable and incorrect valued-data (such as strings where a number is expected). Check that the appropriate error messages are delivered and that the form can't be submitted when there are such violations.
   c. Try submitting a reasonable and normal set of changes.

   Check that no fields are unnecessarily made mandatory either (mandatory fields have bolded title text) and that there are no spelling mistakes.