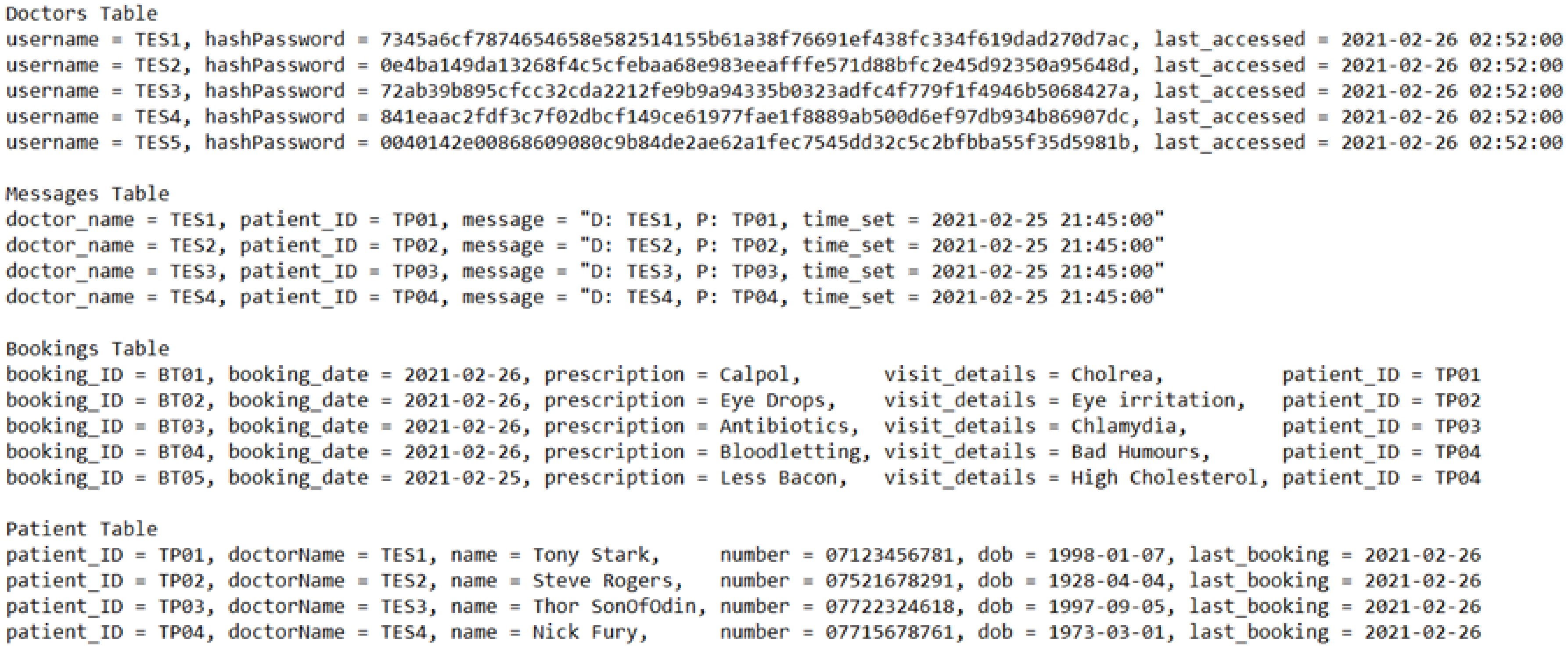
Sprint 1 Test Logs.

These are the Test Logs for the first sprint. The tests that have failed the first time around have been repeated once changes have been made and have been marked as passed. Before each test is performed the jUnit setUp method adds several test records in the database for both ‘Doctors’ and ‘Messages’ tables. After each test these records are removed as well as several linked lists cleared and objects set to null to reset them for other tests.

The values I have used for testing are listed below, these have been inserted into the database in their respective tables before every test and removed after every test:



```
Doctors Table
username = TES1, hashPassword = 7345a6cf7874654658e582514155b61a38f76691ef438fc334f619dad270d7ac, last_accessed = 2021-02-26 02:52:00
username = TES2, hashPassword = 0e4ba149da13268f4c5cfebaa68e983eeafffe571d88bfc2e45d92350a95648d, last_accessed = 2021-02-26 02:52:00
username = TES3, hashPassword = 72ab39b895cfcc32cda2212fe9b9a94335b0323adfc4f779f1f4946b5068427a, last_accessed = 2021-02-26 02:52:00
username = TES4, hashPassword = 841eaac2fdf3c7f02dbcf149ce61977fae1f8889ab500d6ef97db934b86907dc, last_accessed = 2021-02-26 02:52:00
username = TES5, hashPassword = 0040142e00868609080c9b84de2ae62a1fec7545dd32c5c2bfbba55f35d5981b, last_accessed = 2021-02-26 02:52:00

Messages Table
doctor_name = TES1, patient_ID = TP01, message = "D: TES1, P: TP01, time_set = 2021-02-25 21:45:00"
doctor_name = TES2, patient_ID = TP02, message = "D: TES2, P: TP02, time_set = 2021-02-25 21:45:00"
doctor_name = TES3, patient_ID = TP03, message = "D: TES3, P: TP03, time_set = 2021-02-25 21:45:00"
doctor_name = TES4, patient_ID = TP04, message = "D: TES4, P: TP04, time_set = 2021-02-25 21:45:00"

Bookings Table
booking_ID = BT01, booking_date = 2021-02-26, prescription = Calpol,       visit_details = Cholrea,          patient_ID = TP01
booking_ID = BT02, booking_date = 2021-02-26, prescription = Eye Drops,    visit_details = Eye irritation,   patient_ID = TP02
booking_ID = BT03, booking_date = 2021-02-26, prescription = Antibiotics,  visit_details = Chlamydia,        patient_ID = TP03
booking_ID = BT04, booking_date = 2021-02-26, prescription = Bloodletting, visit_details = Bad Humours,      patient_ID = TP04
booking_ID = BT05, booking_date = 2021-02-25, prescription = Less Bacon,   visit_details = High Cholesterol, patient_ID = TP04

Patient Table
patient_ID = TP01, doctorName = TES1, name = Tony Stark,     number = 07123456781, dob = 1998-01-07, last_booking = 2021-02-26
patient_ID = TP02, doctorName = TES2, name = Steve Rogers,   number = 07521678291, dob = 1928-04-04, last_booking = 2021-02-26
patient_ID = TP03, doctorName = TES3, name = Thor SonOfOdin, number = 07722324618, dob = 1997-09-05, last_booking = 2021-02-26
patient_ID = TP04, doctorName = TES4, name = Nick Fury,      number = 07715678761, dob = 1973-03-01, last_booking = 2021-02-26
```

The test table for sprint one is shown below:

Test Name	***Test Purpose***	***Test Date***	***Expected Result***	***Actual Result***	***Test Passed?***	***Rectifications***
loginTestCorrect	Test that the correct records are pulled from the database. 5 assertTrues used	24/02/21	TRUE TRUE TRUE TRUE TRUE	TRUE TRUE TRUE TRUE TRUE	Yes	None
loginTestIncorrect1	Login with completely incorrect credentials and ensure no records are pulled 3 assertTrues used	24/02/21	FALSE FALSE FALSE	FALSE FALSE FALSE	Yes	None
loginTestIncorrect2	Login with correct username and incorrect password 3 assertTrues used	24/02/21	FALSE FALSE FALSE	FALSE FALSE FALSE	Yes	None
loginTestIncorrect3	Login with incorrect username and correct password 3 assertTrues used	24/02/21	FALSE FALSE FALSE	FALSE FALSE FALSE	Yes	None
getNewMessagesTest1	Get a single message for the doctor logged in as 'TES1'	25/02/21	Linked list containing the string : 'D: TES1, P: TP01'	[]	No	Modify the SQL query so that before every test the time of the message sent is changed to a time

						in the future to simulate a new message being sent after login
getNewMessageTest2	Get a single message for the doctor logged in as ‘TES2’	25/02/21	Linked list containing the string : ‘D: TES2, P: TP02’	[]	No	Modify the SQL query so that before every test the time of the message sent is changed to a time in the future to simulate a new message being sent after login
getNewMessageTest1 - redo	Get the a single message for the doctor logged in as ‘TES1’	25/02/21	Linked list containing the string : ‘D: TES1, P: TP01’	[]	No	Use the simple date format api to format the updated string with the new message sent time to match the format in the time_sent field in messages
getNewMessageTest1 - redo	Get the a single message for the doctor logged in as ‘TES1’	26/02/21	Linked list containing the string : ‘D: TES1, P: TP01’	Linked list containing the string : ‘D: TES1, P: TP01’	Yes	None
getNewMessageTest2 - redo	Get the a single message for the doctor logged in as ‘TES2’	26/02/21	Linked list containing the string : ‘D: TES2, P: TP02’	Linked list containing the string : ‘D: TES2, P: TP02’	Yes	None

getNewMessageTest3	Get the a single message for the doctor logged in as ‘TES3’	26/02/21	Linked list containing the string : ‘D: TES3, P: TP03’	Linked list containing the string : ‘D: TES3, P: TP03’	Yes	None
getNewMessageTest4	Get the a single message for the doctor logged in as ‘TES3’	26/02/21	Linked list containing the string : ‘D: TES4, P: TP04’	Linked list containing the string : ‘D: TES4, P: TP04’	Yes	None
getNewMessageTest5	Get the three messages in the database for the doctor logged in as ‘TES1’	26/02/21	Linked list containing the strings : ‘D: TES1, P: TPO1’, ‘D: TES1, P: TPO3’, ‘D: TES1, P: TPO3’	Linked list containing the strings : ‘D: TES1, P: TPO1’, ‘D: TES1, P: TPO3’, ‘D: TES1, P: TPO3’	Yes	None
getNewMessageTest6	Test that no messages are retrieved when this method is called with no doctor logged in.	26/02/21	null	null	Yes	None
viewBookingTest1	Test that the correct booking is retrieved when logged in as doctor ‘TES1’ assertEquals used to compare strings.	26/02/21	TRUE	FALSE	No	Error in SQL query, missing ‘
viewBookingTest1 - redo	Test that the correct booking is retrieved when	26/02/21	TRUE	TRUE	Yes	None

	logged in as doctor ‘TES1’ assertEquals used to compare strings.					
viewBookingTest2	Test that the correct booking is retrieved when logged in as doctor ‘TES2’ assertEquals used to compare strings.	26/02/21	TRUE	TRUE	Yes	None
viewBookingTest3	Test that the correct booking is retrieved when logged in as doctor ‘TES3’ assertEquals used to compare strings.	26/02/21	TRUE	TRUE	Yes	None
viewBookingTest4	Test that the correct bookings are retrieved when logged in as doctor ‘TES4” assertEquals used to compare linked lists containing multiple bookings.	26/02/21	TRUE	TRUE	Yes	None
viewBookingTest5	Test to ensure no bookings are retrieved when no doctors are logged in assertNull used.	26/02/21	Null	Null	Yes	None