HOSPITAL MANAGEMENT SYSTEM
TEST CASES DOCUMENTATION

In fulfillment of the requirements

For Birmingham City University (BCU) in

BSc. (Hons.) Computer Science with

Artificial Intelligence

Submitted By

Raunak Regmi

raunak.regmi@mail.bcu.ac.uk

ID: 26140514

Overview:

This document contains comprehensive test cases for the Hospital Management System. Each test case includes detailed steps, expected results, actual results with screenshots, and pass/fail status. The system has been tested for all major functionalities including admin authentication, doctor management, patient management, doctor-patient assignment, and reporting features.

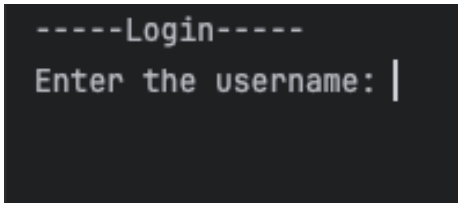
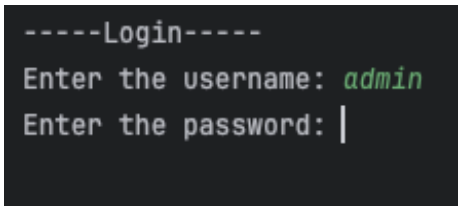
Total Test Cases: 10

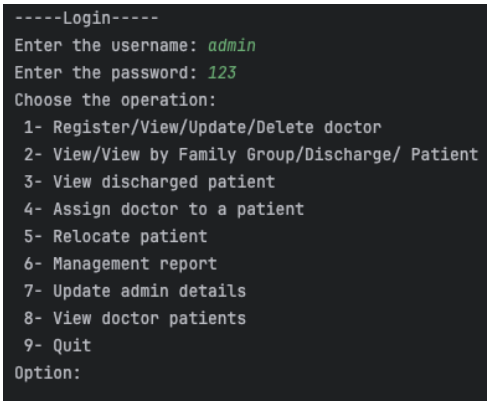
Passed: 9

Failed: 1

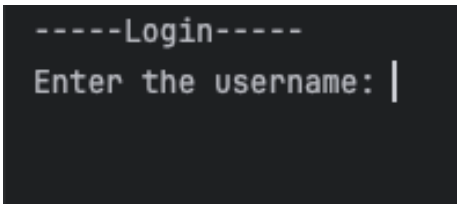
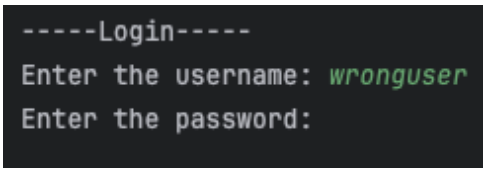
Test Cases:

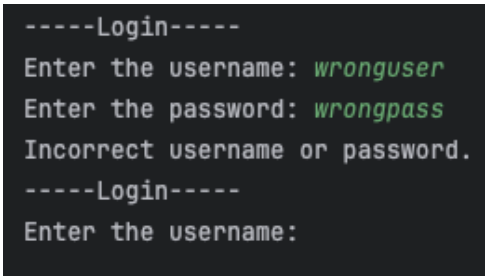
Test Case 1: Successful Admin Login

Test Case ID	Software Feature	Steps	Expected Results	Actual Result	Pass/ Fail
TC-01	Successful Admin Login	1. Launch the Hospital Management System Program by running 'main.py' 2. Wait till the program displays '-----Login-----' 3. Program prompts with 'Enter the username:' and type exactly 'admin' and press Enter 4. Program prompts with 'Enter the password:' and type exactly '123' and press Enter 5. The user successfully accesses	The program starts successfully. The login system is displayed properly. After the user enters username 'admin' the system accepts and asks for password. After the user enters password '123' where the system validates the username and password. The system checks if the username and password match with the one inside "admin.txt" The system then displays the main menu with 9 operations to choose:	 	**Passed**

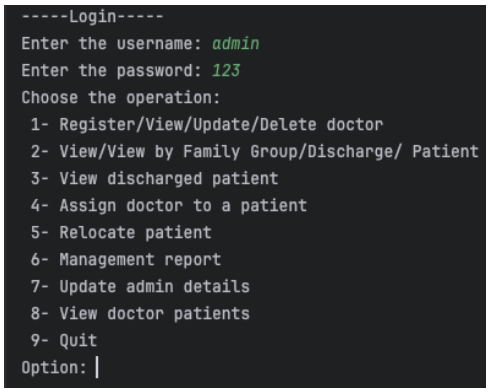
		the menu, and is asked to choose an operation.	1-Register/View/Update/ Delete doctor 2- View/View by Family Group/Discharge/ Patient 3- View discharged patient 4- Assign doctor to a patient 5- Relocate patient 6- Management report 7- Update admin details 8- View doctor patients 9- Quit		

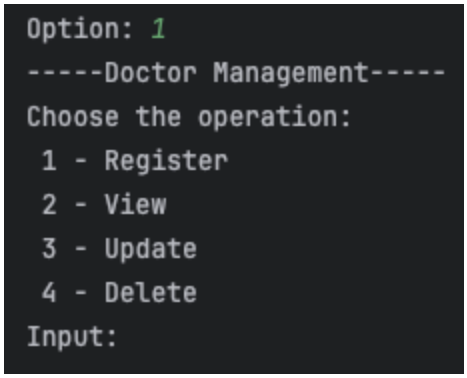
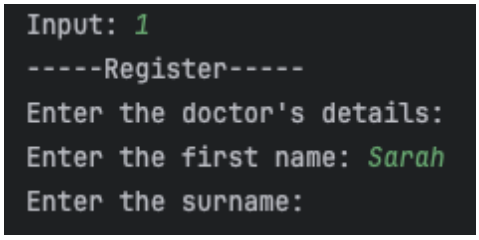
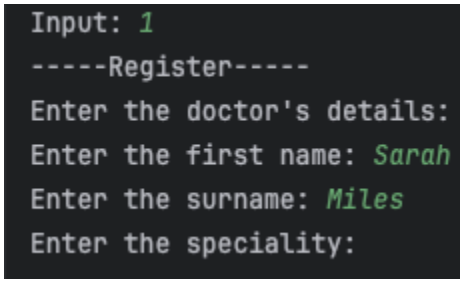
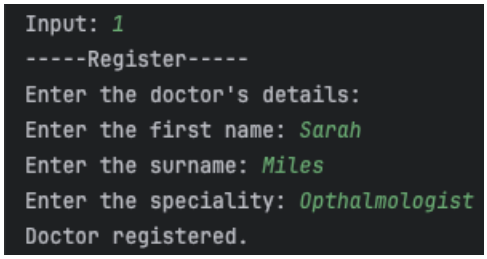
Test Case 2: Failed Admin Login with Incorrect Username and Password

Test Case ID	Software Feature	Steps	Expected Results	Actual Result	Pass/ Fail
TC-02	Failed Admin Login	1. Launch the Hospital Management System Program by running 'main.py' 2. Wait till the program displays '-----Login-----' 3. Program prompts with 'Enter the username:' and type 'wronguser' and press Enter 4. Program prompts with 'Enter the	The program starts successfully. The login system is displayed properly. After the user enters username 'wronguser' the system accepts and asks for password. After the user enters password 'wrongpass' where the system does not validate the username and password with the one inside "admin.txt"	 	**Passed**

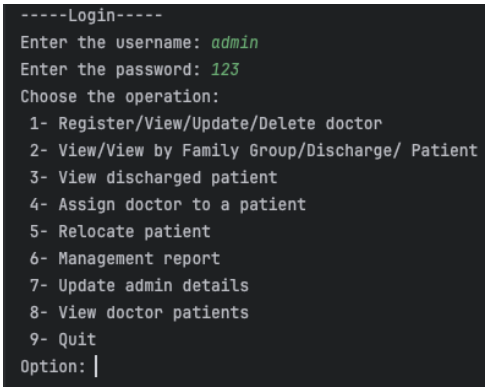
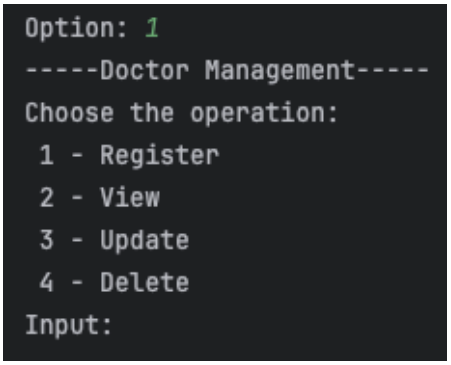
		password:' and type 'wrongpass' and press Enter 5. The system displays a message 'Incorrect username or password.' 6. The program handles the error efficiently. 7. Displays '-----Login-----' and re-prompts 'Enter the username:'	Then the system displays a message 'Incorrect username or password.' The system loops back to the login screen displaying '-----Login-----' Again asks the user for the username and the password. The error handling of the system functions well like it should.		

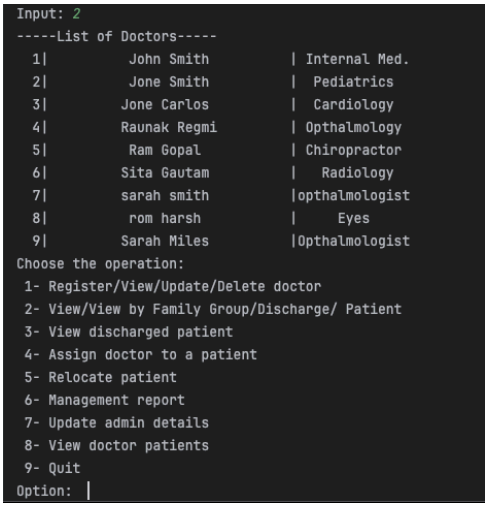
Test Case 3: Register a New Doctor

Test Case ID	Software Feature	Steps	Expected Results	Actual Result	Pass/ Fail
TC-03	Register a New Doctor	1. Successfully log into the system using (username: 'admin', password: '123') 2. From the menu, choose the operation 1, when prompts 'Option:' type '1' 3. Pressing enter provides access to one of the operation: 'Register/View/Update/Delete doctor' 4. The Doctor Management menu	After successful login, the system displays menu 'Choose the operation' with 9 options After typing 1, the system opens new menu of Doctor Management with 4 options: 1 - Register 2 - View 3 - Update 4 - Delete Pressing 1, the system prompts for doctor details beginning with the first name. Then it prompts for the doctor's surname.		**Passed**

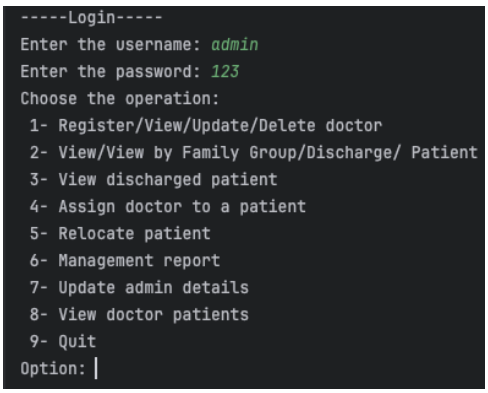
		appears with 4 options and prompting ‘Input:’ 5. Select option 1 by typing ‘1’ and press enter to choose option ‘Register’ 6. When prompted ‘Enter the first name:’, type ‘Sarah’ and press enter. 7. When prompted ‘Enter the surname:’, type ‘Miles’ and press enter. 8. When prompted ‘Enter the speciality’, type ‘Opthalmologist’ and press enter. 9. A confirming message displays ‘Doctor registered.’ which confirms the registration.	And lastly prompts for doctor’s speciality. Before adding, the system checks if a doctor with the same first name and surname exists. Since ‘Sarah Miles’ is a new doctor, the system check is passed. Then, a new object is created by system with the details and new doctor is added to the file ‘ doctors.txt’ The system displays the success message ‘Doctor registered.’	   	

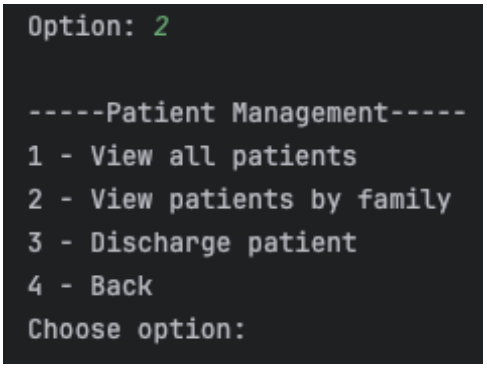
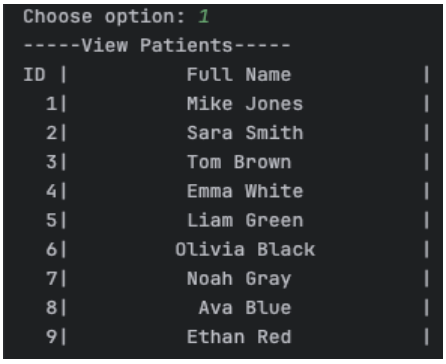
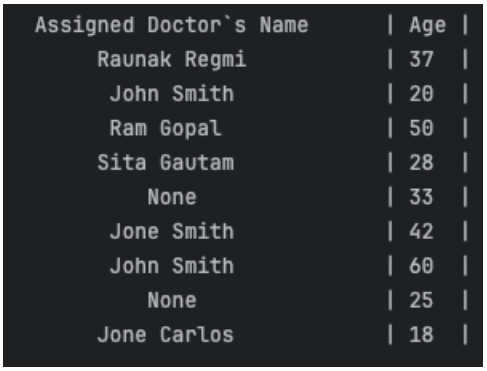
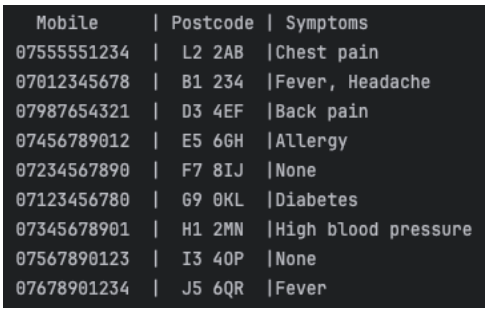
Test Case 4: View All Doctors in the System

Test Case ID	Software Feature	Steps	Expected Results	Actual Result	Pass/ Fail
TC-04	View All Doctors	1. Successfully log into the system using (username: 'admin', password: '123') 2. From the menu, choose the operation 1, when prompts 'Option:' type '1' 3. Pressing enter provides access to one of the operation: 'Register/View/Update/Delete doctor' 4. The Doctor Management menu appears with 4 options and prompting 'Input:' 5. Select option 2 by typing '2' and press enter to choose option 'View' 6. Then it displays the list of all registered doctors in the system.	After successful login, the system displays menu 'Choose the operation' with 9 options After typing 1, the system opens new menu of Doctor Management with 4 options: 1 - Register 2 - View 3 - Update 4 - Delete Pressing 2, the system displays the table header as '-----List of Doctors-----' . Then a formatted table with three columns: - ID: Numbers starting from 1 - Full Name: Registered Doctor's Full Name	 	**Passed**

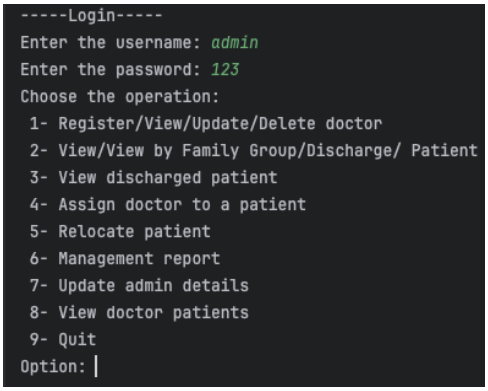
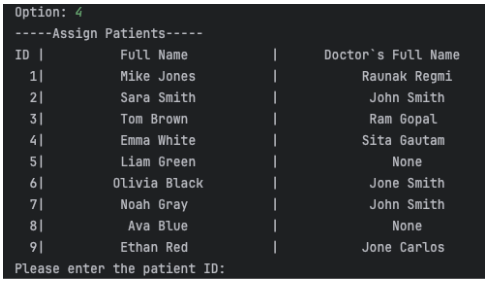
		7. Verify that the display includes doctor's ID number, full name and speciality.	- Speciality: Each doctor's speciality. Appears and displays. New registered doctors are added to the table with their own ID in sequential order to the previous one. All the formatting is done well, with the columns, and proper spacing. After displaying, the system returns back to the Main Menu.	  	

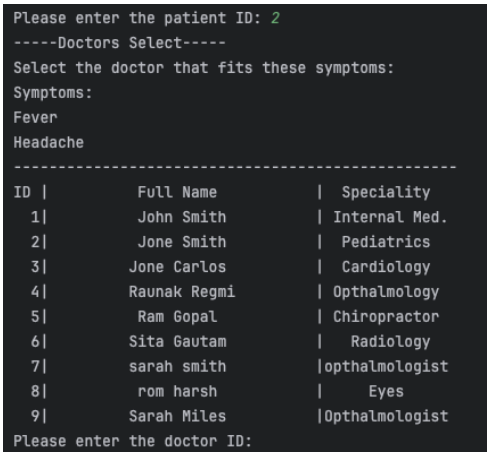
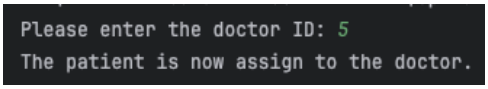
Test Case 5: View All Patients in the System

Test Case ID	Software Feature	Steps	Expected Results	Actual Result	Pass/ Fail
TC-05	View All Patients	1. Ensure the system has patients loaded from 'patients.txt' and doctors from 'doctors.txt' 2. Successfully log into the system using (username: 'admin', password: '123')	During the system loading, the load_patients() function executes. The system reads 'patients.txt' inside data inside of the directory same as 'main.py' After successful login, the system displays menu 'Choose the operation'	 	**Passed**

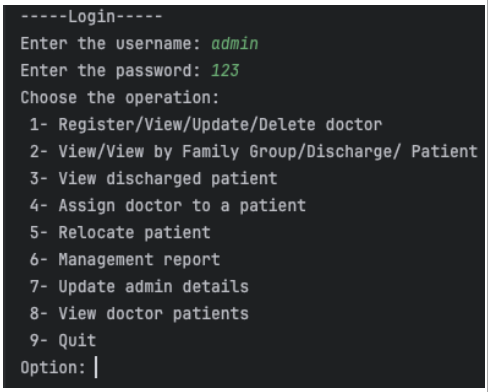
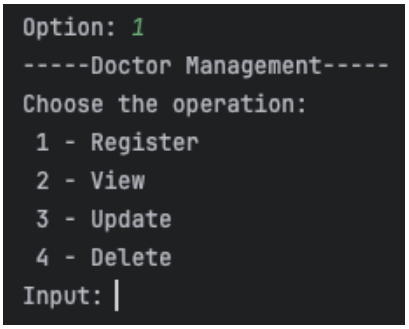
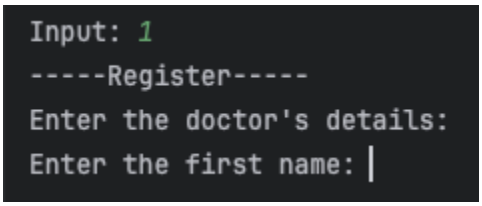
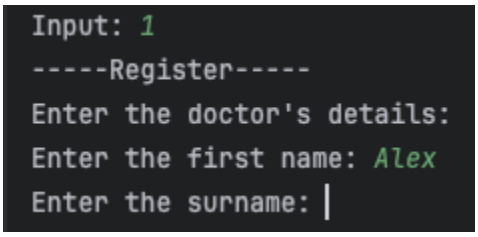
		3. From the menu, choose the operation 2, which when prompts ‘Option:’ type ‘2’ 4. Pressing enter provides access to one of the operation: ‘View/View by Family Group/Discharge/ Patient’ 5. The Patient Management menu appears with 4 options and prompting ‘Choose Option:’ 6. Select option 1 by typing ‘1’ and press enter to choose option ‘View all patients’ 7. Then it displays the list of all admitted patients in the system. 8. Verify that the display includes the patient's ID number, full name, assigned doctor’s name, age, mobile, postcode and symptoms	with 9 options After typing 2, the system opens new menu of Patient Management with 4 options: 1 - View all patients 2 - View patients by family 3 - Discharge patient 4 - Back Pressing 2, the system displays the table header as ‘-----View Patients-----’. Then a formatted table with seven columns with each line linked correctly. Patients are linked to their assigned doctors. (e.g Sara Smith linked to John Smith) Every data (symptoms, mobile, postcode, age) is separated by ‘|’. All the formatting is done well, with the columns, and proper spacing. After displaying, the system returns back to the Patient Management Menu.	   	

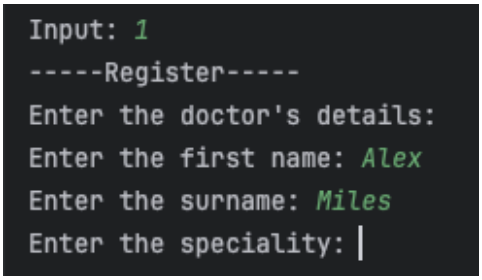
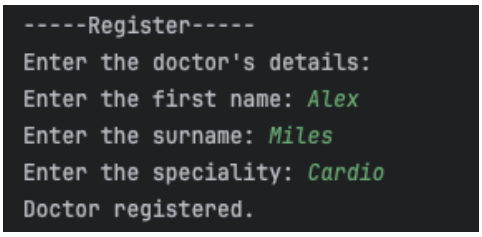
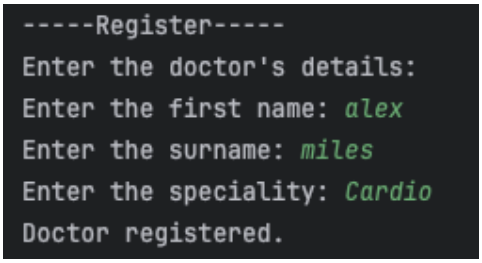
Test Case 6: Assign Doctor To Patient

Test Case ID	Software Feature	Steps	Expected Results	Actual Result	Pass/ Fail
TC-06	Assign Doctor to Patient	1. Ensure the system has patients loaded from ‘patients.txt’ and doctors from ‘doctors.txt’ 2. Successfully log into the system using (username: ‘admin’, password: ‘123’) 3. From the main menu, choose the operation 4 by typing ‘4’ when prompted ‘Option:’ 4. Pressing enter the system displays ‘-----Assign Patients-----’ showing the list of all patients with their IDs, names, current doctors, age, mobile, postcode, and symptoms 5. The system prompts the user ‘Please enter the patient ID:’ 6. Then carefully examine the patients and type an ID number of the patient (e.g 2: Sara Smith)	The system reads ‘patients.txt’ and ‘ doctors.txt’ inside directory data inside of the directory same as ‘main.py’ After successful login, the system displays menu ‘Choose the operation’ with 9 options After typing 4, the system displays the list of patients properly. Patient ID '2' is validated as being in the range [1, len(patients)) for this case Displays message ‘Select the doctor that fits these symptoms:’ after it. Patient ‘Sara Smith’ (for this test case) symptoms are displayed ‘Fever Headache’. Doctors list shows all the available doctors with their specialities. Doctor ID '5' is validated using find_index() method. The patient object (patients[1]) is retrieved successfully. The doctor object (doctors[4]) is retrieved successfully.	 V  	**Passed**

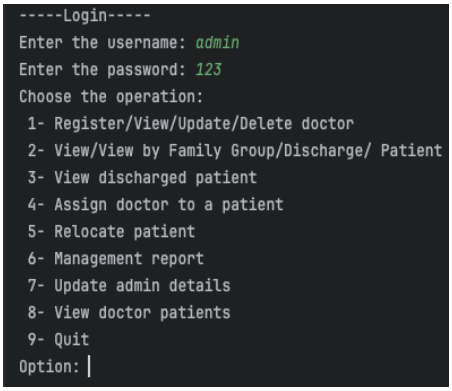
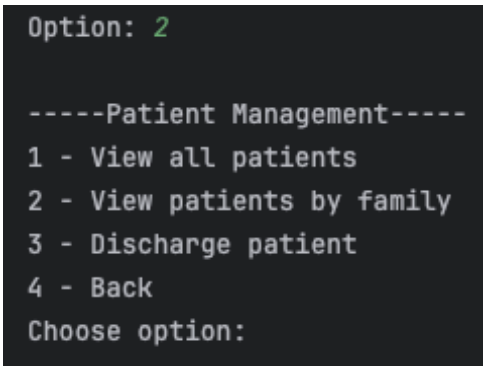
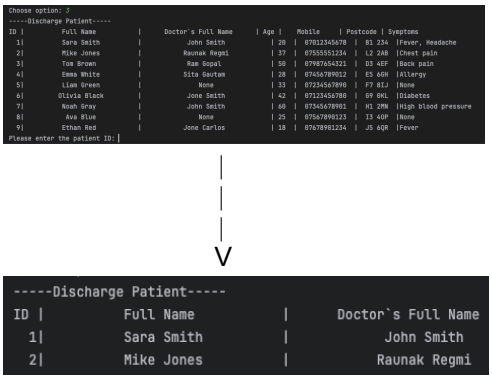
		7. Then it displays the header '-----Doctors Select-----' with instruction 'Select the doctor that fits these symptoms:' 8. Symptoms of the selected patients will be then listed below the message (e.g Symptoms: Fever Headache) 9. The doctor's list is then shown prompting 'Please enter the doctor ID:' to the user 10. Then type doctor's ID to the prompt message (e.g '5') and it displays 'The patient is now assigned to the doctor.' the success message	patient.link(doctor.full_name()) is called, linking Sara Smith to 'Ram Gopal'. doctor.add_patient(patient) is called, adding Sara to Dr. Ram Gopal's patient list. The system displays: 'The patient is now assign to the doctor.' Verification shows Sara Smith's 'Assigned Doctor' field now shows 'Ram Gopal'. The assignment of the doctor stays after system restart (due to file save).	  	

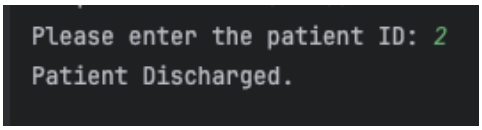
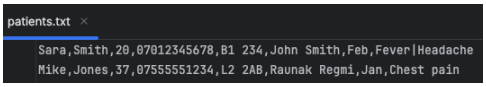
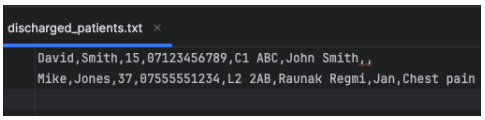
Test Case 7: Case-Sensitive Duplicated Doctor Registration

Test Case ID	Software Feature	Steps	Expected Results	Actual Result	Pass/ Fail
TC-07	Case-Sensitive Duplicate Doctor Registration is not fully Prevented	1. Ensure the system has doctors loaded from doctors from 'doctors.txt' 2. Successfully log into the system using (username: 'admin', password: '123') 3. From the menu, choose the operation 1, when prompts 'Option:' type '1' 4. Pressing enter provides access to one of the operation: 'Register/View/Update/Delete doctor' 5. The Doctor Management menu appears with 4 options and prompting 'Input:' 6. Select option 1 by typing '1' and press enter to choose option 'Register' 7. Then it displays the header '-----Register-----' and 'Enter the doctor's details:' prompting 'Enter the first name: ' 8. During the prompting enter the first name 'Alex.	The system reads ' doctors.txt' inside directory data inside of the directory same as 'main.py' After successful login, the system displays menu 'Choose the operation' with 9 options After typing 1, the system opens new menu of Doctor Management with 4 options: 1 - Register 2 - View 3 - Update 4 - Delete Enter '1' for the Register function into the prompt 'Input:' the system has given. The system displays '------Register-----' header and 'Enter the doctor's details:' message. The system asks for three inputs sequentially after each one is inputted for doctor's detail: - 'Enter the first name: ' - 'Enter the surname: ' - 'Enter the speciality: '	   	**Failed**

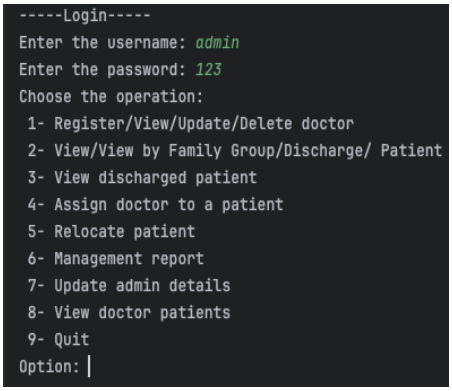
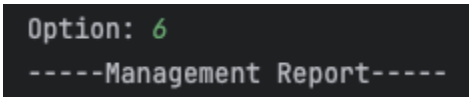
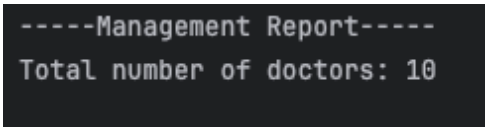
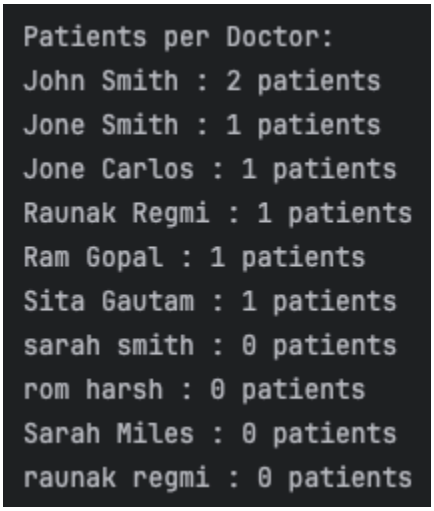
		9. Again, it prompts with ‘Enter the surname: ‘ where you enter ‘Miles’ 10. Then lastly, it prompts with ‘Enter the speciality: ‘ where you enter ‘Cardio’ 11. Then a success message pops up ‘Doctor registered.’ and bring it back to the main menu. 12. Re-do the process till it prompts ‘Enter the first name: ‘ 13. Now, enter ‘alex’ with a small letter ‘A’ 14. Again, when it prompts ‘Enter the surname: ‘ enter ‘miles’ 15. Lastly, when it prompts ‘Enter the speciality: ‘ enter ‘Cardio’ 16. Then a success message pops up ‘Doctor registered.’ and bring it back to the main menu. 17. The system already has doctor ‘Alex Miles’	Registering by giving information ‘Alex Miles, Cardio’ at first. Then doing the same by giving information ‘alex miles, cardio’ after. It should display ‘Doctor already exists’ rather than displaying a success message of ‘Doctor registered.’ The system treats ‘alex miles’ as a new doctor where the doctor is already registered. Creating a duplicate doctor entry in the system. The ‘ doctors.txt’ changes with two same doctors duplicated. You can verify it by Viewing the list of doctors by selecting operation ‘View’ entering ‘2’ inside of Doctor Management.	   	

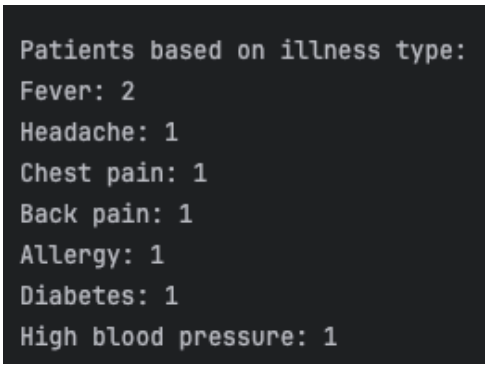
Test Case 8: Discharge a Patient

Test Case ID	Software Feature	Steps	Expected Results	Actual Result	Pass/ Fail
TC-08	Discharge a Patient	1. Ensure the system has patients loaded from ‘patients.txt’. 2. Successfully log into the system using (username: ‘admin’, password: ‘123’) 3. From the menu, choose the operation 2, when prompts ‘Option:’ type ‘2’ 4. Pressing enter provides access to one of the operation: ‘View/View by Family Group/Discharge/ Patient’ 5. The Patient Management menu appears with 4 options and prompting ‘Choose Option:’ 6. Select option 3 by typing ‘3’ and press enter to choose option ‘Discharge patient’ 7. The system displays '-----Discharge Patient-----' header followed by the list of all active patients.	During the system loading, the load_patients() function executes. The system reads ‘patients.txt’ inside data inside of the directory same as ‘main.py’ After successful login, the system displays menu ‘Choose the operation’ with 9 options After typing 2, the system opens new menu of Patient Management with 4 options: 1 - View all patients 2 - View patients by family 3 - Discharge patient 4 - Back Pressing 3, the system displays the table header as ‘-----Discharge Patient-----’. Then All active patients are listed with full details. Patient ID '2' is entered and validated. The system converts input to integer (2) and calculates array index (1). Index validation confirms ID 2 is in valid range.	  	**Passed**

		8. Examine the patients list and identify a patient to discharge. Note their ID number. 9. When prompted 'Please enter the patient ID:', type '2' (e.g discharge Mike Jones) and press Enter. 10. A success message is seen to verify the discharge was successful. 11. Verify by selecting option '1' (View all patients) that Mike Jones is no longer in the active patients list. 12. Return to main menu and select option '3' (View discharged patients) to verify Mike Jones now appears in the discharged patients list inside 'discharged.txt'	The patient object (Mike Jones) is retrieved from patients[1]. The patient is appended to the discharged_patients list. The patient is removed from the active patients list using pop(1). save_patients() is called, updating patients.txt (Mike removed). save_discharged_patients() is called, updating discharged_patients.txt (Mike added). System displays: 'Patient Discharged.' After system restart, Mike Jones remains in discharged_patients.txt and not in patients.txt.	  	

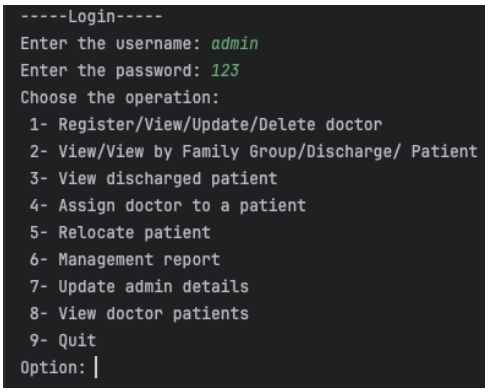
Test Case 9: View Management Report

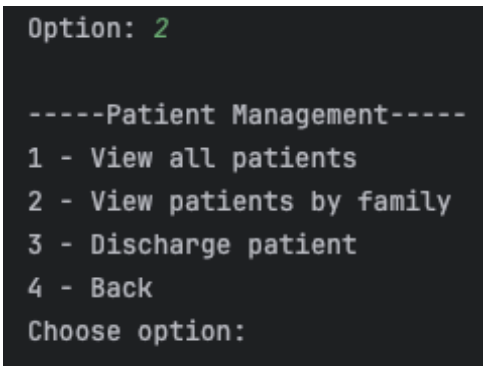
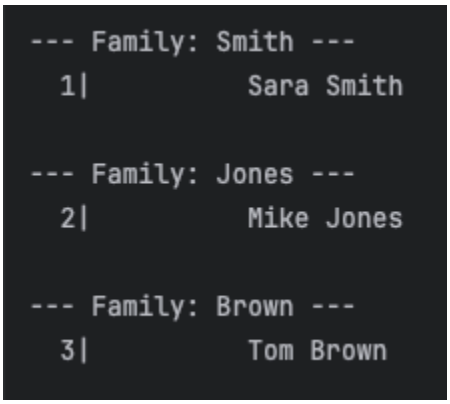
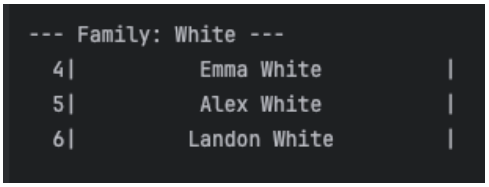
Test Case ID	Software Feature	Steps	Expected Results	Actual Result	Pass/ Fail
TC-09	Management Report Generation	1. Ensure the system has patients loaded from ‘patients.txt’. 2. Successfully log into the system using (username: ‘admin’, password: ‘123’) 3. From the main menu with 9 options, select option '6' (Management report) by typing '6' and pressing Enter. 4. Wait for the system to process and generate the report. 5. Observe and verify the report contains four main sections: - Total number of doctors - Total number of patients per doctor - Appointments per month per doctor - Total patients based on illness type (symptoms) 6. Verify the accuracy of the numbers by cross-referencing with the actual data in the system.	The system reads ‘patients.txt’ inside data inside of the directory same as ‘main.py’ After successful login, the system displays menu ‘Choose the operation’ with 9 options After typing 6, press enter to access the Management Report on option 6. The system displays '-----Management Report-----' header. SECTION 1 - Total Doctors: - System counts the length of the doctors list - Displays: ‘Total number of doctors: ’ SECTION 2 - Patients per Doctor: - Header: 'Patients per Doctor:' - For each doctor in the doctors list: System iterates through all patients Counts how many patients have	   	**Passed**

		7. Note the report formatting and clarity.	- Expected output example: 'John Smith : 2 patients' 'Jone Smith : 0 patients' 'Jone Carlos : 0 patients' SECTION 3 - Appointments per Month per Doctor: - Header: 'Appointments per Month per Doctor:' - System iterates through all patients - Checks which patients are assigned to the current doctor - Retrieves the appointment month for each patient - Groups appointments by month - Counts the number of appointments per month for that doctor - Expected output: 'John Smith: January : 1 appointment March : 1 appointment Jone Smith: No appointments Jone Carlos: February : 2 appointments' SECTION 4 - Patients by Illness: - Header: 'Patients based on illness type:' - System creates illness_count dictionary	 	

			- Iterates through all active patients - For each patient, iterates through their symptoms - Counts occurrences of each symptom - Expected output: 'Fever : 1 patients' 'Headache : 1 patients' - If no symptoms exist, displays: 'No illness data available.' Report is clearly formatted and easy to read. After displaying the report, the system returns to the main menu.		

Test Case 10: View Patients by Family Group

Test Case ID	Software Feature	Steps	Expected Results	Actual Result	Pass/ Fail
TC-10	View Patients Grouped by Family Surname	1. Ensure the system has patients loaded from ‘patients.txt’ and doctors from ‘doctors.txt’ 2. Successfully log into the system using (username: ‘admin’, password: ‘123’) 3. From the menu, choose the operation 2, which when	During the system loading, the load_patients() function executes. The system reads ‘patients.txt’ inside data inside of the directory same as ‘main.py’ After successful login, the system displays menu ‘Choose the operation’ with 9 options		**Passed**

		prompts ‘Option:’ type ‘2’ 4. Pressing enter provides access to one of the operation: ‘View/View by Family Group/Discharge/ Patient’ 5. The Patient Management menu appears with 4 options and prompting ‘Choose Option:’ 6. Select option 2 by typing ‘2’ and press enter to choose option ‘View Patients by Family’ 7. Then it displays the list of all patients in the system grouped by Family 8. Verify that patients are grouped by their surname with clear family headers. 9. Check that the formatting is clear and family groups are seen.	After typing 2, the system opens new menu of Patient Management with 4 options: 1 - View all patients 2 - View patients by family 3 - Discharge patient 4 - Back Pressing 2, the system displays the table header as ‘-----View Patients (Grouped by Family)-----’. System creates empty dictionary: family_groups = {} Iterates through all patients For each patient, extracts surname using patient.get_surname() Groups patients by surname as dictionary keys Each dictionary value is a list of patient objects with that surname For the display logic, the system maintains the id_counter starting from 1 for the first patient. For each family surname, the system prints header ‘--- Family: Smith ---’ Inside the header, the system lists all the patients in that family with a unique sequence of ID. After displaying, the system returns back to the Patient Management Menu.	  	

Summary:

Total cases: 10

Passed: 9

Failed: 1

TC-01: Successful Admin Login
TC-02: Failed Login with Incorrect Credentials
TC-03: Register a New Doctor
TC-04: View All Doctors
TC-05: View All Patients (File Loading)
TC-06: Assign Doctor to Patient
TC-07: Duplicated Doctor Registration
TC-08: Discharge a Patient
TC-09: Management Report Generation
TC-010: View Patients by Family Group

Conclusion

The Hospital Management System demonstrates strong functionality across most features with a 90% test pass rate. Core features including authentication, data management, file persistence, and reporting all work as expected. The system successfully handles valid inputs and most error conditions gracefully. The identified failure highlights an area for future improvement involving case-insensitive validation to prevent duplicate doctor records.