Test Cases

Black-Box Test Cases

These test cases are defined based on the available documentation and execution of the program. The code was not inspected.

#	Test case (very brief description)	Preconditions (any required setup)	Test steps (steps executed during testing)	Expectation	Observation ("pass" or failure description)
1	Verifying that the registered companies list is displayed correctly	User is logged in as a member.	● Logging in ● Navigate to the registered companies list page	The page displays a table with firm names, email addresses, and landline numbers	pass Now asks for login authentication
2	Verify that a user can propose three company names for reservation	-	● Navigate to Company Registration ● Enter three different proposed company names. ● Fill in other required details (address, contact person, contact details).	The system should accept the three proposed names and display a confirmation message indicating that the names are being processed	Pass
3	Verifying the confirmation and payment process for company incorporation	User has proposed three company names.	● Select an available company name. ● Pay the incorporation fee (e.g., €40) using one of the provided payment methods (bank transfer, PayPal, Stripe, etc.)	The system should confirm the payment and display a message that the registration is successful and sent for approval	Pass
4	Verifying that the user is getting by email unique login credentials generated by the system	The system should generate unique login credentials	● After registration checking the email for credentials	Getting the login credentials from email	#updated, not possible to do it. Need server and will have to change the complete code and its structure.

5	Verifying that a registered member can log in to the member portal	Member login with valid credentials	● Navigate to the member login page. ● Enter the username and password.	The member should be successfully logged in and redirected to the member dashboard.	Pass
6	Verifying that a member can access the news forum	Member is logged in	● Navigate to the news forum section from the member dashboard	The news forum should display the latest updates, announcements, and industry news.	Pass
7	Verifying that a member can submit a query	Member is logged in	● Navigate to member dashboard Fill and Submit the Query	The system should display and message that the query is sent	Pass
8	Verifying that a member can view their previous queries	Member is logged in and has submitted at least one query	● Navigate to the "My Queries" section	The page should display a list of previous queries with their status and answers	Pass
9	Verifying that a member can provide feedback on a resolved query	Member is logged in and has a resolved query	● Navigate to the "View Previous Queries" section. ● Select a resolved query. ● Provide feedback by rating satisfaction on a scale of 1 to 3. ● Submit the feedback	The feedback should be successfully submitted and saved in the system	Failed, Still not able to fix as the way the application is created it is too complicated.
10	Verifying that an employee can sign up to the employee portal.	The employee registration feature must be available and accessible on the application and must have a valid employee ID	● Navigate to the employee sign-up page. ● Enter the employee ID and other details	▯ The system should validate the provided employee ID against the database to ensure it is valid. ▯ The employee should be able to log in with the newly created account credentials.	pass Now able to login using new account after account creation

11	Verifying that an employee can manage member queries	Employee is logged in and has appropriate access rights	● Navigate to the query management section. ● View a list of submitted queries. ● Select a query to address. ● Provide a response or solution and update the query status.	The query status should be updated, and the member should be notified of the resolution.	Pass
12	Verifying that an employee can create news for the news forum	Employee is logged in and has appropriate access rights	● Navigate to the news creation section. ● Enter news details, including title and content. ● Submit the news article.	The news article should be successfully created and visible in the news forum for members	Pass
13	Verifying that role-based access control	Employee accounts with different roles exist (e.g., CEO, lower-level staff)	● Log in as a lower-level staff member.	Lower-level staff should be restricted from accessing certain areas, while the CEO should have full access.	Fail need to change the whole code from the beginning.
14	Verifying that employees can approve company registration requests	Employee login with valid credentials; Pending company registration requests exist	● Navigate to the pending registration requests section. ● Approve a request	The selected company registration request should be approved successfully	Pass
15	Verifying that an invalid member login attempt is handled correctly	-	● Navigate to the member login page. ● Enter invalid credentials (wrong username or password)	The system should display an error message indicating invalid login credentials	Pass
16	Verifying that an invalid employee login attempt is handled correctly	-	● Navigate to the employee login page. ● Enter invalid credentials (wrong employee ID or password)	The system should display an error message indicating invalid login credentials	Pass
17	Verifying that the system prevents duplicate employee registrations	An employee is already registered with a specific Employee ID	● Navigate to the employee registration page.	The system should display an error message indicating	Pass Duplicate employee names not coming now

			● Enter the already registered Employee ID and other required details.	that the Employee ID is already registered	Employee already exists error is thrown . And can’t register with an empty string.
18	Verifying that members can update their profile information	Member login with valid credentials	● Navigate to the member profile page. ● Update profile details (e.g., email, phone number)	The system should successfully save the updated profile information	Pass Now we can update profile.
19	Verifying that a member can successfully submit a review for an employee or service	Member is logged in to the portal	● Navigate to the review section in the member dashboard. ● Select the interaction or employee to review from the list (if applicable). ● Provide a rating (e.g., on a scale of 1 to 5) and write a review in the provided text box	The system should validate the review input (e.g., rating is within the acceptable range, text box is not empty). Upon successful validation, the review should be saved in the database. The member should see a confirmation message indicating the review has been submitted successfully. The review should be visible in the member's review history and accessible to relevant employees	Pass #update, members can review and feedback for a specific employee, and employees can see it.

White-Box Test Cases

These additional test cases were defined during inspection of the code.

#	Test case (very brief description)	Preconditions (any required setup)	Test steps (steps executed during testing)	Expectation	Observation ("pass" or failure description)
1	Test if feedback is correctly written to CSV. AddMemberFeedback Test Class	Test CSV file is created and empty	• Call AddMemberFeedback.writeToCSV with comment "This is a test feedback." and user "testuser". • Read the CSV file. • 3Verify the comment and user in the CSV file.	The comment and user should be present in the CSV file.	Pass
2	Test if empty comment is not written to CSV. AddMemberFeedback Test Class	Test CSV file is created and empty	• Call AddMemberFeedback.writeToCSV with empty comment and user "testuser". • Read the CSV file. • Verify that no data is written to the CSV file.	No data should be present in the CSV file for the empty comment	Pass Comments field cannot be empty
3	Test if news is successfully added. AddNewsTest Class	News file exists and is accessible	• Call AddNews.addNews with news "Test news". • Verify the method returns true. • Read the news file. • Verify the news "Test news" is present in the file.	The news should be added successfully and present in the file.	Pass
4	Test if empty news is not added. AddNewsTest Class	News file exists and is accessible	• Call AddNews.addNews with empty news. • Verify the method returns false.	The empty news should not be added and the method should return false.	Pass Empty news cannot be submitted
5	Test if queries are correctly fetched from the CSV file.	CSV file exists with	• Call AnswerQuery.fetchQueries to read the queries from the CSV file. • Verify that three queries are read.	The CSV file should be read correctly, and the	Pass

	AnswerQueryTest Class	predefined test data	● Check the details (question, answer, user, answered status) of each query to ensure they match the expected values.	queries should match the expected data.	
6	Test if a line in the CSV file is correctly replaced. AnswerQueryTest Class	CSV file exists with predefined test data	● Call AnswerQuery.replaceLineInCSV to update the second query with a new answer. ● Call AnswerQuery.fetchQueries to read the queries from the CSV file. ● Verify the second query's details to ensure the answer is updated. ● Verify the other queries remain unchanged.	The second query should be updated with the new answer, and other queries should remain unchanged	Pass
7	Test if the answer for a selected question is correctly set. AnswerQueryTest Class	CSV file exists with predefined test data	● Set AnswerQuery.selectedIndex to 0 (first question). ● Call AnswerQuery.setAnswerForQuestion to set a new answer for the first question. Call AnswerQuery.fetchQueries to read the queries from the CSV file. ● Verify the first query's details to ensure the answer is updated. ● Verify the other queries remain unchanged.	The first query should be updated with the new answer, and other queries should remain unchanged.	Pass
8	Test setting an empty answer for a question AnswerQueryTest Class	CSV file exists with predefined test data	● Set AnswerQuery.selectedIndex to 1 (second question). ● Call AnswerQuery.setAnswerForQuestion to set an empty answer for the second question. ● Call AnswerQuery.fetchQueries to read the queries from the CSV file. ● Verify the second query's details to ensure the answer is updated but not empty. ● Verify the other queries remain unchanged.	The second query should have a non-empty answer, and other queries should remain unchanged.	Pass No empty strings are being stored and it throws an exception
9	Test getters and setters of Company class. CompanyTest Class	A Company object is created with initial values	● Create a Company object with initial values ● Verify initial values using getter methods. ● Update values using setter methods. ● Verify updated values using getter methods.	Getter methods should return initial values correctly, and updated	Pass

				values after using setter methods.	
10	Test Company object with empty string values. CompanyTest Class	A Company object is created with empty string values	● Create a Company object with empty string values ● Verify that getter methods return empty strings. ● Ensure test fails because empty string values are not desirable.	Getter methods should return empty strings, and the test should fail because empty values are not allowed.	pass Empty values are not being saved in the csv
11	Test if valid employee credentials are accepted. successfully EmployeeLoginTest Class	Valid loginId and password exist in the system	● Set loginId and password to valid data. ● Call EmployeeLogin.validateEmployee with these credentials. ● Verify the method returns true	The method should return true indicating the credentials are valid.	Pass
12	Test if invalid employee credentials are rejected. EmployeeLoginTest Class	Invalid loginId and password are provided	● Set loginId to "unknown_user" and password to "wrong_password". ● Call EmployeeLogin.validateEmployee with these credentials. ● Verify the method returns false.	The method should return false indicating the credentials are invalid.	Pass
13	Test if valid members are validated correctly MemberLoginTest Class	Valid loginId and password	● Set loginID and password to valid data. ● Calling memberLogin.validateMember ● Verify method returns True	The method should return true indicating the credentials are valid.	Pass
14	Test if invalid members are not validated MemberLoginTest Class	Invalid loginId and pass	● Set loginID and password to invalid data. ● Calling memberLogin.validateMember ● Verify method returns False	The method should return False indicating the credentials are invalid.	Pass
15	Test if Empty/Null Credentials are not valid MemberLoginTest Class	Empty/Null loginId and pass	● 1.Set loginID and password to empty/null. ● 2.Calling memberLogin.validateMember ● 3.Verify methods returns False	The method should return False indicating the credentials are invalid.	Pass

16	Test searching queries by valid user ID	Valid userID	- Call MyQueries.searchQueriesByUserId with a valid user ID.	The method should return a non-null list containing at least one Queries object with the matching user ID.	pass
17	Test searching queries by invalid user ID	Invalid UserID	- Call MyQueries.searchQueriesByUserId with an invalid user ID.	The method should return a non-null empty list.	Pass
18	Test successful company registration on payment success PaymentTest Class	Valid company information and successful payment	- Create a company object with valid information. - Simulate a successful payment. - 3. Call Payment.registerCompany with the company object.	The registerCompany method should return true.	Pass
19	Test unsuccessful company registration on payment failure PaymentTest Class	Valid company information and failed payment	- Create a company object with valid information. - Simulate a wrong payment value. - 3. Call Payment.registerCompany with the company object.	The registerCompany method should return false.	pass
20	Test unsuccessful company registration with missing information PaymentTest Class	Company information with missing fields and successful payment	- Create a company object with missing information (e.g., empty username). - Simulate a successful payment. - 3. Call Payment.registerCompany with the company object.	The registerCompany method should return false.	pass
21	Test getters and setters of Queries object QueriesTest Class	A Queries object is created with initial values.	- Call getters on the object. - Modify the object's properties using setters (setQuestion, setAnswer, setUserId, setAnswered). - Call getters again to verify the changes.	Getters should return the initial values. Setters should modify the object's properties, and subsequent calls to getters should reflect the changes.	Pass

22	Test validation of empty strings in Queries object QueriesTest Class	A Queries object is created with initial values.	● Attempt to set the question and answer of the object to empty strings using setters.	The setters should throw an IllegalArgumentException.	pass
23	Test valid query validation QueryAndSupportTest Class	-	● Call QueryAndSupport.isValidQuery with a valid query string (e.g., "This is a valid question").	The method should return true.	Failure (The method return false indicating that the method accepts queries in another format)
24	Test invalid query validation (comma) QueryAndSupportTest Class	-	Call QueryAndSupport.isValidQuery with a query string containing a comma (e.g., "Invalid query, contains comma").	The method should return false.	Pass
26	Test empty query validation QueryAndSupportTest Class	-	Call QueryAndSupport.isValidQuery with an empty string.	The method should return false.	pass
27	Test successful employee credential saving RegisterEmployeeGUITest Class	A test CSV file exists.	● Call RegisterEmployeeGUI.saveEmployeeCredentials with valid user information. ● Read the contents of the test CSV file.	The method should write the user information to the CSV file.	pass
28	Test saving employee credentials with missing fields RegisterEmployeeGUITest Class	A test CSV file exists.	● Call RegisterEmployeeGUI.saveEmployeeCredentials with some missing fields (e.g., empty password).	The method should throw an IllegalArgumentException.	pass
29	Test showing an alert UtilClassTest Class	JavaFX is initialized.	● Define alert properties. ● Call showAlert on the JavaFX application thread.	The retrieved alert should match the defined properties.	Pass

			● Retrieve the last shown alert.		