|  |  |
| --- | --- |
|  |  |
| **Group 9A**  **Element 3**  **Testing** |  |
|  |  |
|  | UFCF8S-30-2  Advanced Software Development  Horizon Cinemas Booking System |
|  | Amira Soumid  Abdelmadjid Lachichi  Nathan Baker  Terence Bakhda |

**Element 3 (Testing- 10 marks)**

A series of manual test cases were designed and executed to validate the booking system. These test cases simulate realistic user interactions across core modules such as login, film listings, booking, cancellation, cinema management, and reporting. Each case outlines specific inputs and expected outcomes, including both valid and invalid scenarios. This structured testing approach verified functional correctness, input validation, and user feedback mechanisms, contributing to overall system robustness.

**Login:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case #** | **Description** | **Test Input** | **Expected Output** | **Actual Output** |
| 1 | Valid login | Correct username and password | User redirected to dashboard | Pass – Output as expected |
| 2 | Invalid login (wrong password) | Wrong password | Login failure message | Pass – Output as expected |
| 3 | Invalid login (wrong username) | Wrong username | Login failure message | Pass – Output as expected |
| 4 | Empty login fields | No input | Warning message for missing data | Pass – Output as expected |

**Film Listings:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case #** | **Description** | **Test Input** | **Expected Output** | **Actual Output** |
| 5 | Cinema selection to view show listings | Select cinema in dropdown menu | Show date, time, cost and location listed for all films | Pass – Output as expected |
| 6 | Show films 1 week in advance only | Select ‘next day’ until a week has past | Warning message for no show listings more than 1 week in advance | Pass – Output as expected |

**Create Booking:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 7 | Check ticket price and availability | Valid selection and ticket type | Total and available seats shown | Pass – Output as expected |
| 8 | Book ticket successfully | Valid show, user details, quantity | Booking confirmed with receipt | Pass – Output as expected |
| 9 | Invalid email booking attempt | Invalid email format | Error message shown | Pass – Output as expected |
| 10 | Booking with invalid quantity | 0 or negative number | Error message shown | Pass – Output as expected |
| 11 | Duplicate booking | Same email and show | Warning of duplicate booking | Pass – Output as expected |

**Cancel Booking:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 12 | Cancel by reference | Valid booking reference | Booking shown and cancelled | Pass – Output as expected |
| 13 | Cancel by email | Valid email | All user bookings listed | Pass – Output as expected |
| 14 | Cancel past show | Show that’s already passed | Cancellation blocked | Pass – Output as expected |

**Manage Cinemas:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 15 | Add cinema successfully | Valid details for cinema, screens, capacity | Cinema added + shows auto-created | Pass – Output as expected |
| 16 | Add cinema with missing data | Omit some required fields | Error message shown | Pass – Output as expected |
| 17 | Delete a cinema | Select from list and confirm | Cinema and all data deleted | Pass – Output as expected |

**Manage Film Listings:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 18 | Add film with all data | Title, genre, rating, actors, description, poster | Film added and listed | Pass – Output as expected |
| 19 | Update film info | Modify existing film and save | Changes reflected | Pass – Output as expected |
| 20 | Delete film | Select from list and delete | Film removed | Pass – Output as expected |

**Generate Report:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 21 | Export reports as CSV | Click export | CSV file saved | Pass – Output as expected |
| 22 | Export reports as PDF | Click export | PDF file saved | Pass – Output as expected |

**Unit Testing:**

|  |
| --- |
| To ensure the reliability and correctness of the Horizon Cinema system, unit testing was implemented using Python’s unittest framework along with unittest.mock for simulating GUI components and database interactions. A total of 13 targeted tests were created, covering key areas such as booking functionality (including price calculation, input validation, and seat availability), login validation, cancellation processes, film and cinema management, and reporting features. An in-memory SQLite database was used for isolated and repeatable test environments. This approach helped verify core logic without relying on the actual UI, resulting in efficient automated validation of system behavior. |

A computer screen shot of a program

AI-generated content may be incorrect.