MediConnection

Test Plan Structure (Validation Tests)

Scope:

The scope of this test plan is to validate the functionality of the MediConnect website's three significant use cases, namely:

1. Scheduling appointments
2. Access Medication
3. Chatting with physicians
4. Searching for a physician

Approach:

The testing approach will be followed as such:

1. Integration tests to be performed on subunits that form clusters. May not be necessary for every endpoint in our use case due to the specific type of structure of code achieved.
2. Regression testing to be performed after any modification to a unit/endpoint. Tests to be adjusted accordingly.
3. After the tests have been completed and the product is in a state suitable for validation testing, a series of validation tests to be performed on the deployed system (Alpha testing).
4. Interface testing for front-end by a group of users, including our team, to test content, navigation, and front-end components.
5. Backend endpoints to be tested by the development team.
6. Frontend tests, Beta tests, and Interface testing to be performed by the development team.

Validation Tests

Use Case 1: Book Appointment

Test cases:

1. Verify that the user can select an available time slot on the physician's calendar.
2. Verify that the user can schedule an appointment and receive a confirmation email.
3. Verify that the user can cancel an appointment and receive a cancellation email.
4. Verify that the user can reschedule an appointment and receive a confirmation email.
5. Verify that the user can view their upcoming appointments in the appointments section.
6. Verify that the user receives an error message if they try to schedule an appointment during the physician's unavailable hours.

Use Case 2: Contact Doctor

Test cases:

1. Verify that the user can start a chat session with the selected physician.
2. Verify that messages are rendered properly.
3. Verify that users can delete messages.
4. Verify that the user can send and receive messages in real-time.
5. Verify that the user can resume a previous chat session with a physician.

Use Case 3: Access Medication

Test cases:

1. Verify that physicians can add medication.
2. Verify that patients can view medications assigned to them.
3. Verify that physicians can delete medications.
4. Verify that updates to medication on the physician side reflects on the patient side.
5. Verify that patients cannot mutate medication lists.

Use Case 4: Search Physician

Test cases:

1. Verify that the search bar on the homepage returns relevant results based on the user's search query.
2. Verify that the user can filter the search results based on the physician's name.
3. Verify that the user can filter the search results based on specialization.
4. Verify that the user can view the physician's profile.

Alternate scenarios:

5. Verify that the search bar returns results even if the user misspells the physician's name or specialty.