|  |  |
| --- | --- |
| Ticket reservation for on campus events Description: Online student campus activities and ticket reservation website. | Group 5 Akshay Bhat Giridhar Chalumuri Shivangi Gaur Aditi Nitin Ghamandi Dedipya Jain |

**Table of Contents**

1. **Project Description**
   1. **Scope**
2. **Requirements**
3. **Domain Model**
4. **Use Case Model**

**4.1. Admin UCD**

**4.2. Student UCD**

**4.3. Expanded Use Case**

**4.4. Traceability Matrix**

**4.5. Successful/Unsuccessful Scenarios**

**5. Sequence Diagram**

**5.1. Admin Add Event**

**5.2. Admin Edit Event**

**5.3. Admin Cancel Event**

**5.4. Student Login**

**5.5. Search Event**

**5.6. Manage Cart**

**6. Class Diagram**

**7. Screenshots**

**8. Testing**

**9. Code**

1. **Project Description**

To design and create a centralized platform to provide students access to all the events happening on campus and book advanced tickets for those events.

There are lots of events going on in the campus. To keep track of these events is a big workload. Our event system keeps track of these events. Moreover, many events have limited seats and our ticket reservation will help the student to pick a spot for the event. Many a times, if it’s a new student, they have very limited resources and are particularly clueless about what’s happening in the college. These events may be sporting or educational. This system also helps the staff and students to reduce the hassle during the event while buying tickets. The student can view all events happening like sports, educational lectures, movies, fun activities near library among others. They can see its timings and book seats according to their liking. We are helping to bridge the gap between the college events and students. The application provides a good opportunity to students to keep track of the events happing on campus and RSVP for them. It also provides opportunity to all the event managers to reach out to the students.

* 1. **Scope**

The application provides a good opportunity to students to keep track of the events happing on campus and RSVP for them. It also provides opportunity to all the event managers to reach out to the students.

1. **Requirements**

R1- The system must provide students/ administrator to login into an account.

R2- System must display all the current on-campus events.

R3- System must provide search based on different criteria (event type, date, price, location) and display search results based on selected criteria.

R4-System must provide the details of selected event.

R5-System must provide a shopping cart to add multiple event tickets and check them out.

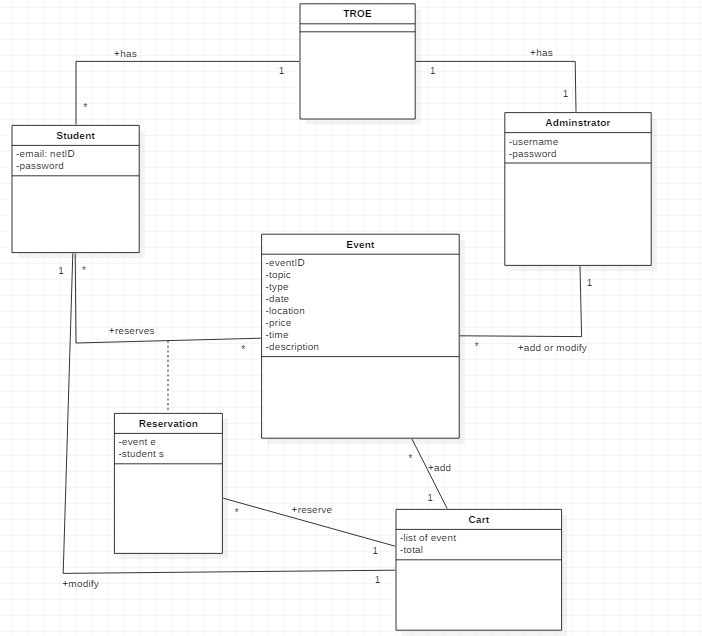
R6-System must generate a receipt for successful checkout.

R7-System must generate an email confirmation to the student.

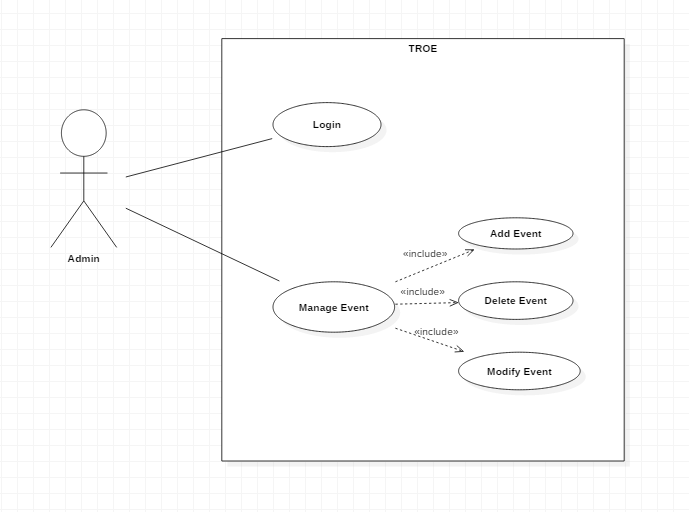
R8-System must provide students to modify their existing reservation.

R9-System must provide a way to administrator to add and modify events.

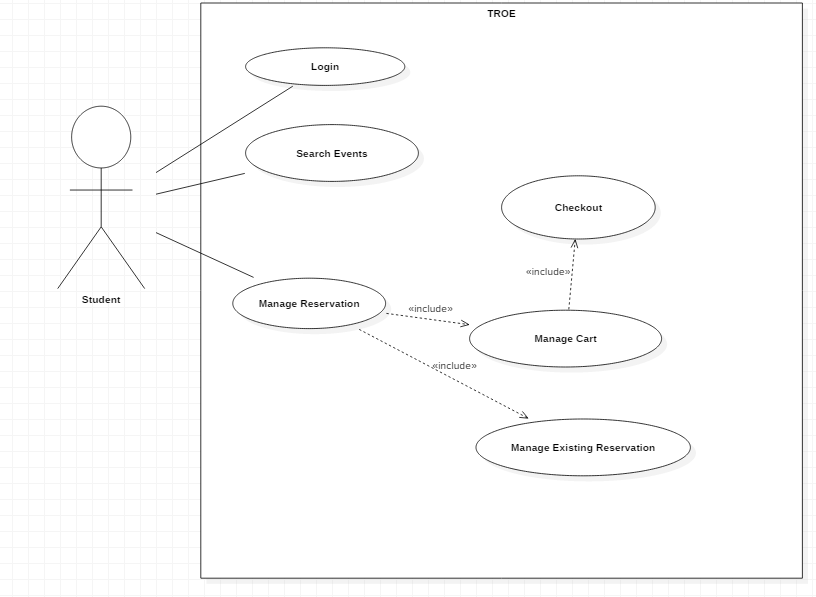
1. **Domain Model**



1. **Use Case Model**
   1. **Admin UCD**



* 1. **Student UCD**



* 1. **Expanded Use Cases**

UC1: Login

|  |  |
| --- | --- |
| Precondition: System should be up and running | |
| Actor: Student/Administrator | System: TROE |
| 1. TUCBW, User will enter username and password. User will click on login button. | 1. System will authenticate the credentials. 2. Valid credentials. 3. Invalid credentials. |
| 1. a. TUCEW, User will see the welcome page   b. TUCEW, Ask the user to enter credentials again. |  |

UC2: Manage Event – Add Event

|  |  |
| --- | --- |
| Precondition: Administrator should be logged in. | |
| Actor: Administrator | System: TROE |
| 1. TUCBW, Administrator will click on add events button. | 1. System will display a form to add events. |
| 1. Administrator will fill the form to add the event. | 1. a. System will add the event to the database.   b. System will show event cannot be added because of the validations. |
| 1. a. TUCEW, Administrator will see that event is added successfully.   b. TUCEW, Administrator will fill the form again and re-submit. |  |

UC2: Manage Event – Edit Event

|  |  |
| --- | --- |
| Precondition: Administrator should be logged in. | |
| Actor: Administrator | System: TROE |
| 1. TUCBW, Administrator will click on edit events button. | 1. System will display a textbox and ask the Administrator to enter the Event-Id to be edited. |
| 1. Administrator will enter the Event-Id of the event to be edited. | 1. a. System will display an already filled form to be edited.   b. System will show that there is no event related to that Event-Id and ask the Administrator to enter it again. |
| 1. a. Administrator will edit the form and submit.   b. Administrator will enter the correct Event-Id again. | 1. System will add the event with modifications in the database. |
| 1. TUCEW, Administrator will see that event is edited successfully. |  |

UC2: Manage Event – Cancel Event

|  |  |
| --- | --- |
| Precondition: Administrator should be logged in. | |
| Actor: Administrator | System: TROE |
| 1. TUCBW, Administrator will click on cancel events button. | 1. System will display a textbox and ask the Administrator to enter the Event-Id to be cancelled. |
| 1. Administrator will enter the Event-Id of the event to be cancelled and click on the delete button. | 1. a. System will delete the event from the database.   b. System will show that there is no event related to that Event-Id and ask the Administrator to enter it again. |
| 1. TUCEW, Administrator will see that event is deleted successfully. |  |

UC4: Search for events

|  |  |
| --- | --- |
| Precondition: System should have list of current events | |
| Actor: Student | System: TROE |
| 1. TUCBW, Student will search program based on different criteria (event type, date, price) by selecting given checkboxes. Student will click on search button. | 1. a. System will display all the programs according to search criteria.   b. The search results are empty. |
| 1. a. TUCEW, Student can see the desired events.   b. TUCEW, Display message “no events available”. |  |

UC5: Manage Cart

|  |  |
| --- | --- |
| Precondition: Student should be logged in and a list of current events should be displayed by the system. | |
| Actor: Student | System: TROE |
| 1. TUCBW, Student will select desired event to open for more details. | 1. a. System will display the details of the event selected.   b. System will display that event is no longer available. |
| 1. a. Student can add the selected event into the cart.   b. Student can go back to the current list to select a different event or to add more events into the cart | 1. a. System will add the events into the cart.   b. If the event is no longer available or if a student is already added that event then a message “event already added” will be displayed. |
| 1. a. Student can open the cart and press the checkout button.   b. Student can open the cart, delete any event from there and then press checkout. | 1. a. System will display a thank you page stating you have registered for selected events.   b. If the cart is empty then system will display an error message and ask student to reserve for the even again. |
| 1. a. TUCEW, Student will receive a confirmation message.   b. TUCEW, Display message “you are not registered try again”. |  |

UC5: Manage Existing Reservation

|  |  |
| --- | --- |
| Precondition: Student should be logged in and at least registered in one event | |
| Actor: Student | System: TROE |
| 1. TUCBW, Student will open the My Reservations tab and select the event to be deleted. | 1. a. System will display all the rest events.   b. The list of registered events is empty. |
| 1. a. TUCEW, Student can see that he is no longer registered for that event.   b. TUCEW, display no events registered. |  |

**4.4. Traceability Matrix**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Priority Weight | UC1 | UC2 | UC3 | UC4 | UC5 |
| R1 | 3 | X |  | X |  |  |
| R2 | 2 |  |  |  | X |  |
| R3 | 2 |  |  |  | X |  |
| R4 | 2 |  |  |  | X |  |
| R5 | 2 |  |  |  |  | X |
| R6 | 2 |  |  |  |  | X |
| R7 | 2 |  |  |  |  | X |
| R8 | 1 |  |  |  |  | X |
| R9 | 1 |  | X |  |  |  |
| Score | | 3 | 1 | 3 | 6 | 7 |

**4.5.** **Successful/Unsuccessful Scenarios**

1. The student enters his/her login credentials.

2. Validate the student.

2.1 Invalid User

Ask the user to enter again.

3. The system will display list of current on-campus events.

4. Student will search the events based on different criteria.

4.1 The search results are empty.

5. Student selects the event.

6. The system will display the details of the event.

7. Student will add the event in the cart.

8. Student will checkout his cart.

8.1 The event is full, and no seats are left.

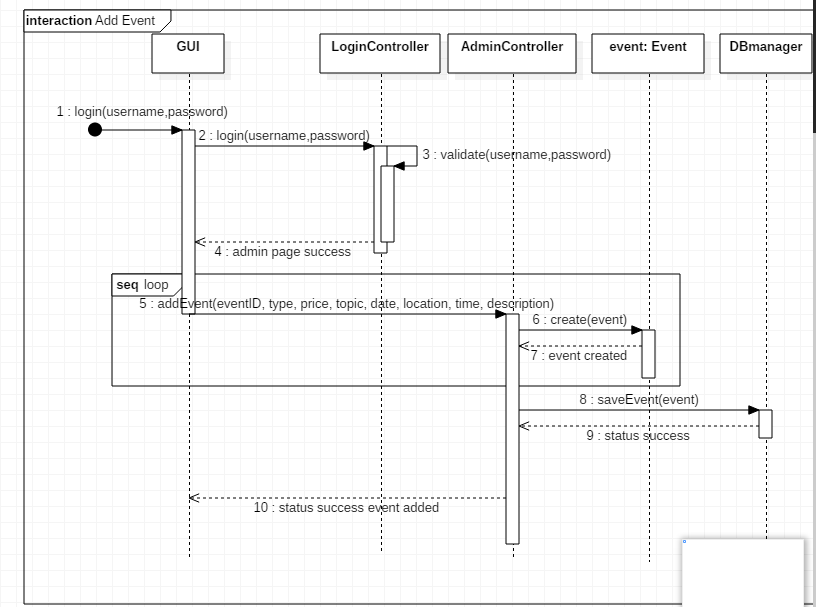
Display the event is full and the student cannot register for the event.

9. Receipt will be generated for the registered events.

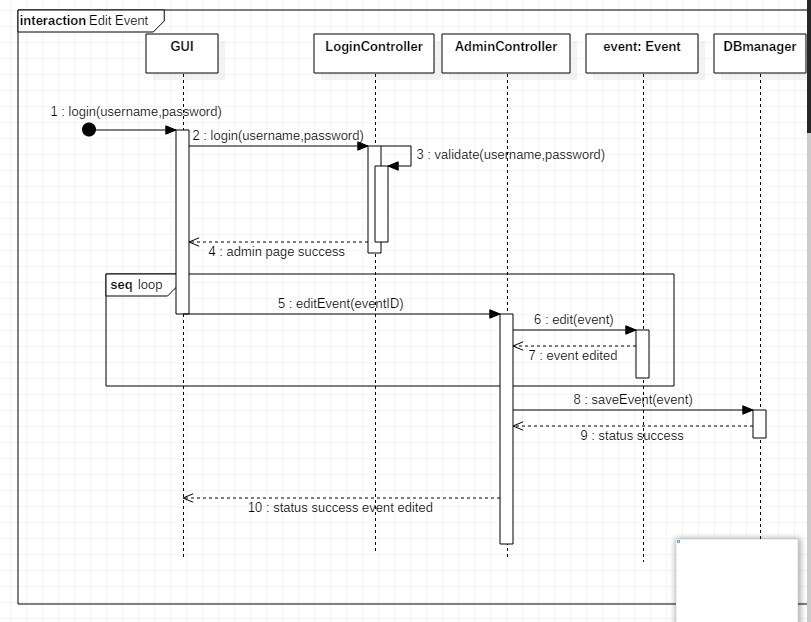
10. Success email is send to the student.

**5. Sequence Diagram**

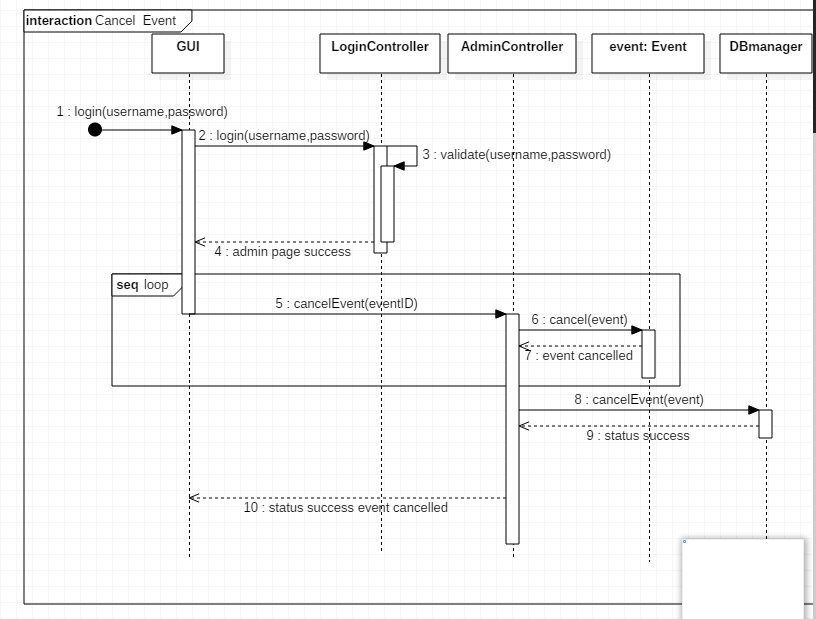
**5.1. Admin Add Event**



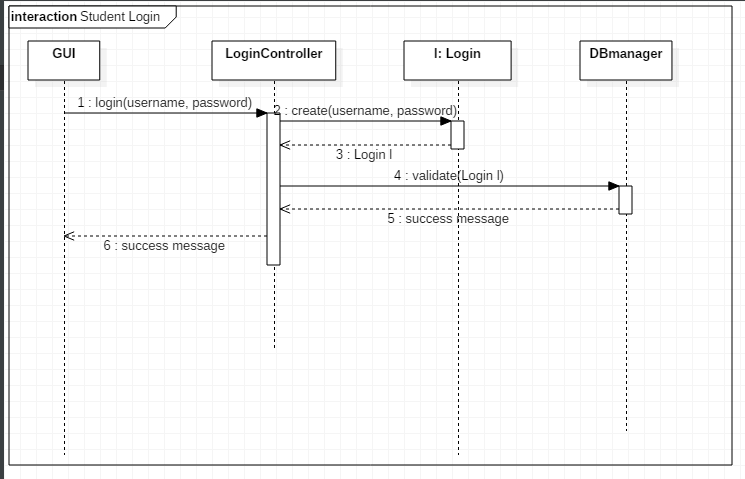
**5.2. Admin Edit Event**



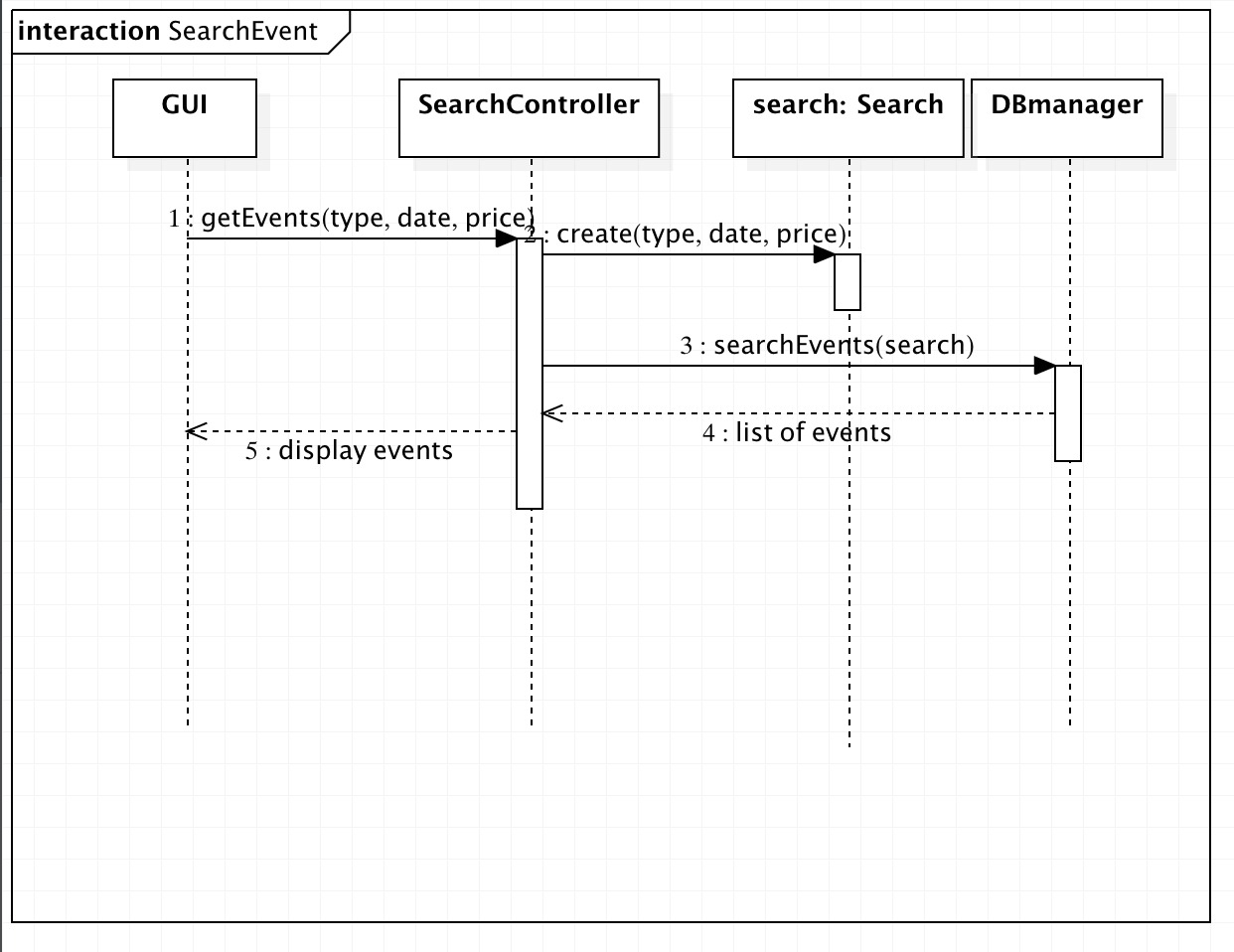
**5.3. Admin Cancel Event**



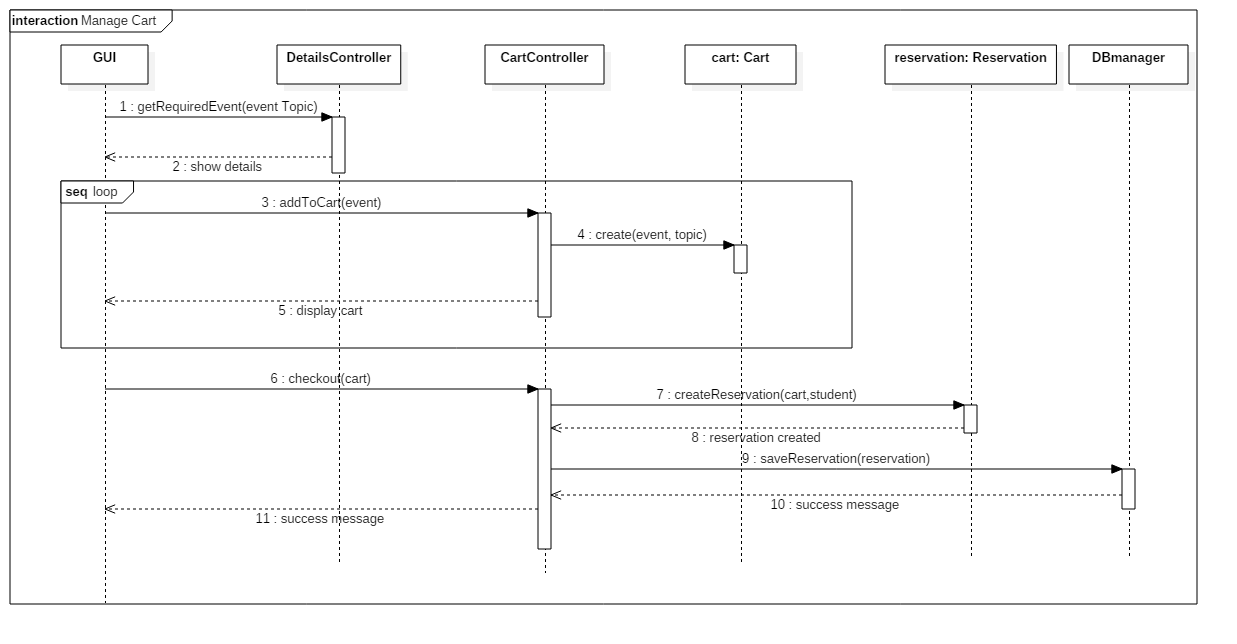
**5.4. Student Login**



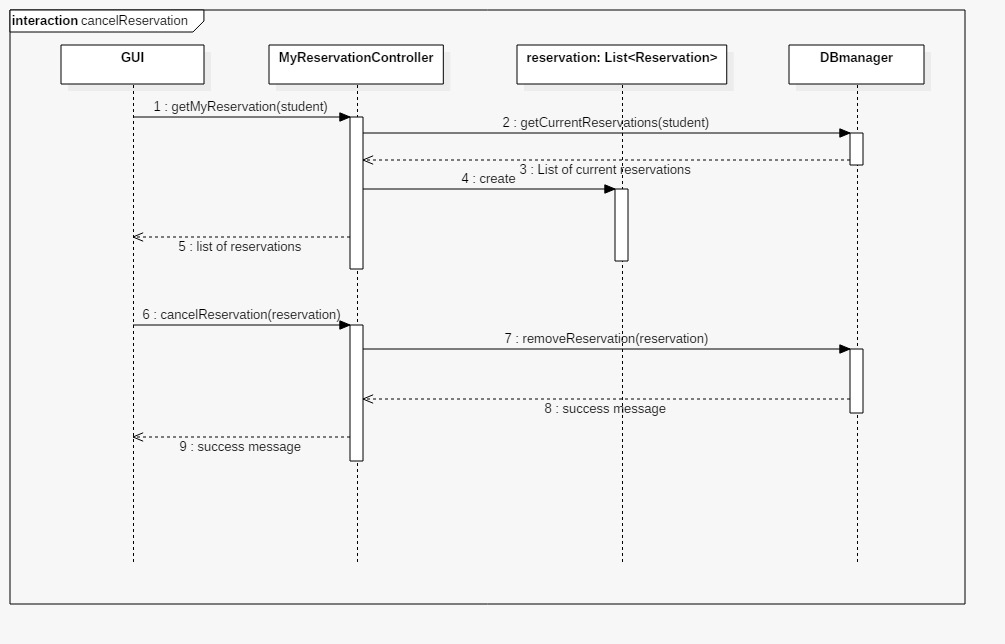
**5.5. Search Event**



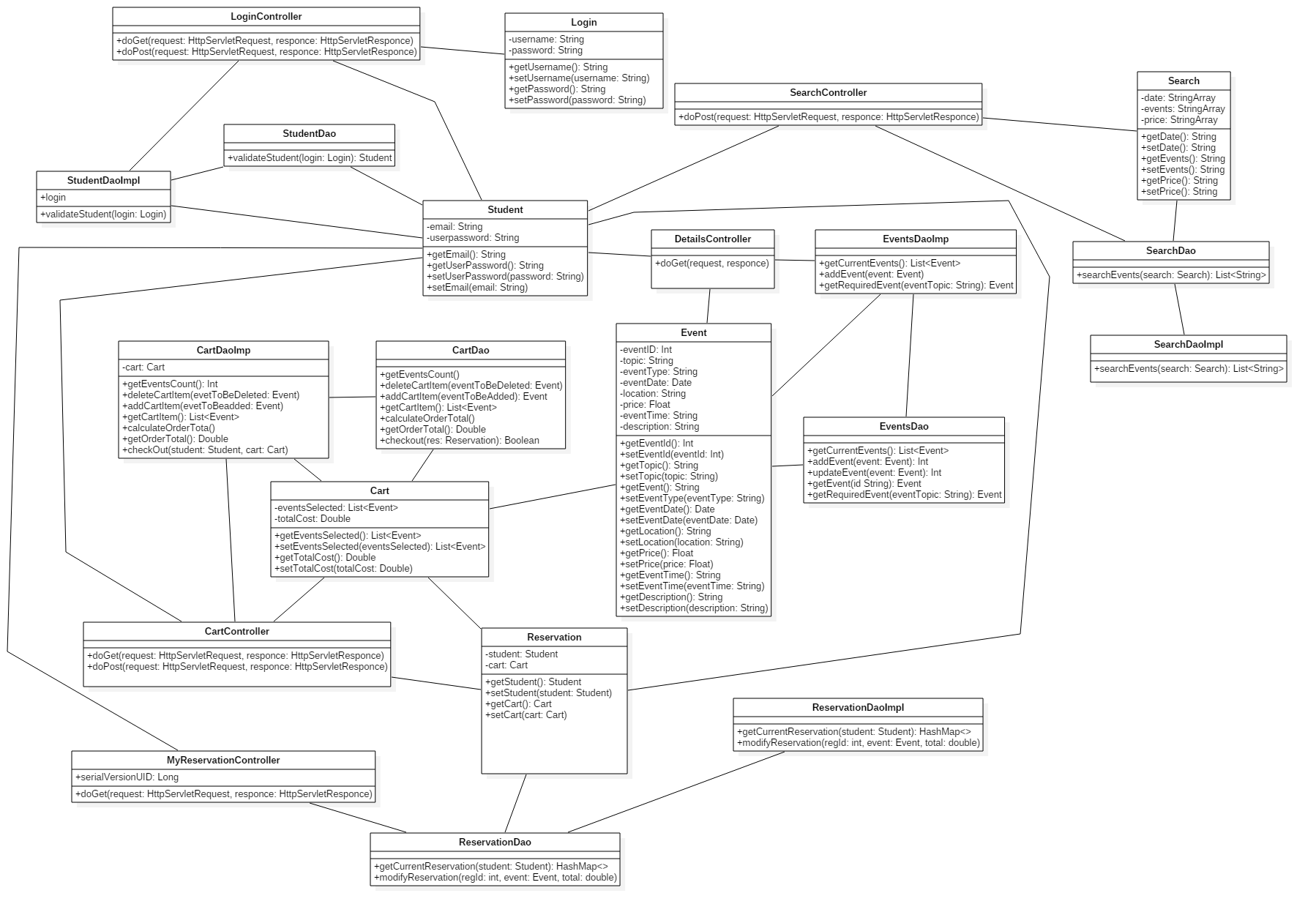
**5.6. Manage Cart**



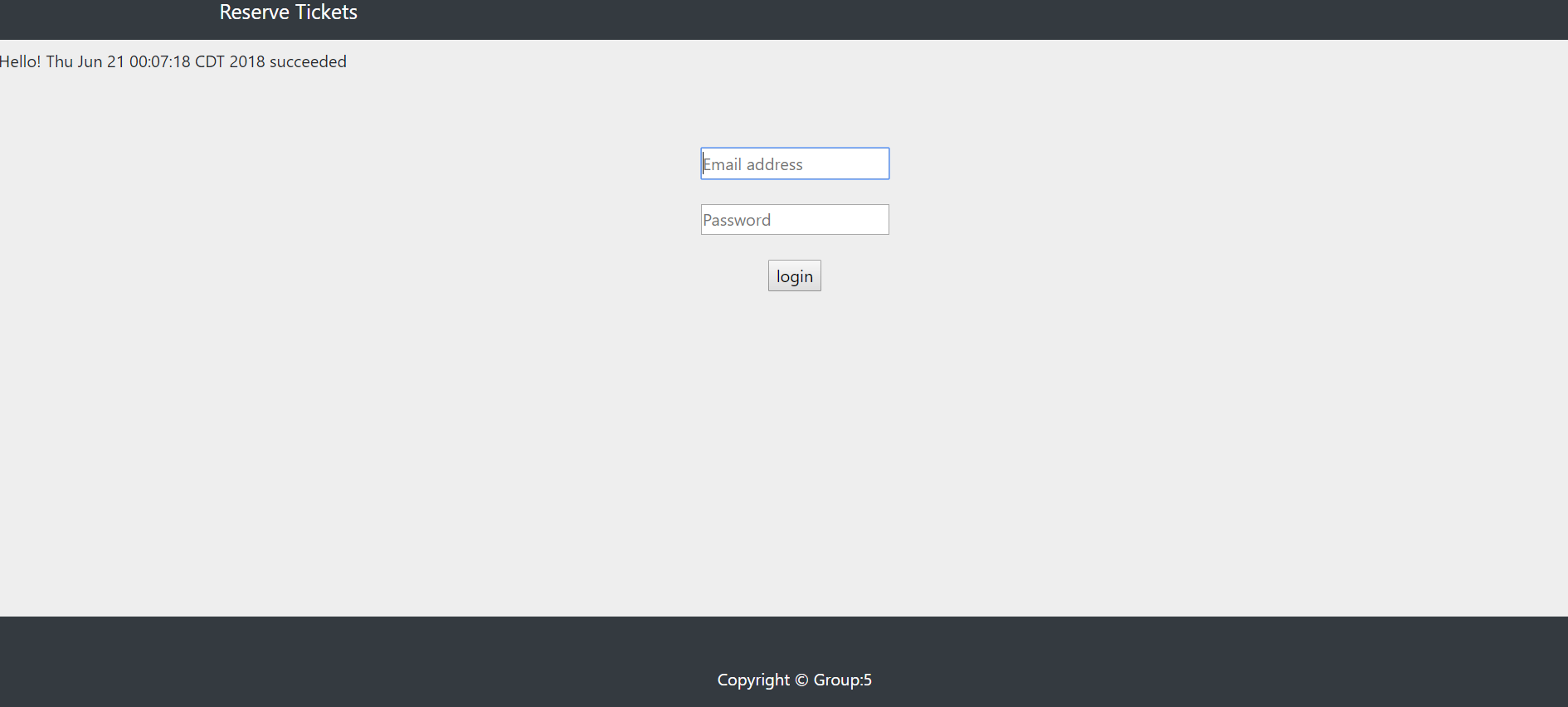
**5.7. Cancel Reservation**

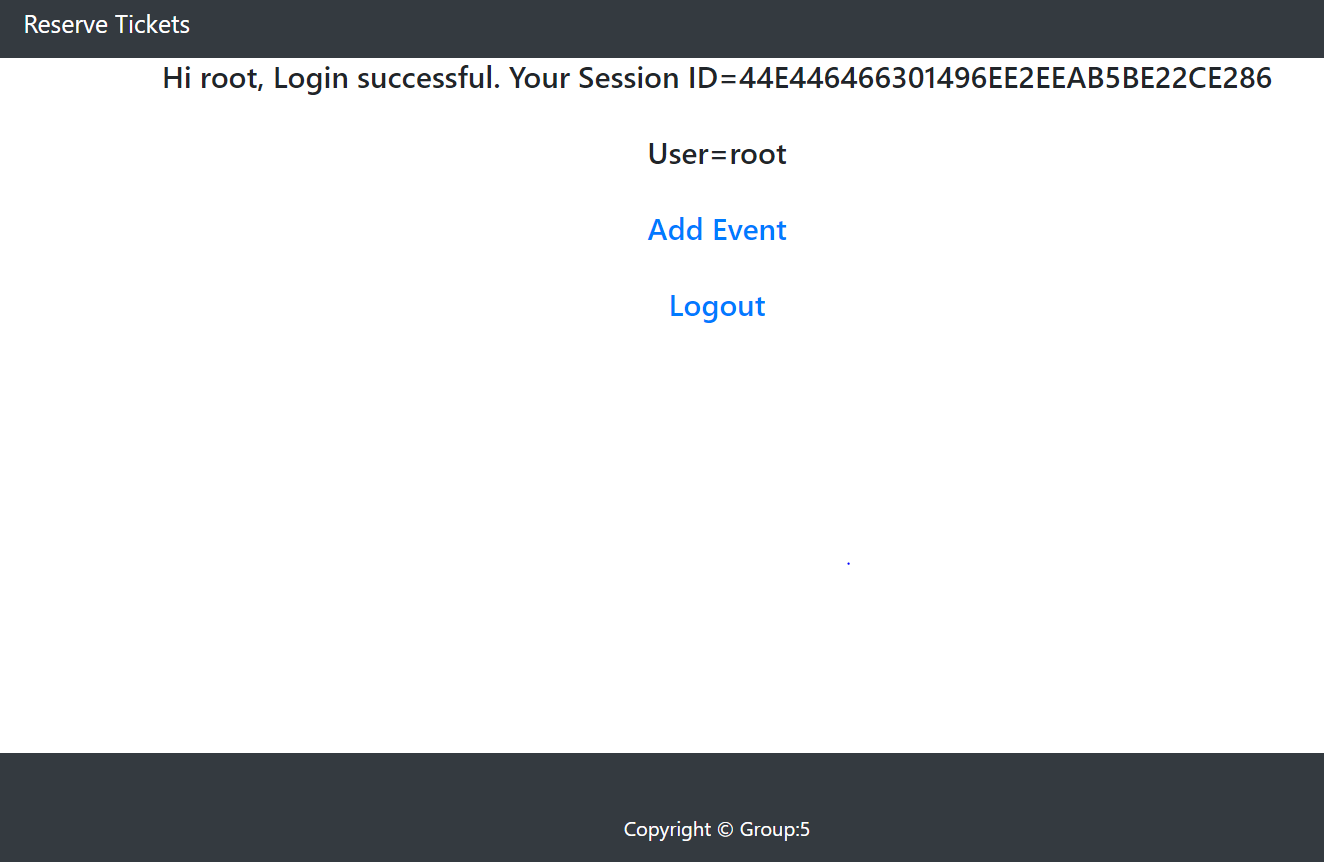


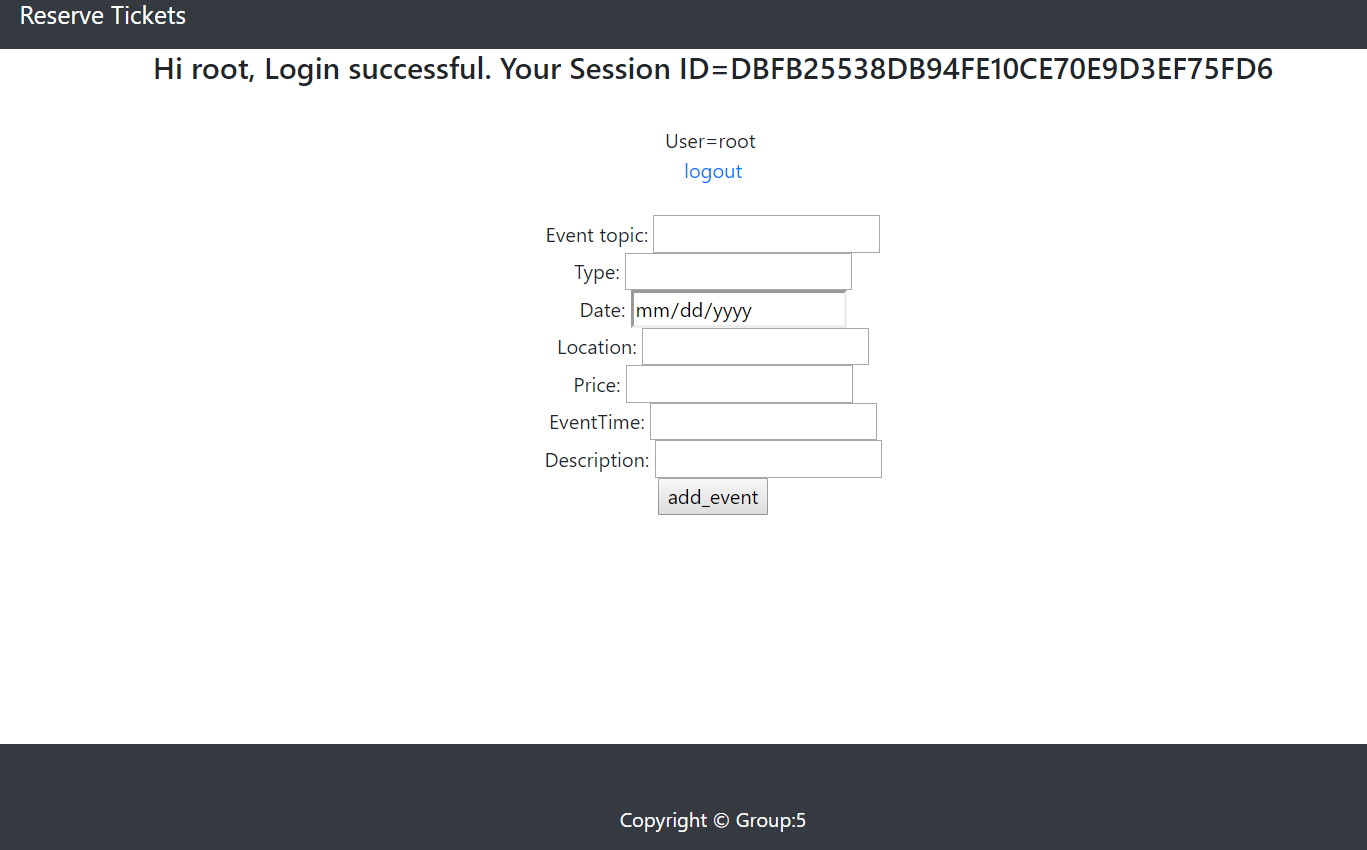
**6. Class Diagram**

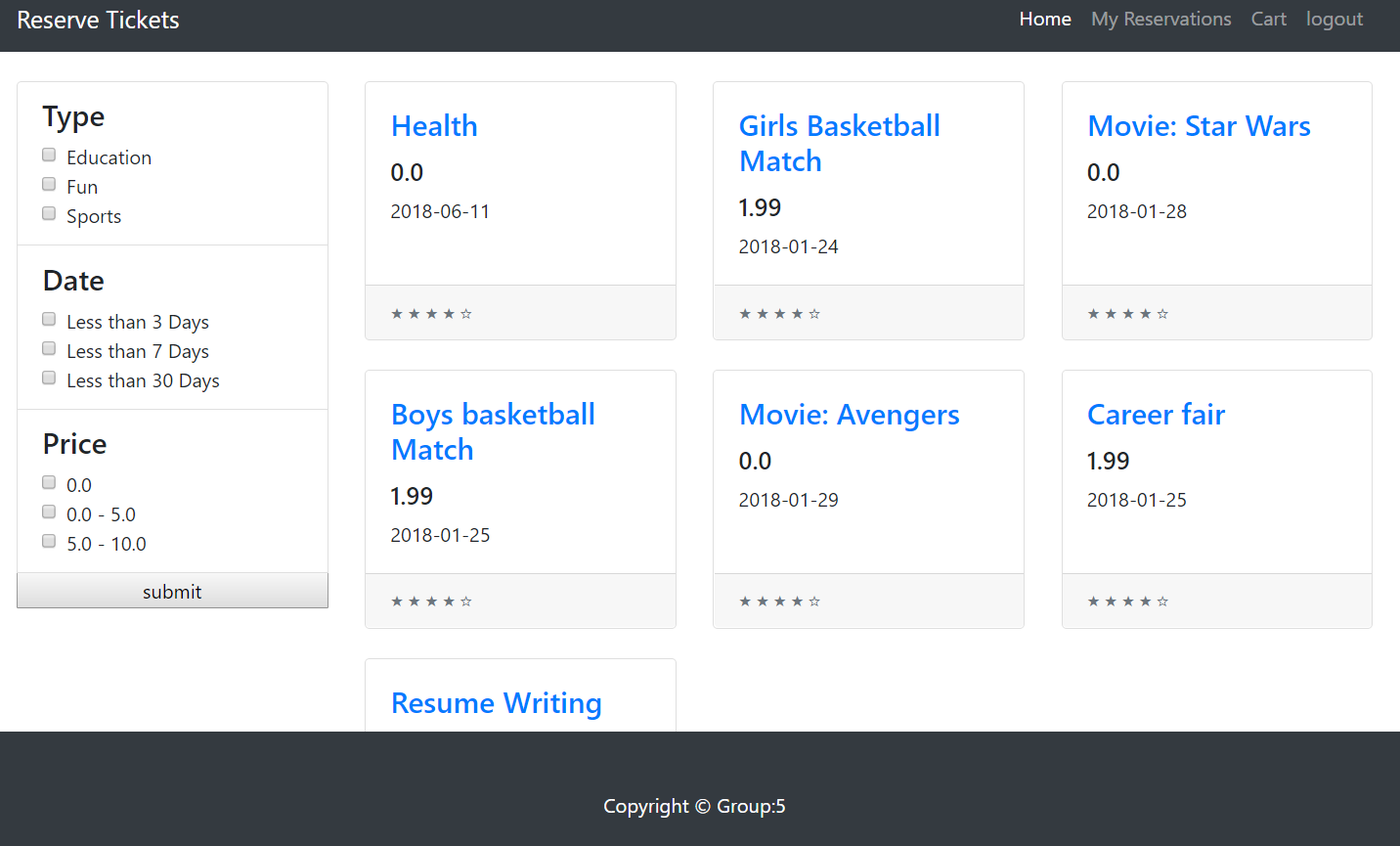


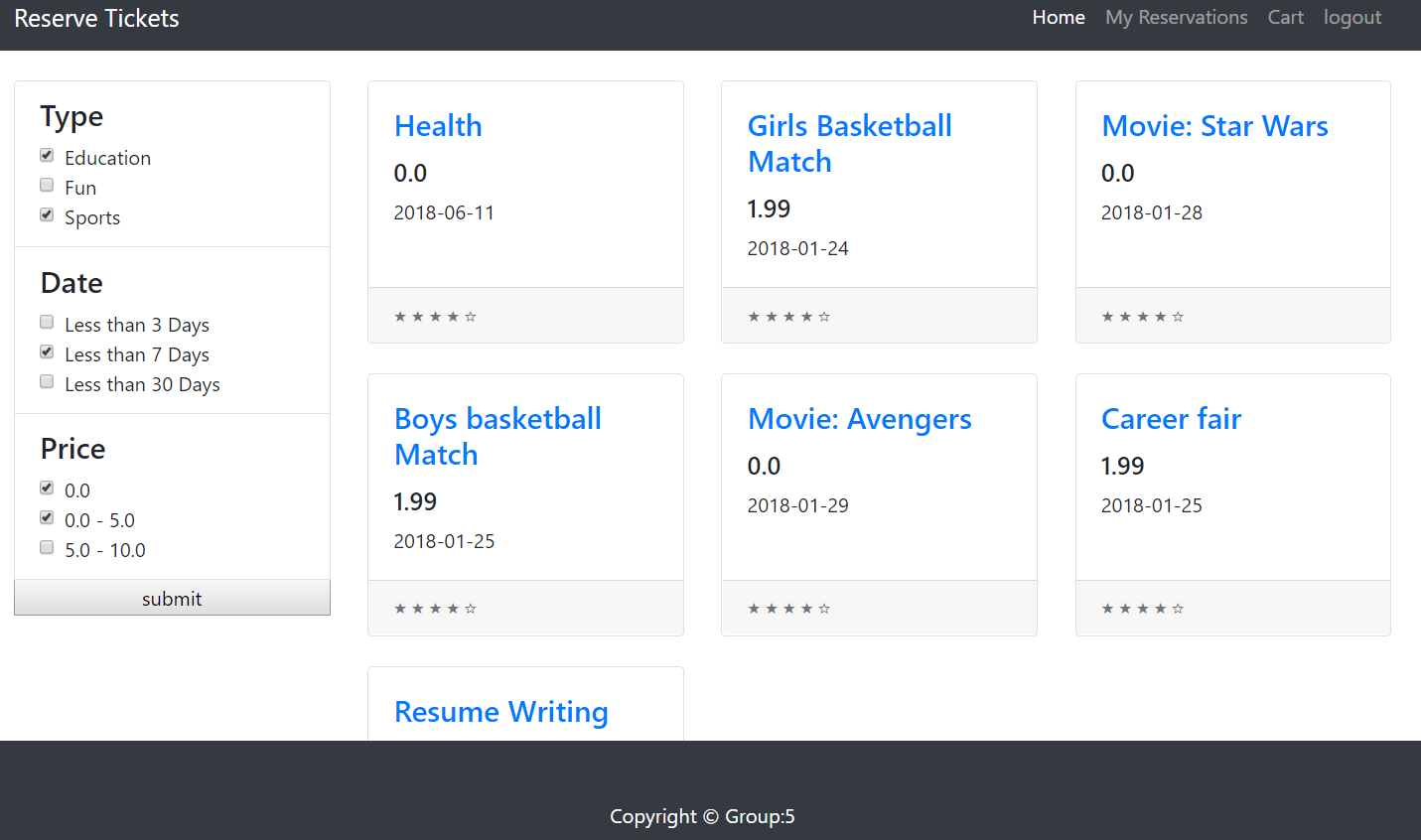
**7. Screenshots**











**8. Testing**

* Testing is done on the system to check if the system can finish tasks according to the requirements stated earlier successfully and smoothly.
* Facilitate users to finish desired tasks.
* Respond to user’s requests within specific time.
* System can run on IE, Chrome, and Firefox.
* Junit with Selenium is used in our project for testing.

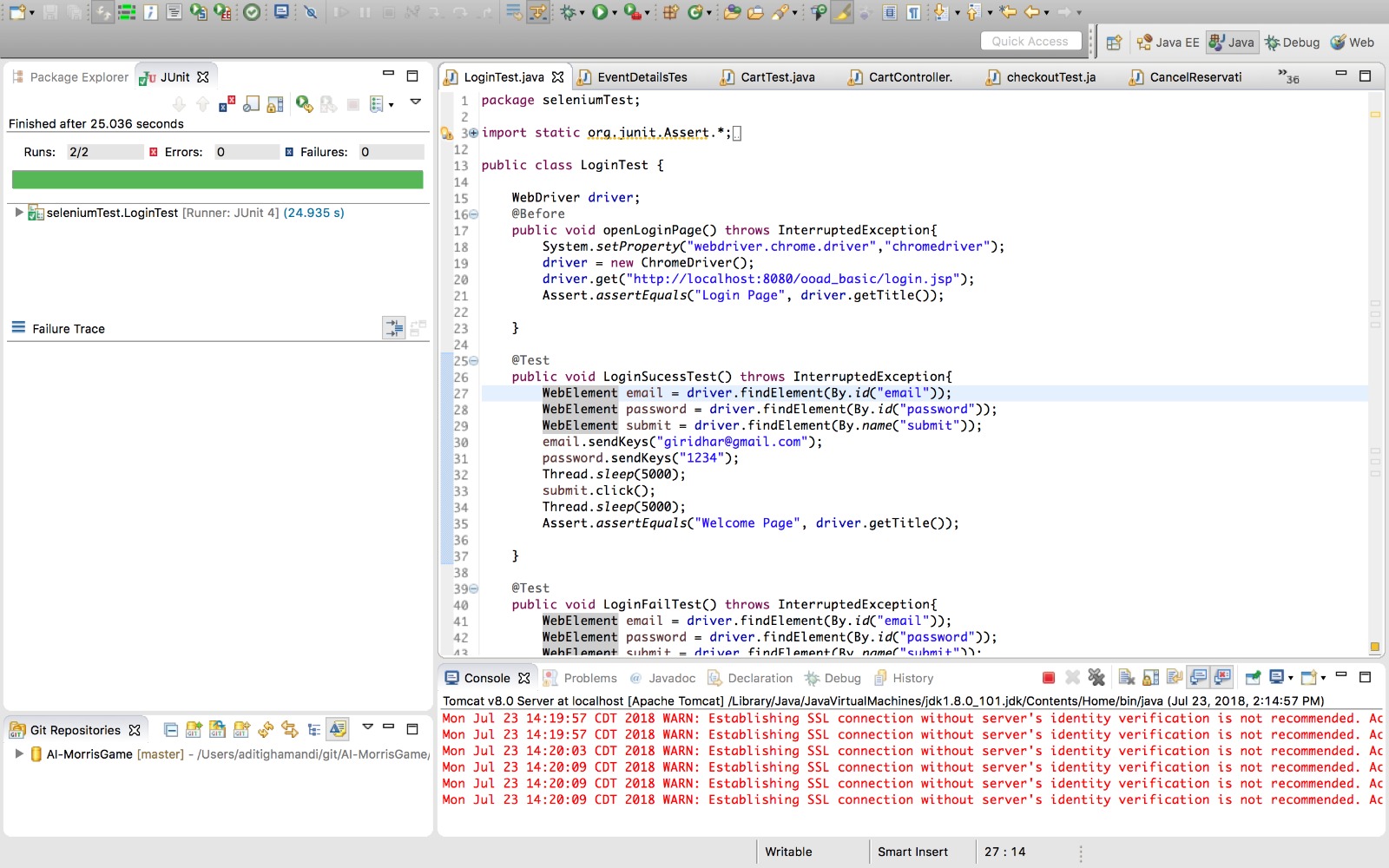
**Testing plan**

The following activities and test cases are used to check the quality and to map the implementation with the requirements of the system:

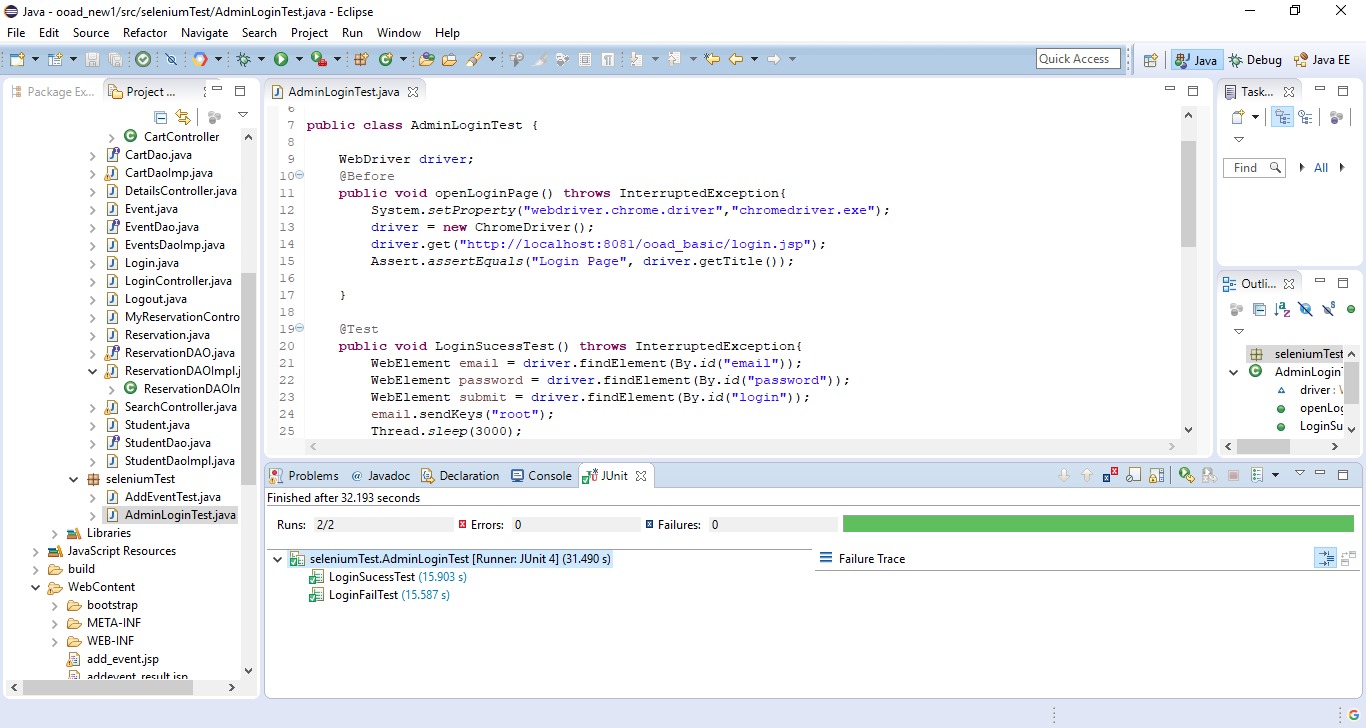
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.No. | Test Case | Test Case Description | Expected Output | Observed Output |
| 1. | Login (Student) | 1. Student enters email address and password to login.  2. Student information is fetched from the database. | Validates the current student and the welcome page is displayed. | Student can see the welcome page displaying all the current events. |
| 2. | Login (Administrator) | 1. Administrator enters email address and password to login.  2. Administrator information is fetched from the database. | Validates the current administrator and the administrator page is displayed. | Administrator can see the administrator page. |
| 3. | Add Events (Admin) | Admin can add events. | Event details to be added in the database. | A success message after the addition of current event and details are reflected in the database. |
| 4. | Search Events | 1. Student can search the events using the available filters. | Events to be displayed according to the selected filters. | Student can see the selected events according to the criteria selected. |
| 5. | Event Details | 1. Student selects an event from the searched results. | Details of the selected event should be displayed in a new web page. | Details of selected event are displayed on the next web page. |
| 6. | Cart Test | Add to Cart  1. Student selects an event to be added into the cart.  2. Student clicks on the add to cart button from the details page.  Delete from Cart  1. Student selects an event from the cart to be deleted.  2. Student clicks on the delete button. | Add to Cart  After student clicks on the add to cart button, the event should be displayed in the cart.  Delete from Cart  After student clicks on the delete button, the event should be deleted from the cart. | Add to Cart  Student can see the added events in the cart.  Delete from Cart  Student can no longer see the deleted event in the cart. |
| 7. | Checkout Cart | 1. Student can add required events to the cart.  2. Student should click on the checkout button to reserve the tickets. | Events are reserved after student clicks on checkout and a thank you page should be displayed. | After the checkout is done a thank you page is displayed confirming the reservation. |
| 8. | Cancel Reservation | 1. Student clicks my reservation tab from the welcome page.  2. Student cancels the desired event by clicking on the cancel button next to the event. | Event cancelled should not be displayed in my reservation tab. | Student can no longer see the cancelled event in my reservation tab. |

**Test Screenshots**

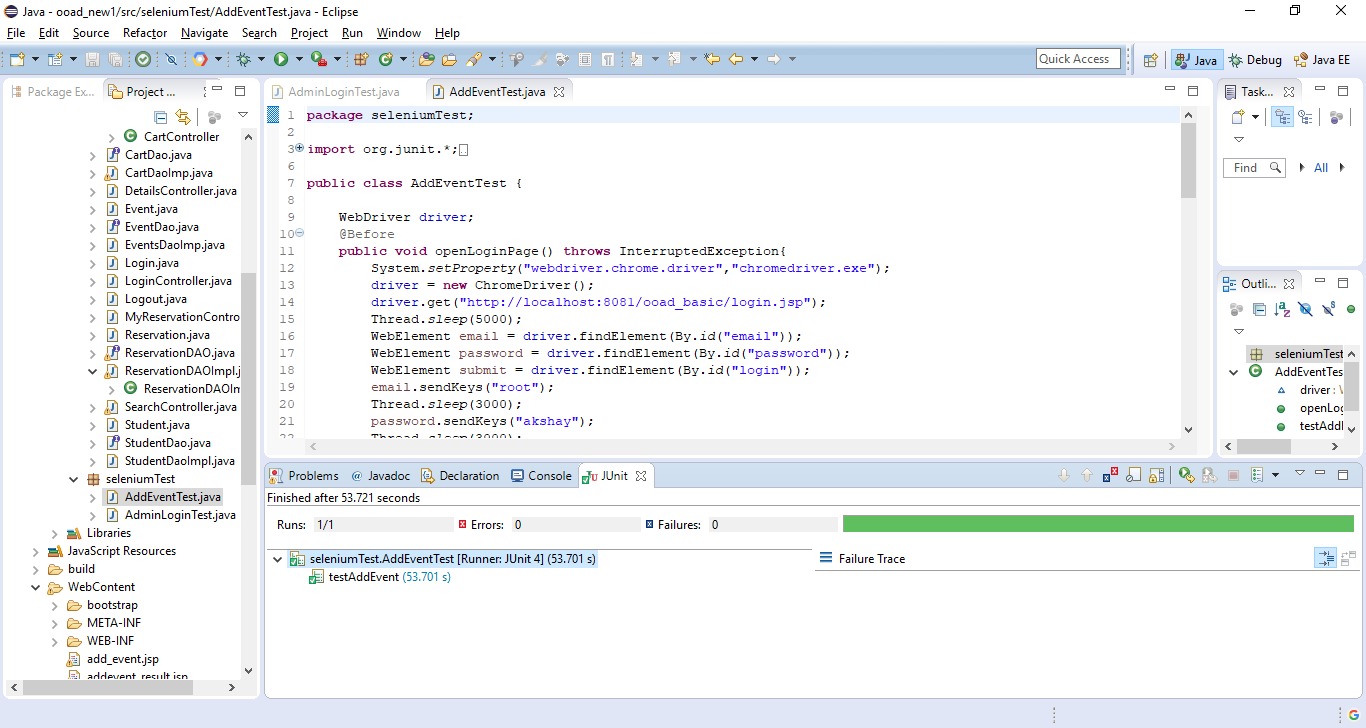
1. Login Student



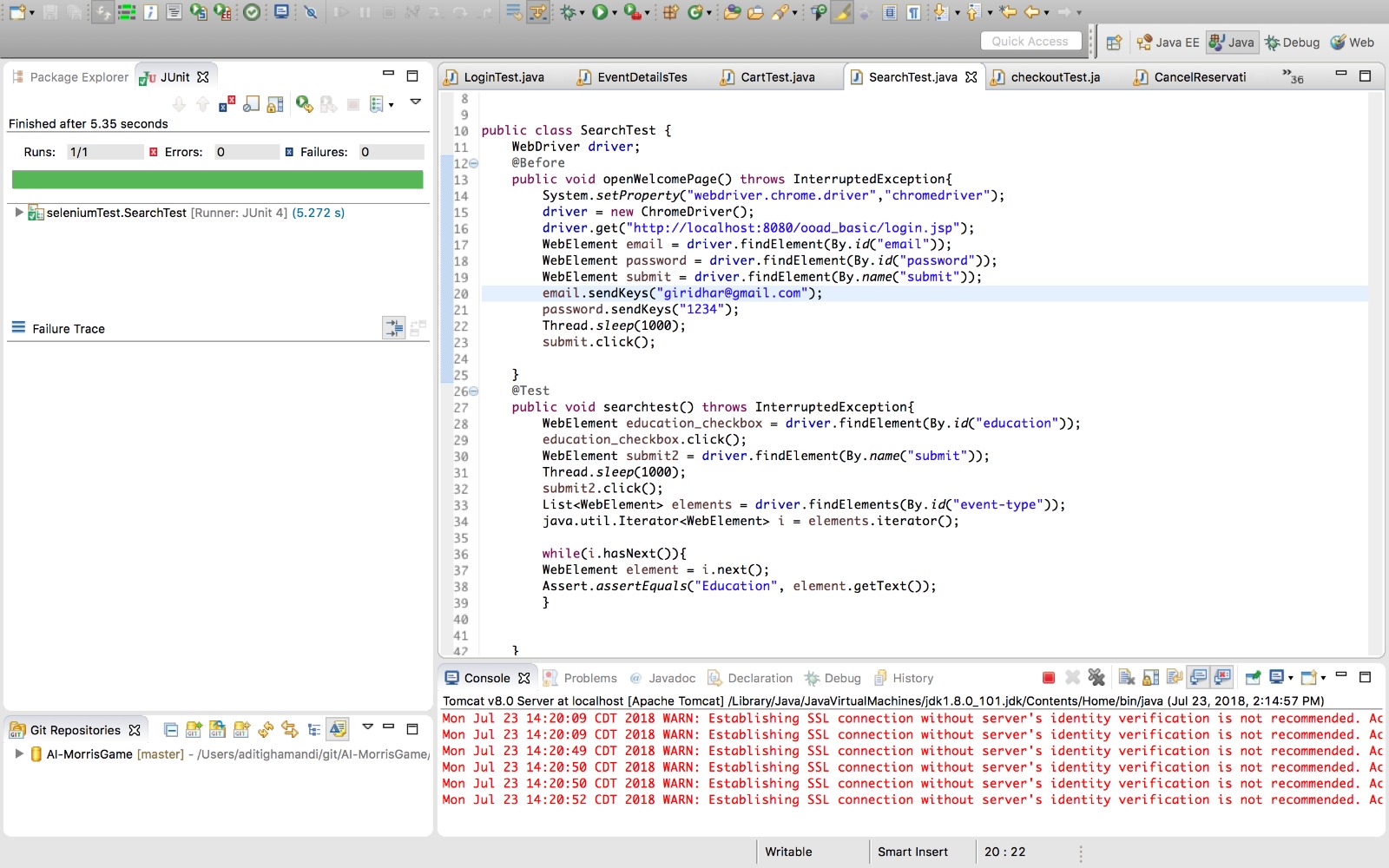
2. Login Admin



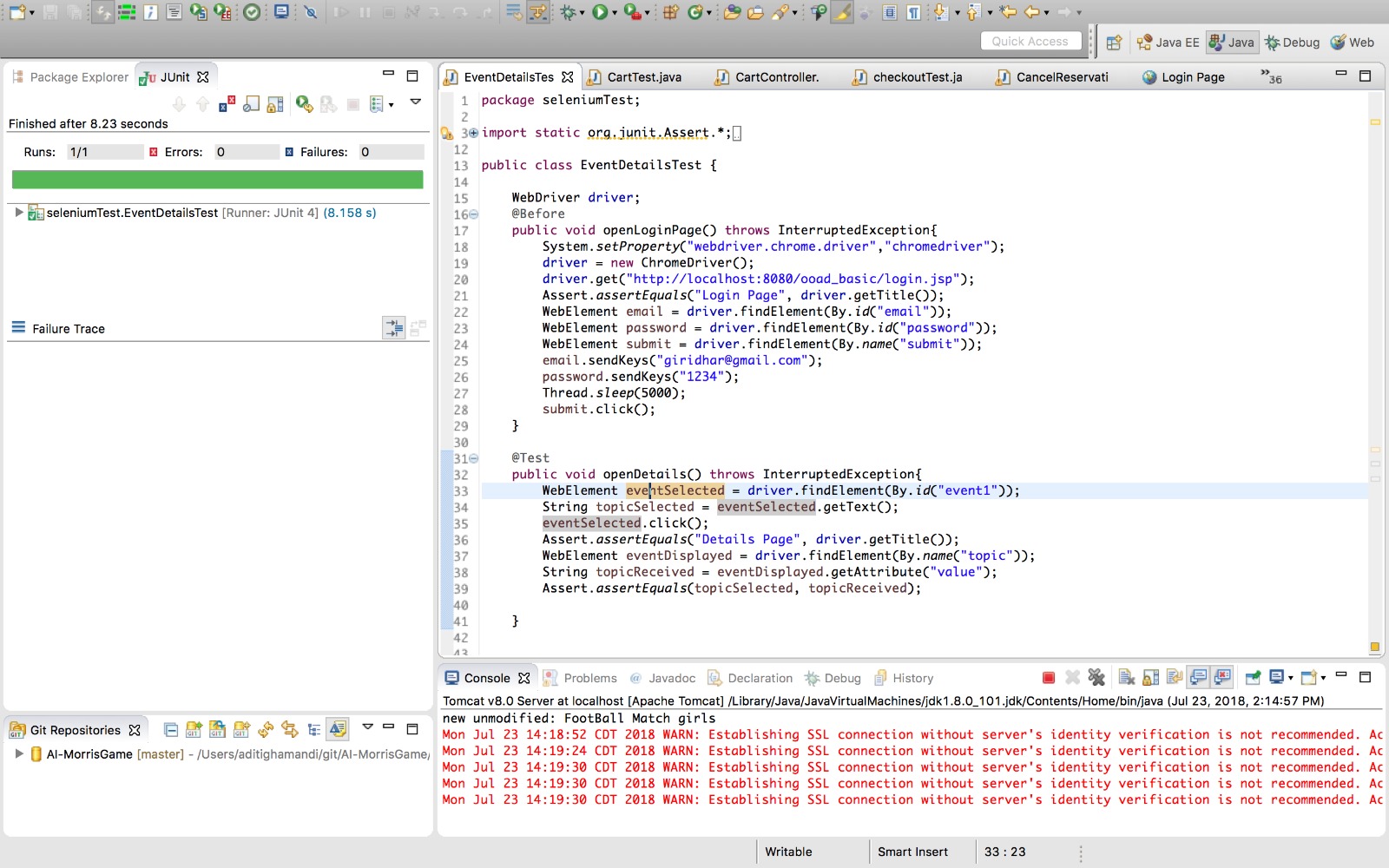
3. Add Event (Admin)



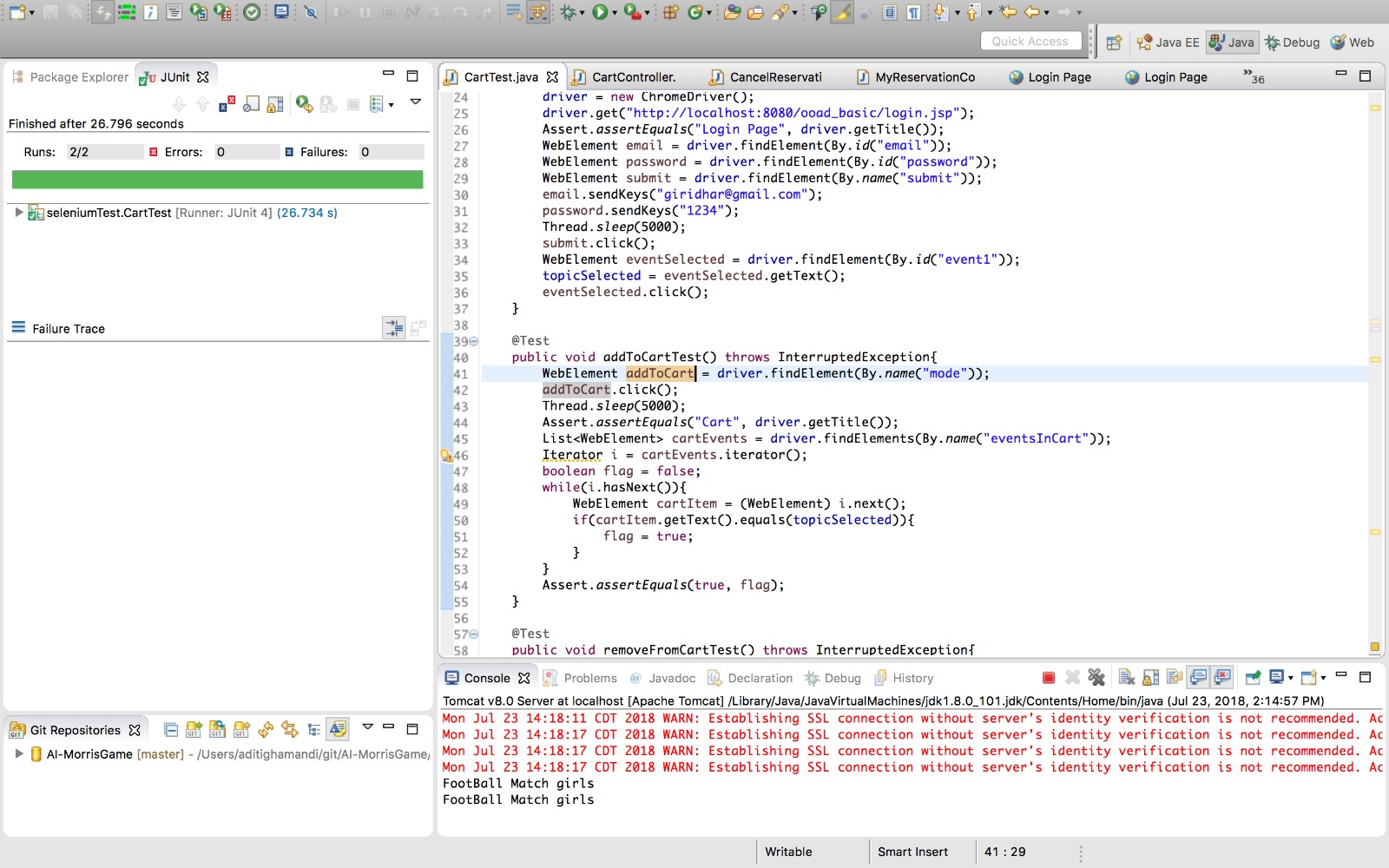
4. Search Event



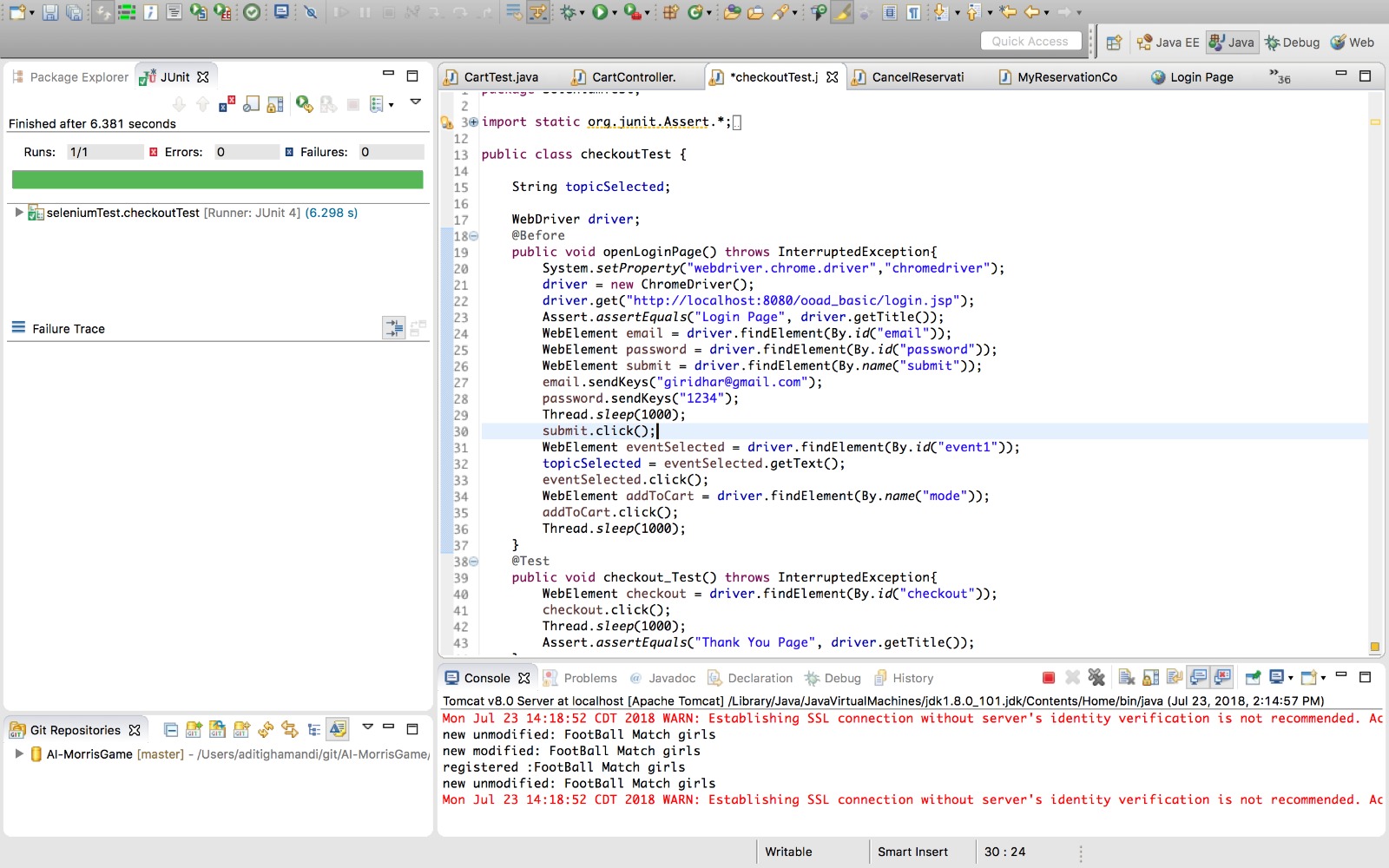
5. Event Details



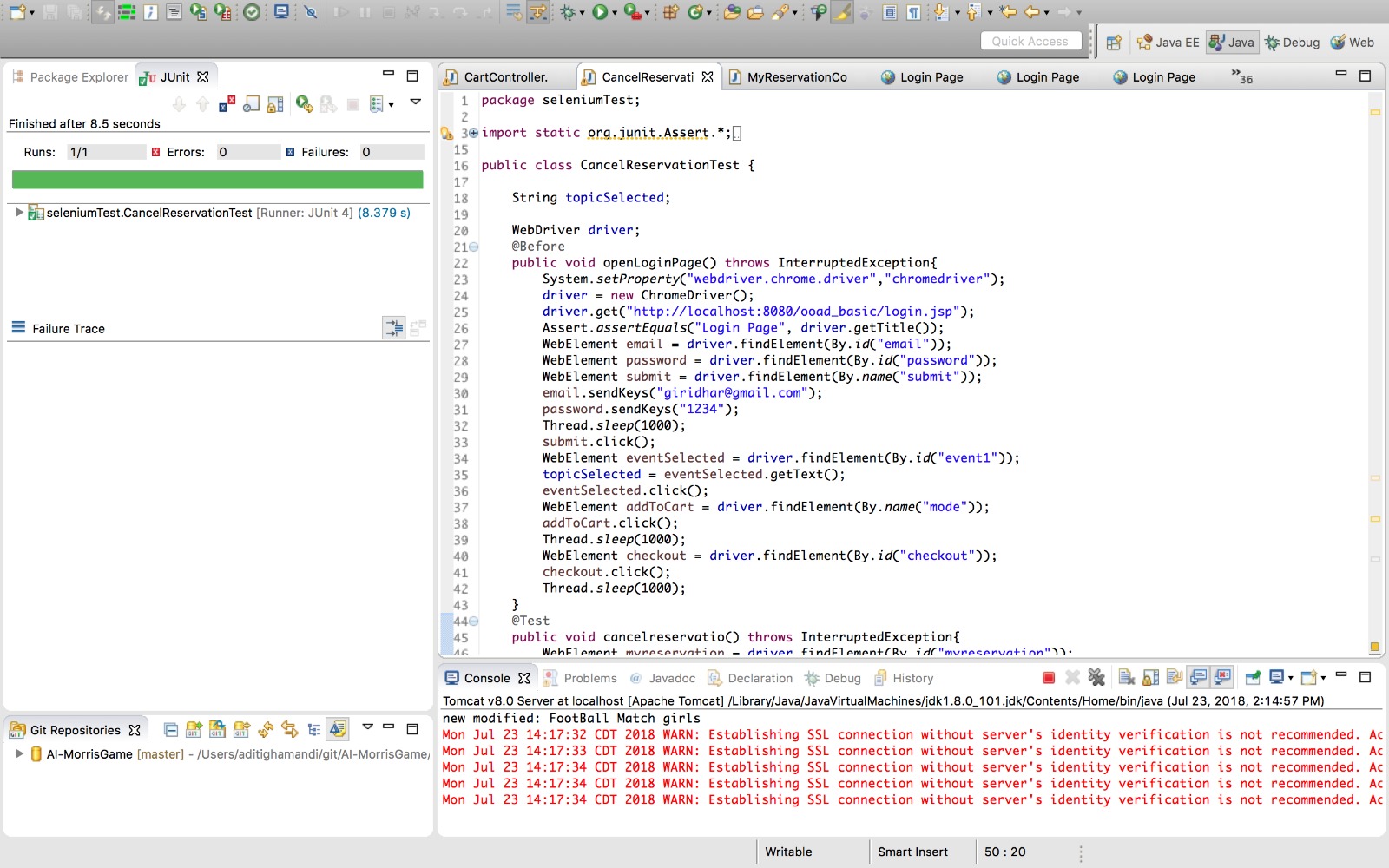
6. Cart Test



7. Checkout



8. Cancel Reservation



**9. Code**

https://github.com/Akshay1022/ooad\_project