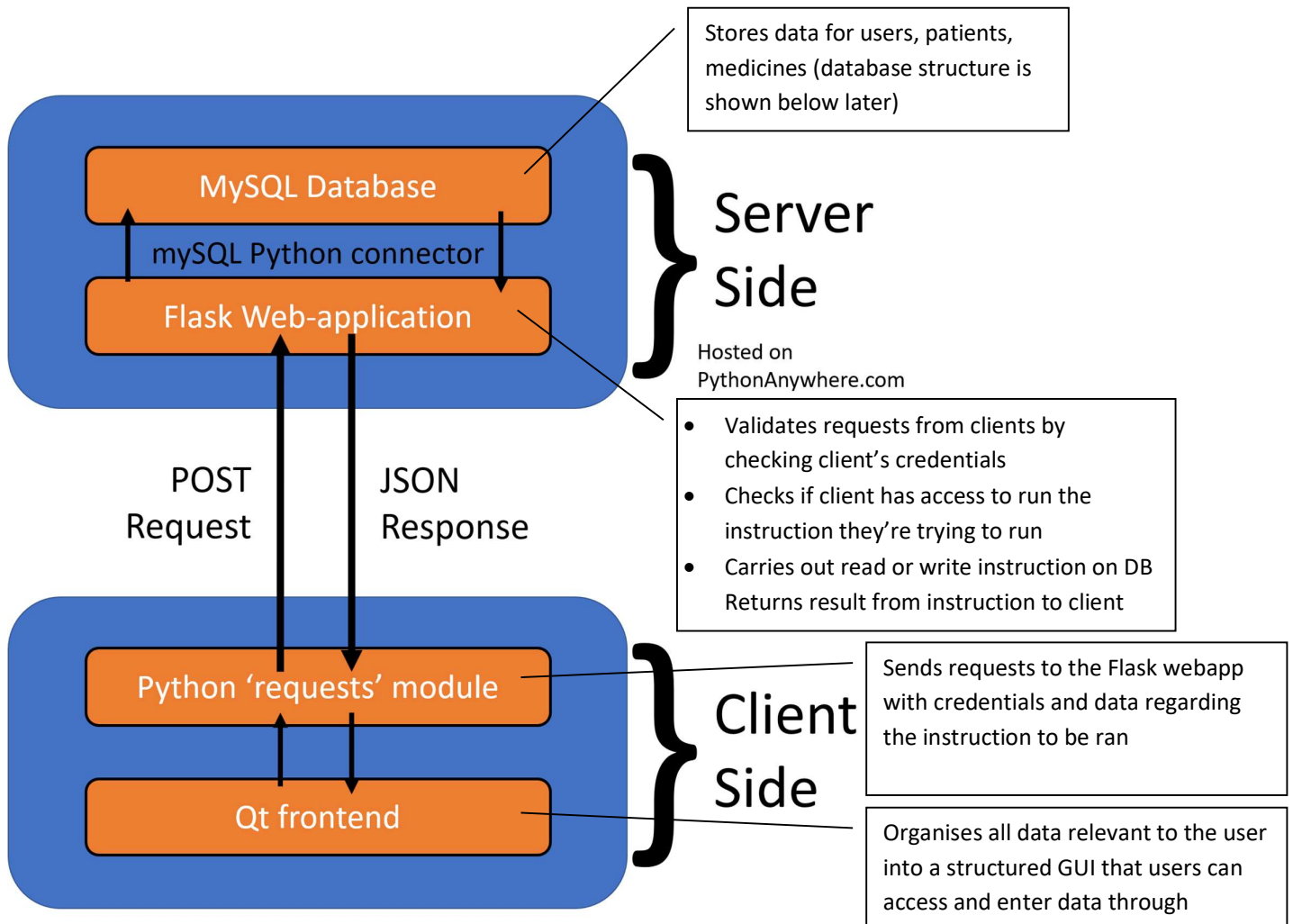
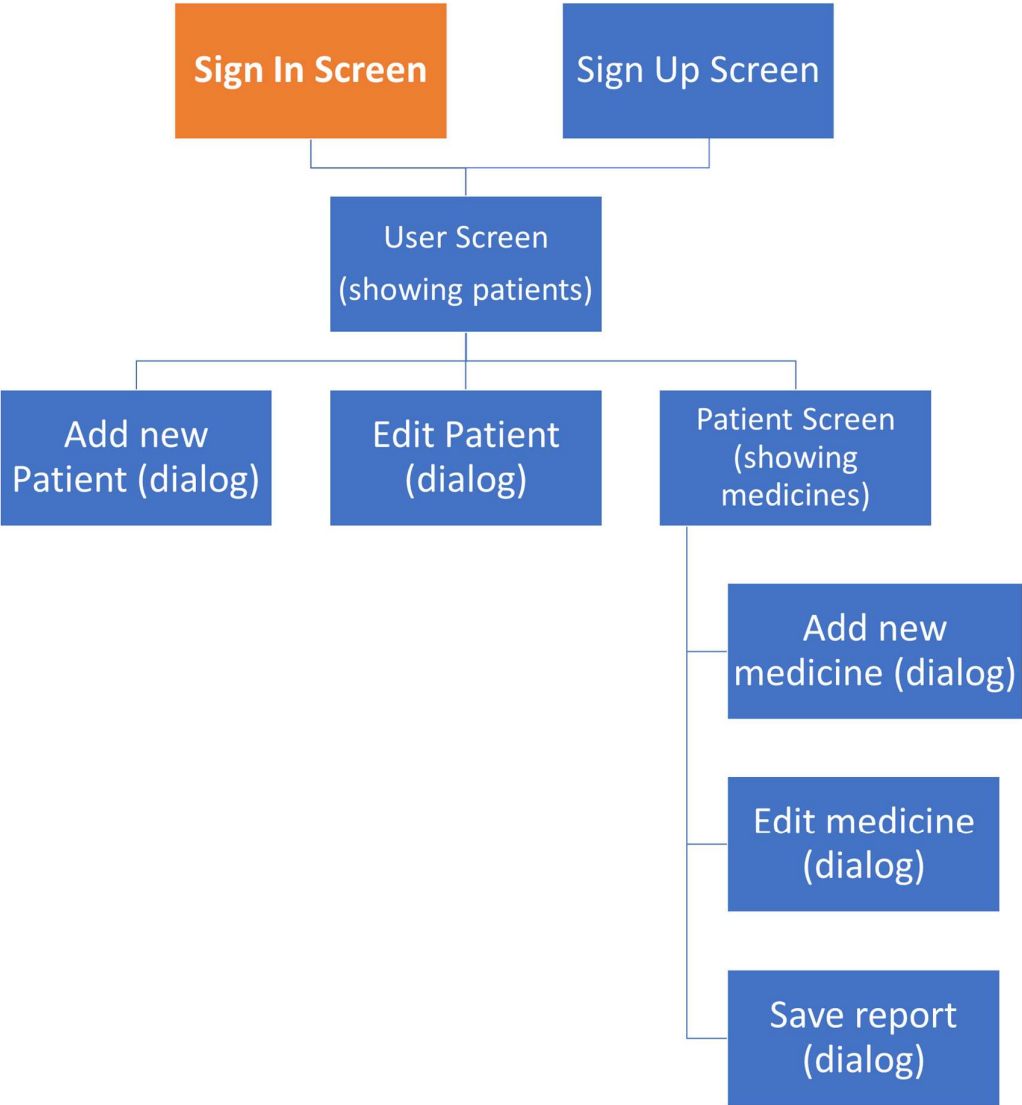


Criterion B: Design

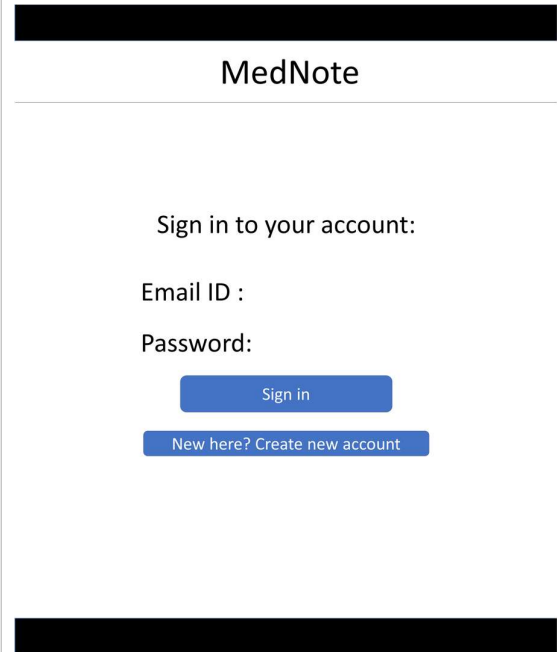
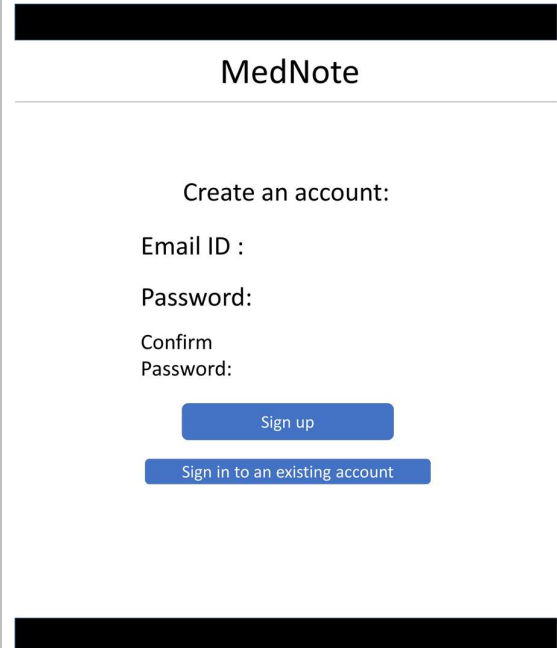
Product Structure Overview



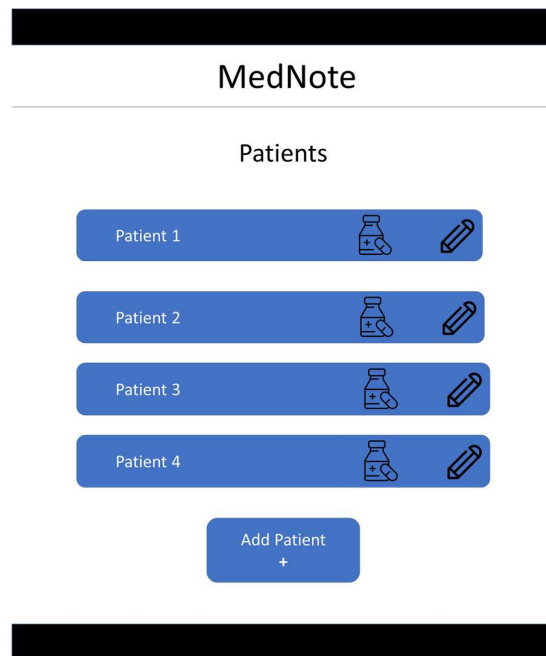
User Interface Overview



User Interface Screen Design

Screen	Purpose
<p>Sign In Screen:</p>  <p>The Sign In screen features a black header bar. Below it, the 'MedNote' logo is centered. A horizontal line separates the header from the main content area. The text 'Sign in to your account:' is displayed. Below this are two input fields labeled 'Email ID :' and 'Password:'. A blue 'Sign in' button is positioned below the password field. At the bottom of the main content area, there is a link that says 'New here? Create new account'.</p>	<p>Allow user to:</p> <ul style="list-style-type: none">• sign in using their existing credentials• switch to the 'sign up' screen if they need to create a new account
<p>Sign Up Screen:</p>  <p>The Sign Up screen features a black header bar. Below it, the 'MedNote' logo is centered. A horizontal line separates the header from the main content area. The text 'Create an account:' is displayed. Below this are three input fields labeled 'Email ID :', 'Password:', and 'Confirm Password:'. A blue 'Sign up' button is positioned below the 'Confirm Password:' field. At the bottom of the main content area, there is a link that says 'Sign in to an existing account'.</p>	<p>Allow user to:</p> <ul style="list-style-type: none">• create a new account• switch to the 'sign in' screen if they want to use an existing account.

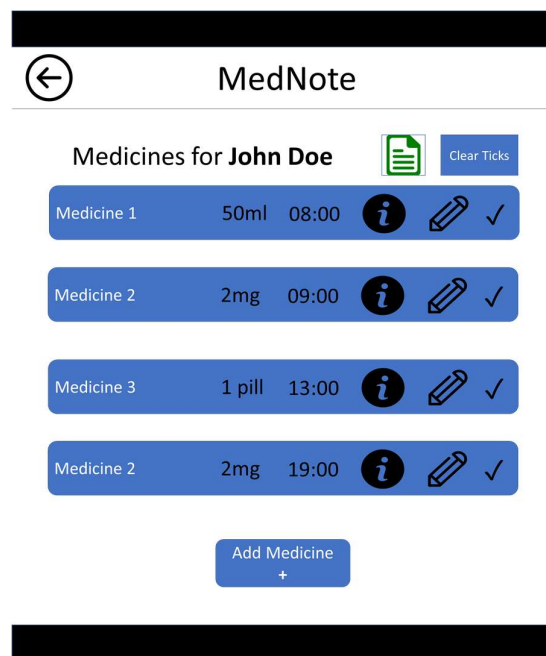
User Screen:



Allows user to:

- see the patients available under their account
- add new patients
- edit existing patients
- choose a patient to view medicines for



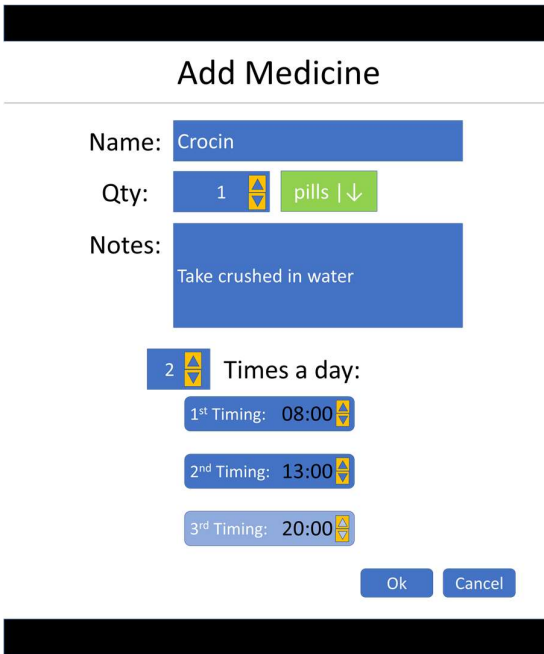
Patient Screen:



Allows user to:

- see the medicines for a certain patient, with different timings listed in different rows, and quantities and timings mentioned
- open notes dialog for a medicine
- add new medicines
- edit existing medicines
- tick a medicine taken at a certain timing
- clear all medicine timing ticks
- save a report of medicines taken
- Go back to the user screen

User Interface Dialogs Design

Dialog	Purpose
<p>Add patient dialog:</p>  <p>The 'Add Patient' dialog box has a title bar 'Add Patient'. Below it is a label 'Name:' followed by a text input field containing 'John Doe'. At the bottom right are two buttons: 'Ok' and 'Cancel'.</p>	<p>Allow users to add a new patient with a specified name</p>
<p>Edit patient dialog:</p>  <p>The 'Edit Patient' dialog box has a title bar 'Edit Patient'. Below it is a label 'Name:' followed by a text input field containing 'John Doe'. Below the input field is a button labeled 'Delete Patient'. At the bottom right are two buttons: 'Ok' and 'Cancel'.</p>	<p>Allow users to edit an existing patient's name or delete their record</p>
<p>Add medicine dialog:</p>  <p>The 'Add Medicine' dialog box has a title bar 'Add Medicine'. Below it are three labels: 'Name:', 'Qty:', and 'Notes:'. The 'Name:' label is followed by a text input field containing 'Crocin'. The 'Qty:' label is followed by a numeric input field containing '1' and a unit dropdown menu currently showing 'pills' with a downward arrow. The 'Notes:' label is followed by a text area containing 'Take crushed in water'. Below these are three timing labels: '2 Times a day:', '1st Timing: 08:00', '2nd Timing: 13:00', and '3rd Timing: 20:00'. Each timing label is followed by a time input field. At the bottom right are two buttons: 'Ok' and 'Cancel'.</p>	<p>Allow users to add a new medicine including:</p> <ul style="list-style-type: none">• its name• its quantity and the unit for the quantity (pills, ml, mg)• notes• the times of the day when it is to be taken (upto 3)

Edit medicine dialog:

Edit Medicine

Name:

Crocin

Qty:

1

pills | ↓

Notes:

Take crushed in water

2

Times a day:

1st Timing:

08:00

2nd Timing:

13:00

3rd Timing:

20:00

Delete

Ok

Cancel

Allow users to edit all details of an existing medicine or delete it from the patient's record

Save report dialog:

Do you want to save a report of the medicines taken?

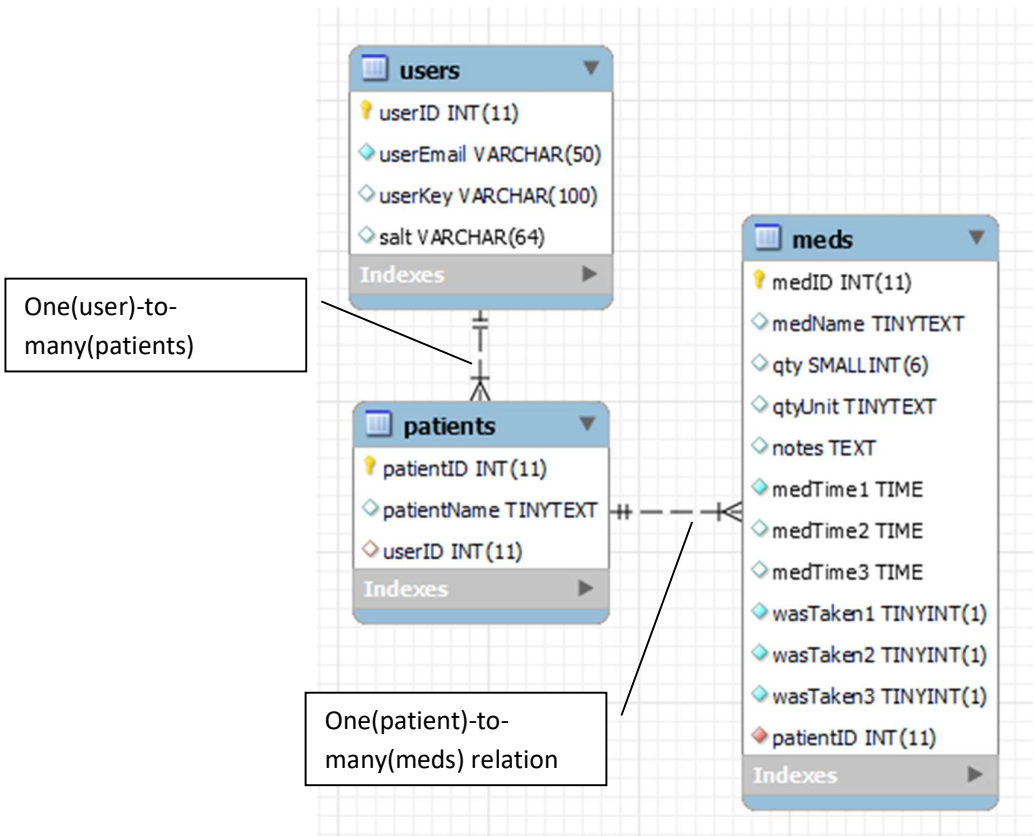
File Name: John Doe 2021-11-23

Yes

No

Allow users to save the report with a specified file name

Database Structure



Client-Server Interactions

Interactions between the client and server occur in the form of the client sending a request to the server with an instruction, its credentials, and data, and the server giving a response with a status and the relevant requested data. If the client makes an invalid request, the server responds with status 'false' and an error message in the data field.

Instruction	Description	Client POST request data	Server JSON response
registerUser	Register a new user account	Credentials (user email & password)	Status: true if signup successful, false if error (email already exists) Data if request is valid: nothing
loginUser	Log into existing user account / validate credentials	Credentials	Status: true if login successful, false if error (email doesn't exist, or password doesn't match) Data if request is valid: nothing
fetchData	Fetch all of user's stored data	Credentials	Status: true if authentication successful, else false Data if request is valid: table for patients (ID, name) and table for meds (ID, name, qty, qty Unit, notes, times (x3), ticked-or-not (x3), patient ID)
addPatient	Add new patient to database	Credentials Patient Name	Status: true if addition successful, false if error (patient name already exists under user, or maximum patient count exceeded) Data if request is valid: patient ID
editPatient	Edit existing patient in database	Credentials Patient ID Patient New Name	Status: true if edit successful, false if error (patient name already exists under user, or patient does not belong to user)

			Data if request is valid: nothing
deletePatient	Delete an existing patient from database	Credentials Patient ID	Status: true if deletion successful, false if error (patient does not belong to user) Data if request is valid: nothing
addMed	Add new medicine to database	Credentials Med Name Med Quantity and Quantity Unit Notes Med Timing (upto 3) Patient ID	Status: true if addition successful, false if error (medicine name already exists under user, maximum medicine count exceeded, or patient does not belong to user) Data if request is valid: med ID
editMed	Edit existing medicine in database	Credentials Med ID Med Name Med Quantity and Quantity Unit Med Notes Med Timing (upto 3) Med taken-or-not (upto 3) Patient ID	Status: true if edit successful, false if error (medicine name already exists under user (with other med ID), or medicine does not belong to user) Data if request is valid: nothing
deleteMed	Delete an existing medicine from database	Credentials Med ID	Status: true if deletion successful, false if error (medicine does not belong to user) Data if request is valid: nothing
tickMedTiming	Mark a particular timing of a medicine as ticked or unticked in the database	Credentials MedID Med Timing Index (1,2, or 3)	Status: true if tick/untick successful, false if error (medicine does not belong to user)

		tickState (ticked or not, 1 or 0)	Data if request is valid: nothing
clearTicks	Mark all timings for all medicines under a patient as unticked in the database	Credentials Patient ID	Status: true if unticking successful, false if error (patient does not belong to user) Data if request is valid: nothing

Code directory structure

Client-side:		Server-side:	
File	Description	File	Description
appClient.py	Defines different functions for every instruction, each containing the format for sending server requests. Separation from appMain.py makes client-side code easy to manage & understand	appServer.py	Receives requests from clients, checks what instruction the request is, and accordingly calls a function from dbFunctions.py
.ui files	Auto-generated by Qt Creator, equivalent to xml files that dictate the screen and dialog designs	creationScript.sql	Auto-generated from MySQL workspace, contains the SQL script defining the database structure
Assets: images	Images to be imported for use in user interface	createDB.py	Connect to local address and initialises mySQL database by executing creationScript.sql (only run once)
appMain.py	<ul style="list-style-type: none"> Converts .ui files to .py files and imports them as classes, which serve as parent classes for the final screens & dialog classes Defines all user interface logic by binding methods and UI elements. Calls on functions from appClient.py 	dbFunctions.py	<ul style="list-style-type: none"> Connects to mySQL database through the mySQL-Python connector Defines different functions for every instruction. Each function contains SQL scripts which are combined with details from the client request to access or make changes to the database

	<ul style="list-style-type: none"> Saves program output (medicine report) in local file directories 		
--	--	--	--

Test Plan

No.	Action to test	Method of testing
T1	Creating a user account	Try entering invalid registration details (invalid email, mismatching passwords, short password) to ensure error is shown, then enter valid details to check if a new account with blank records is created
T2	Signing into user account	Try entering invalid account details (non-existent email, incorrect passwords) to ensure error is shown, and then valid details to check if the existing account with existing records is shown
T3	Adding, renaming, and deleting patients	<p>Try entering invalid patient names (blank field, duplicate name) to ensure error is shown, and then valid unique names to check if new patients are created.</p> <p>Try renaming existing patient to another existing patient's name to ensure error is shown, then to a valid unique name to check if patient is renamed.</p> <p>Delete patients to check if they are removed from view.</p>
T4	Adding medicines	<p>Try entering invalid medicine names (blank field, duplicate name) to ensure error is shown, and then valid unique names to check if new medicines are created.</p> <p>Try entering medicine quantity, unit, and times using the interface.</p>
T5	Editing medicines	<p>Try renaming existing medicine to another existing medicine's name to ensure error is shown.</p> <p>Edit medicine's 'number of times taken' to check if new medicine timings are shown on the screen view, and that any ticks for previous timings are preserved.</p>
T6	Deleting medicines	Delete medicines to check if all timings for a medicine are removed from view.

T7	Ticking/unticking medicines	Ensure clicking on the tick button for any medicine timing toggles the 'ticked' state (for only that timing and not for other timings of the same medicine).
T8	Viewing notes for medicines	Click on the information icon for any medicine to check if the description added for that medicine shows up.
T9	Clearing ticks	Click on the 'clear ticks' button and check if all ticked medicine timings are restored to the unticked state.
T10	Saving Report	<p>Click on the 'save report' icon, and try saving the file with an invalid name to ensure an error is shown.</p> <p>Try saving the file with an existent filename to check if "overwrite?" prompt is shown, and first select 'no' to check if the file is not overwritten, and then select 'yes' to check if it is.</p> <p>Try saving the file with a new/unique name to ensure it is saved.</p>
T11	All records being preserved when product is opened again	Close the product and reopen to it to check if all patient and medicine records have been maintained.
T12	Password being stored securely in database	Run an SQL script in the online database to view the 'users' database and ensure no password is stored in plaintext