Campus Activities

# 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

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.

## 1.1. 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.

# 2. 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.

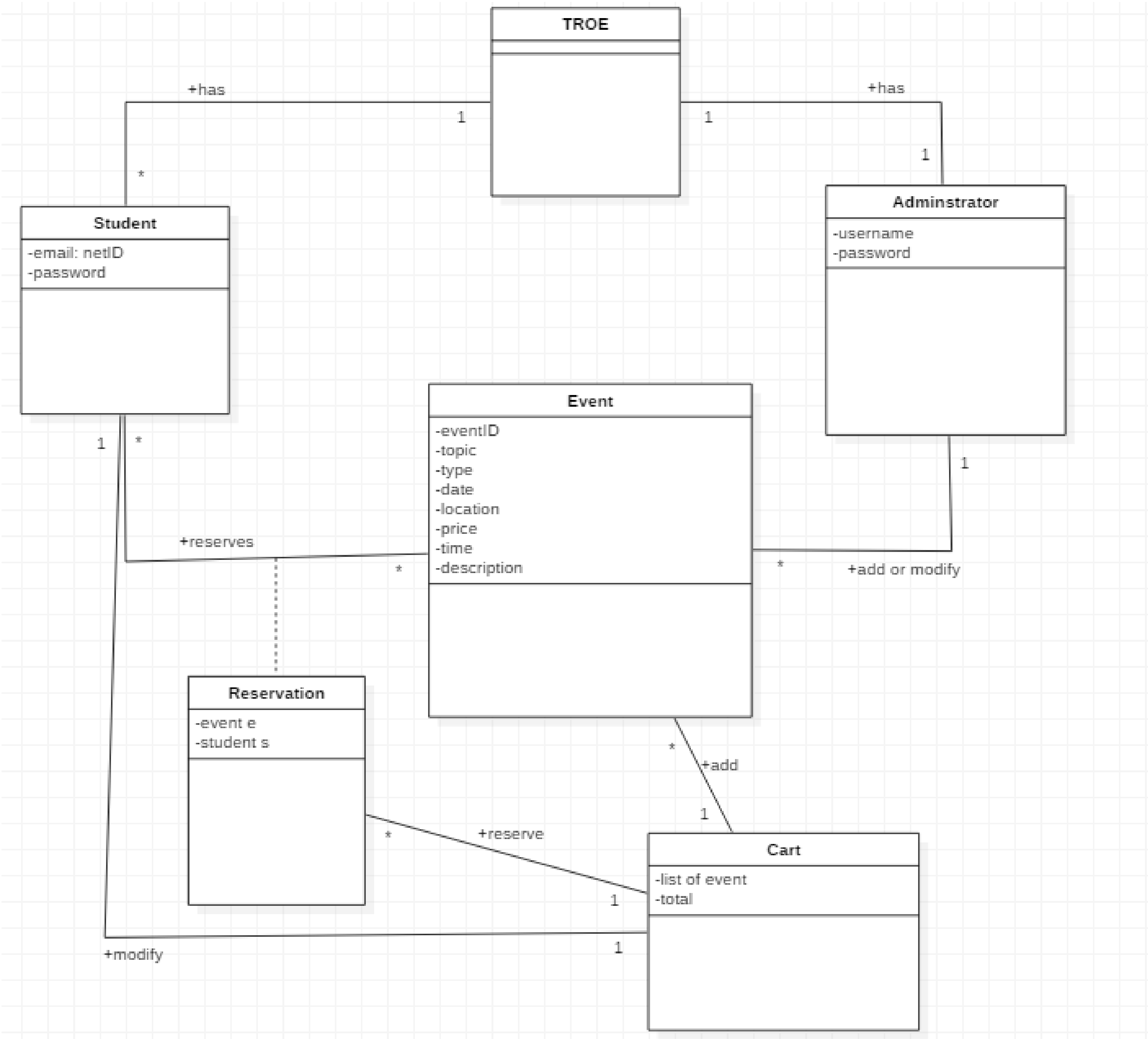
R6-System must generate a receipt for successful checkout.

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

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

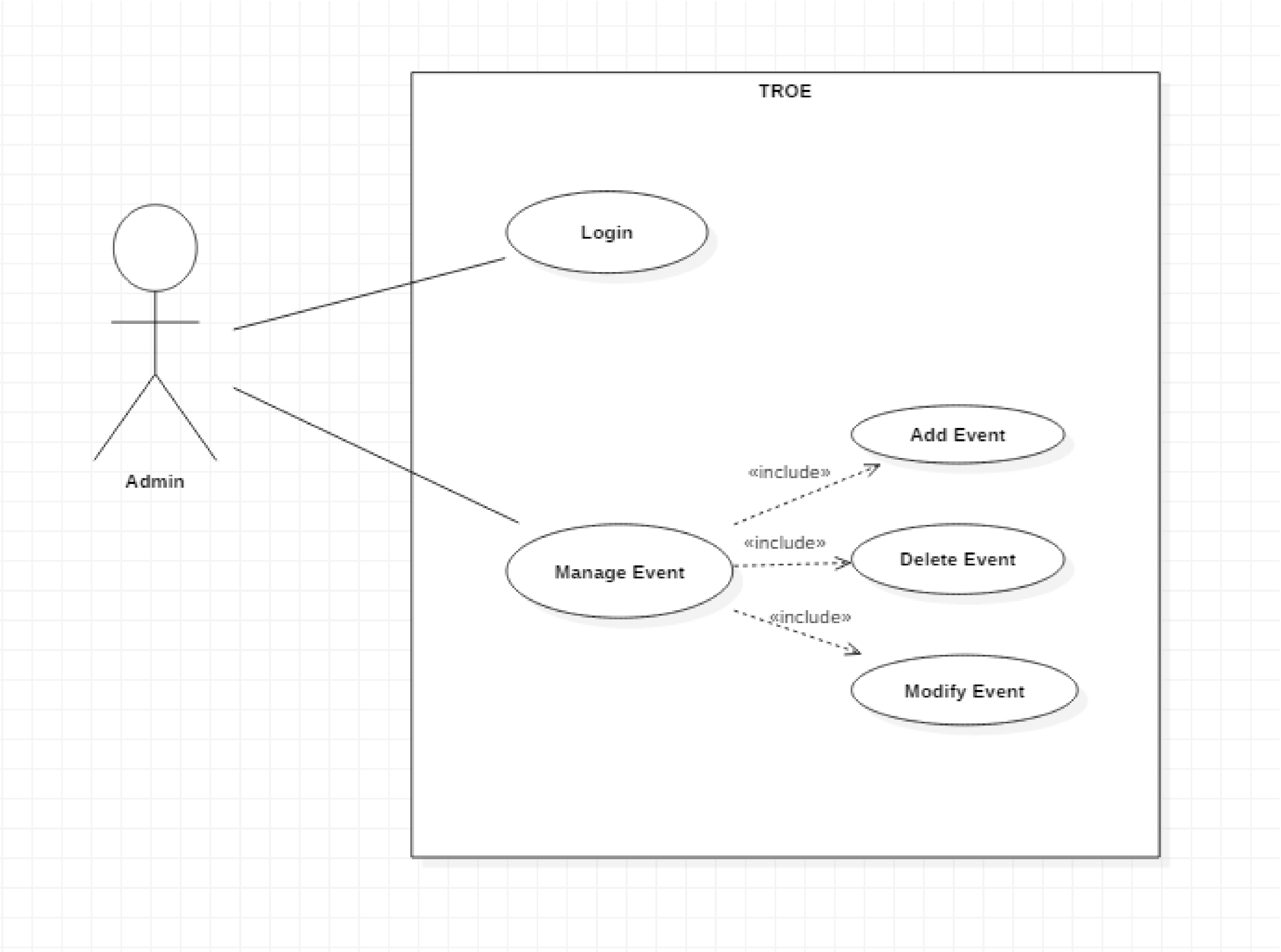
R9-System must provide a way to administrator to add and modify events.

## 3. Domain Model

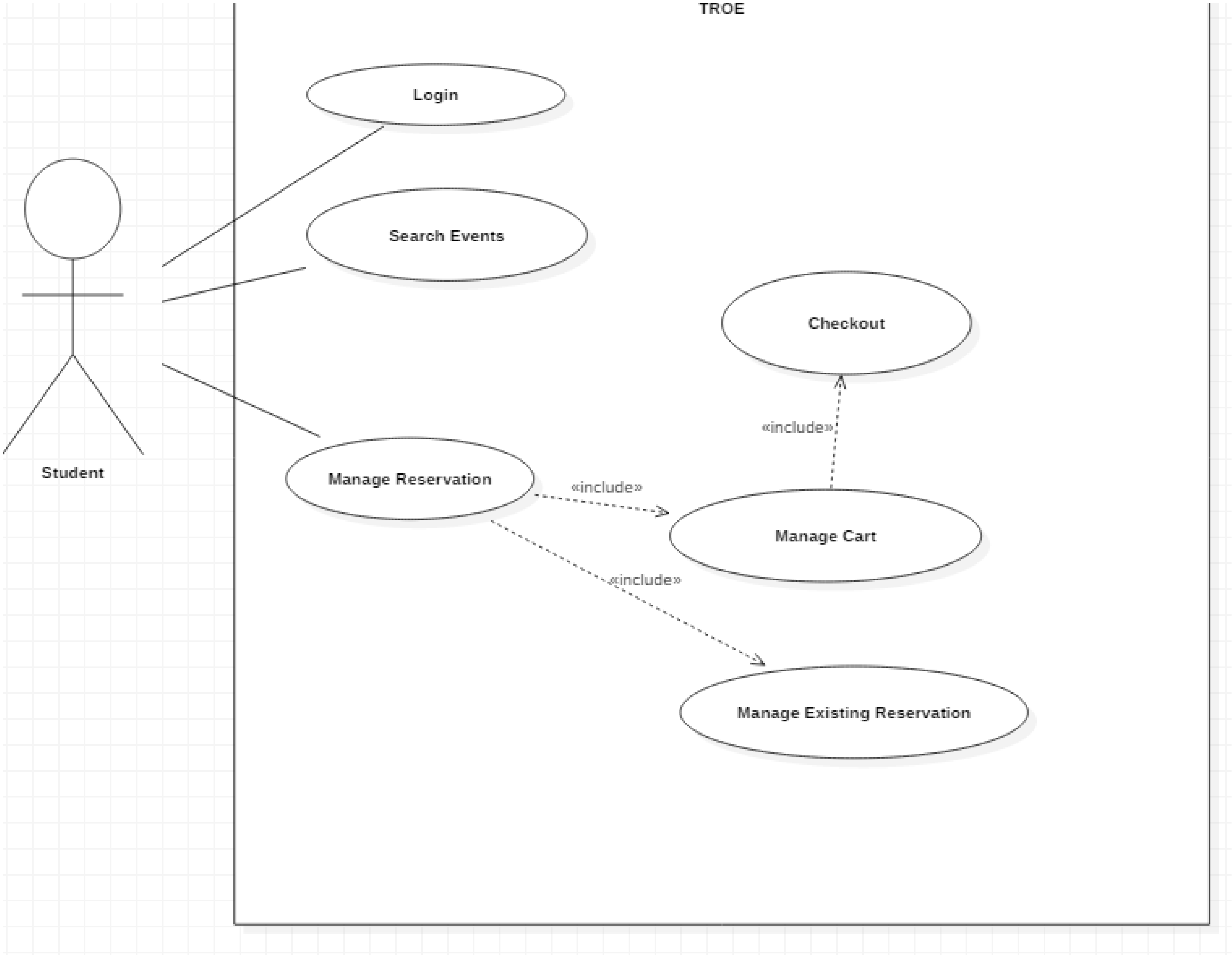


# 4. Use Case Model

## 4.1. Admin UCD



## 4.2. Student UCD



## 4.3. Expanded Use Cases

UC1 & UC3: Login

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

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, location). | 2. a. System will display all the programs according to search criteria.<br>b. The search results are empty. |
| 3. a. TUCEW, Student can select the desired events.<br>b. TUCEW, Display message “no events available”. | |

UC5: Manage Reservations

| 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. | 2. a. System will display the details of the event selected.<br>b. System will display that event is no longer available. |
| 3. a. Student can add the selected event into the cart.<br>b. Student can go back to the current list to select a different event or to add more events into the cart | 4. a. System will add the events into the cart.<br>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. |
| 5. a. Student can open the cart and press the checkout button.<br>b. Student can open the cart, delete any event from the and then press checkout. | 6. a. System will display a thank you page stating you have registered for selected events.<br>b. If the cart is empty then system will display an error message and ask student to reserve for the even again. |
| 7. a. TUCEW, Student will receive a receipt for the registered events.<br>b. TUCEW, Display message "you are not registered try again". | |

## 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.

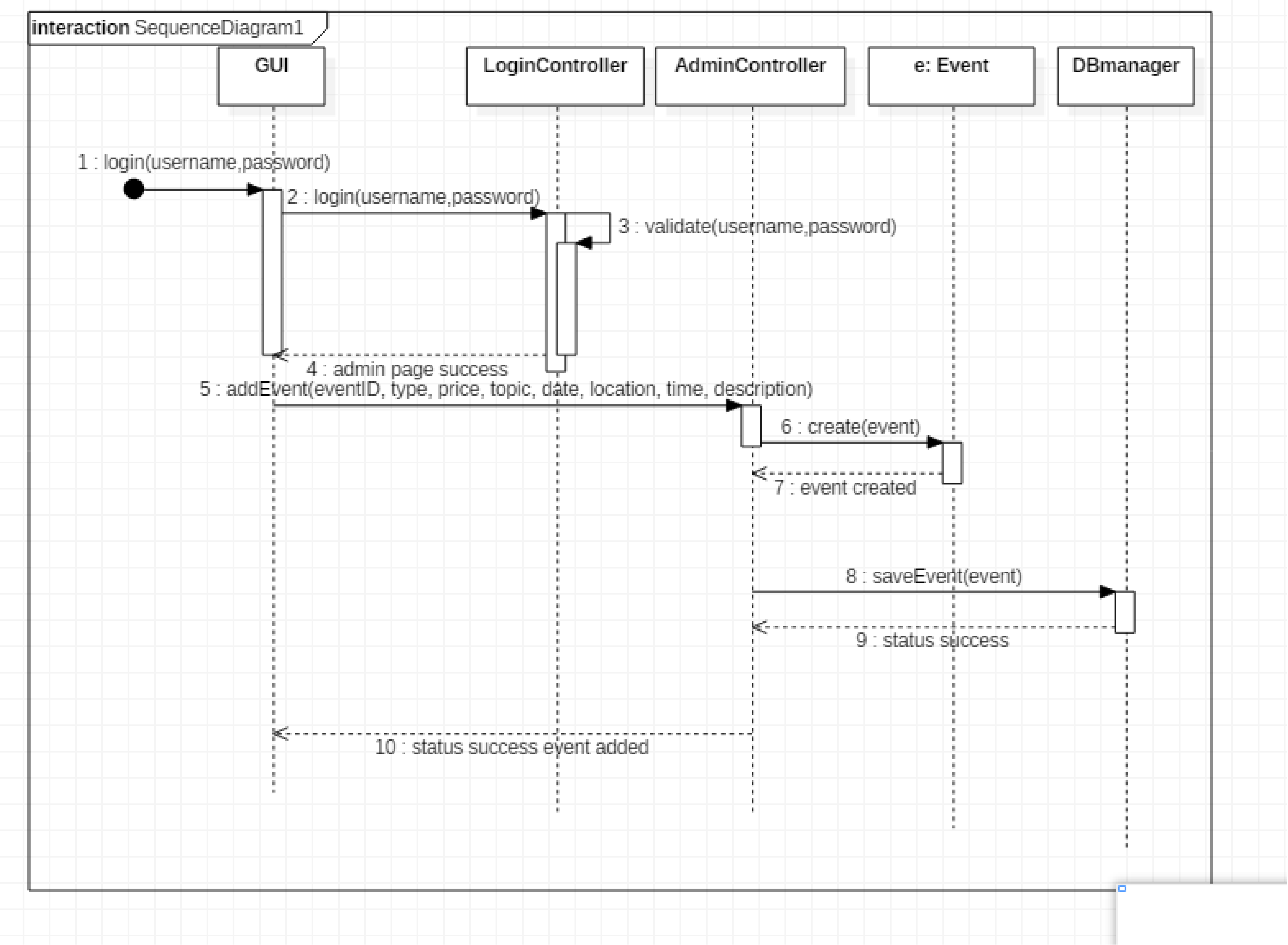
Display the event is full and the student cannot register for the event.

9. Receipt will be generated for the registered events.

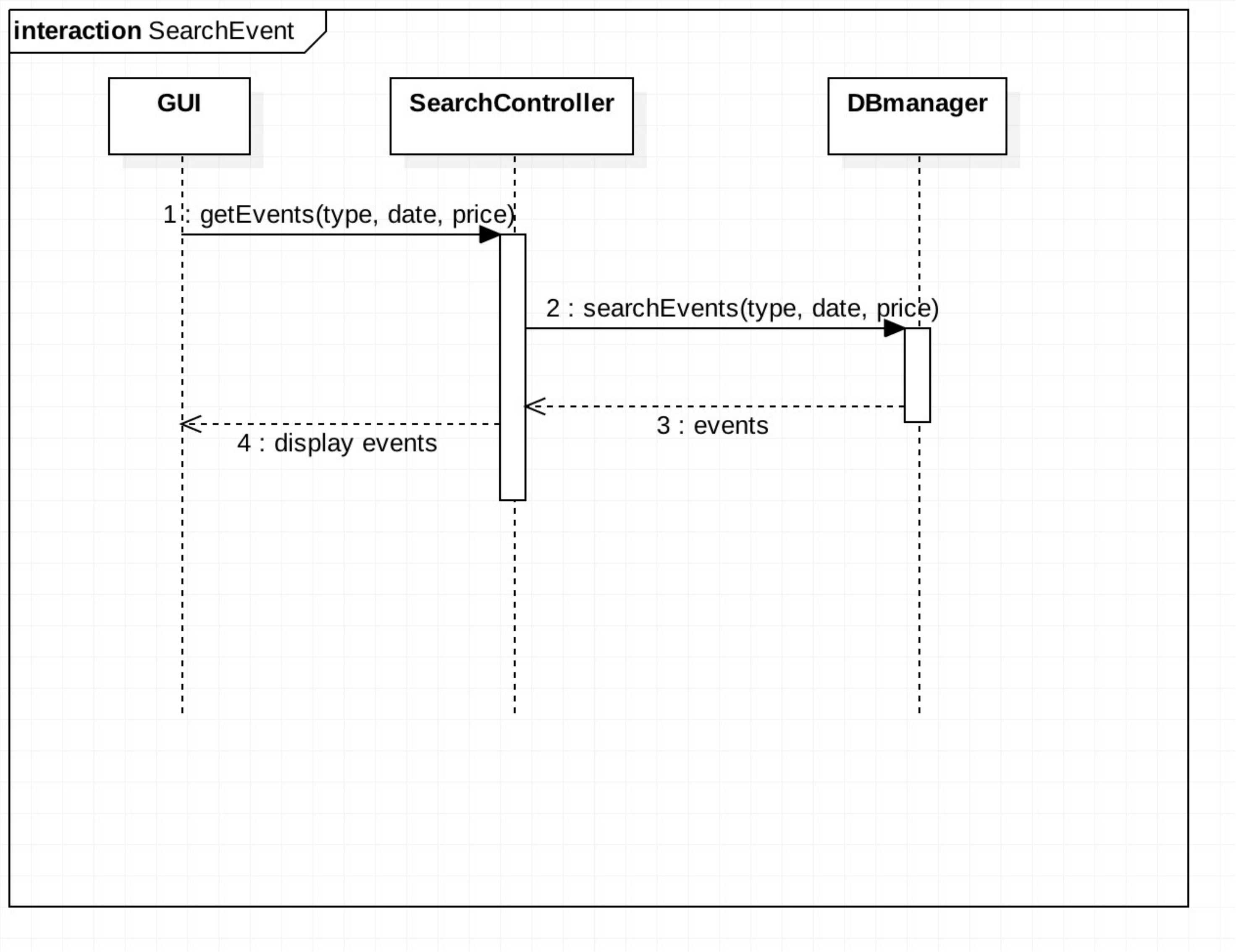
10. Success email is send to the student.

## 5. Sequence Diagram

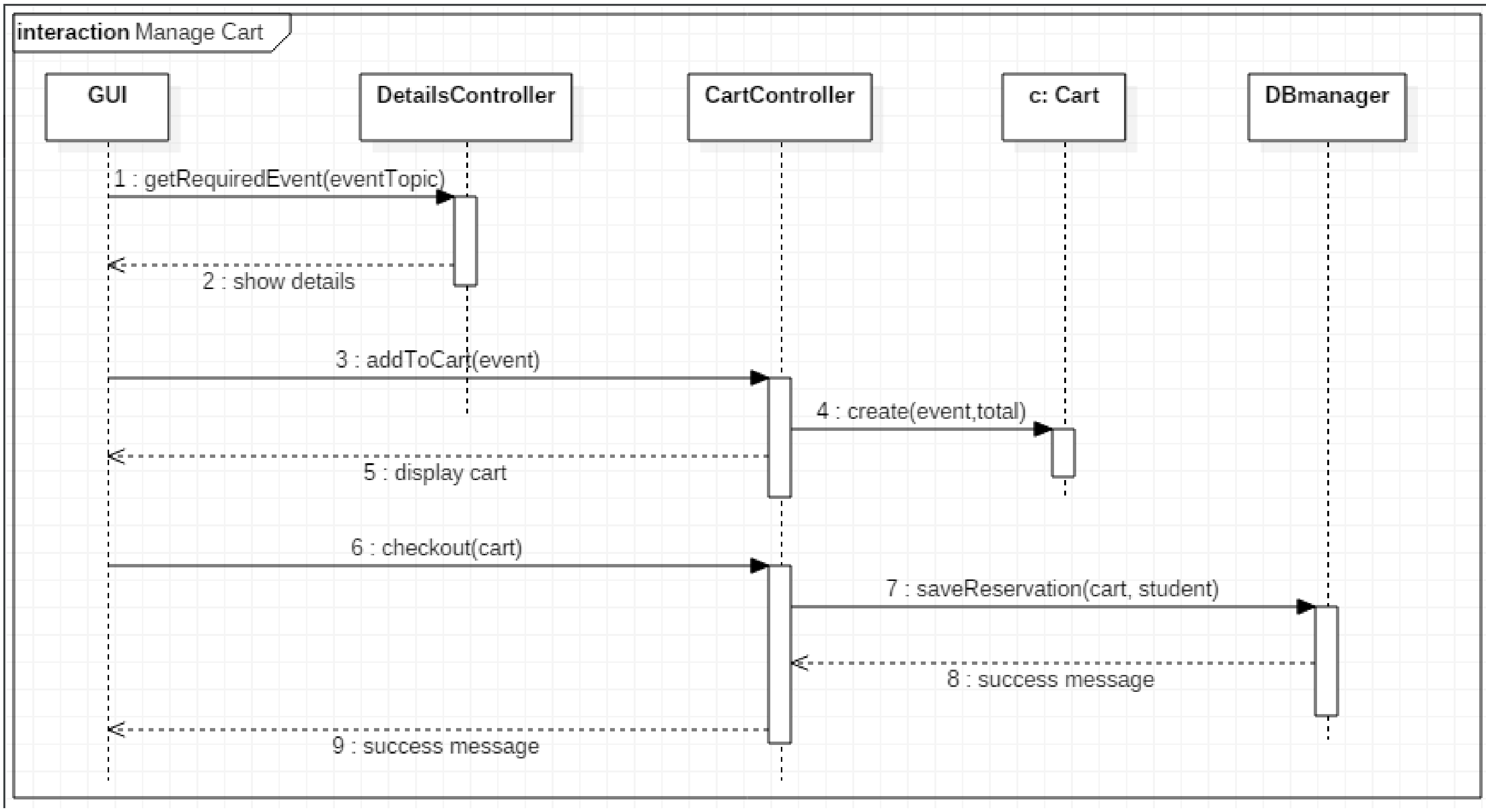
### 5.1. Admin Add Event


interaction SequenceDiagram1
GUI
LoginController
AdminController
e: Event
DBmanager
1 : login(username,password)
2 : login(username,password)
3 : validate(username,password)
4 : admin page success
5 : addEvent(eventID, type, price, topic, date, location, time, description)
6 : create(event)
7 : event created
8 : saveEvent(event)
9 : status success
10 : status success event added

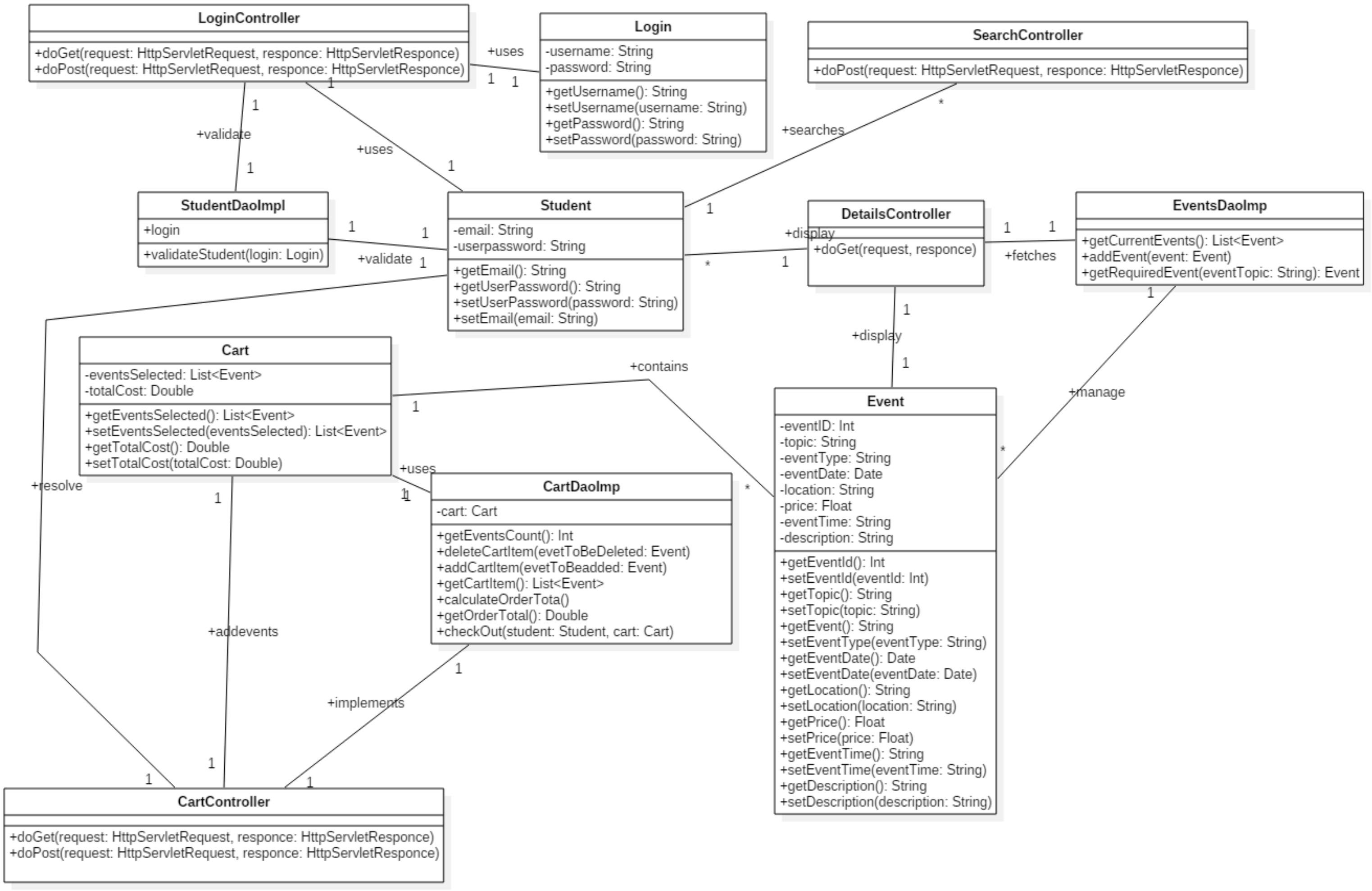
### 5.2. Search Event



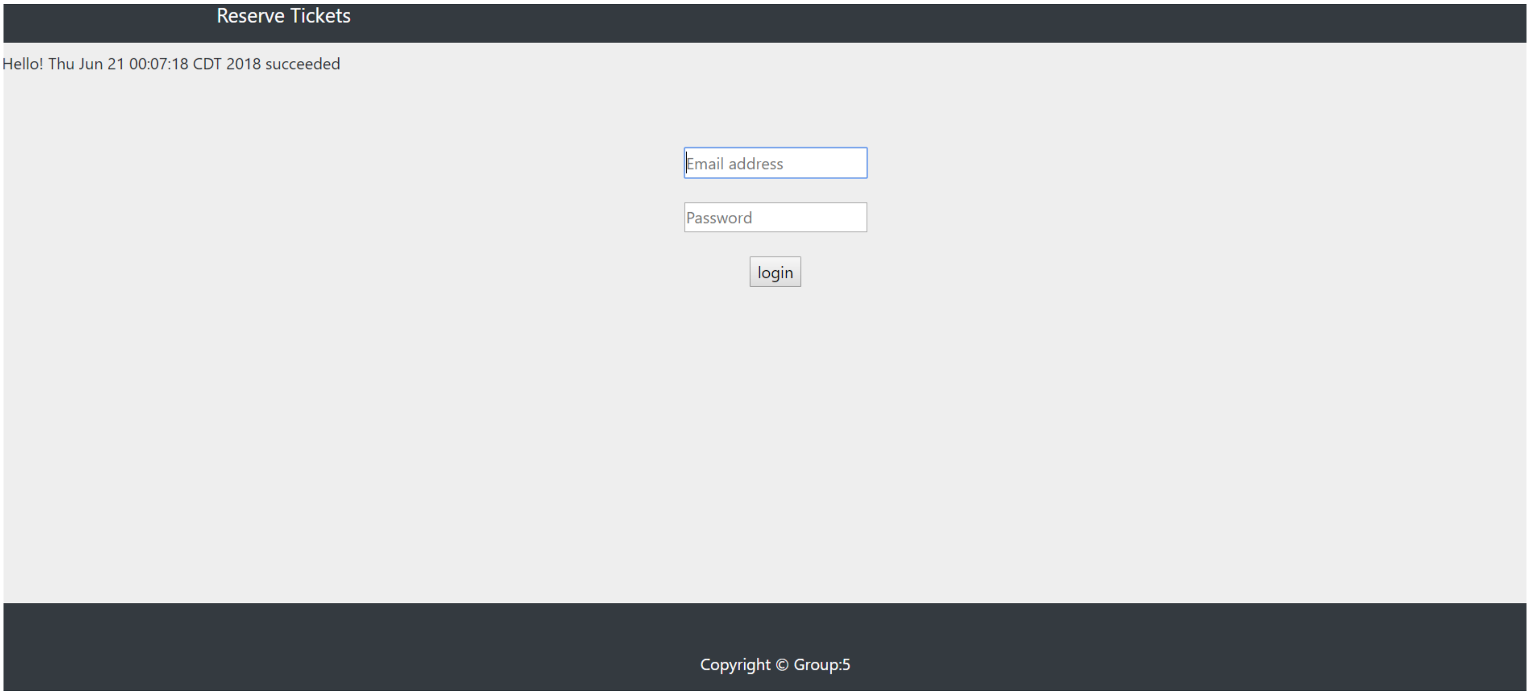
### 5.3. Manage Cart



## 6. Class Diagram

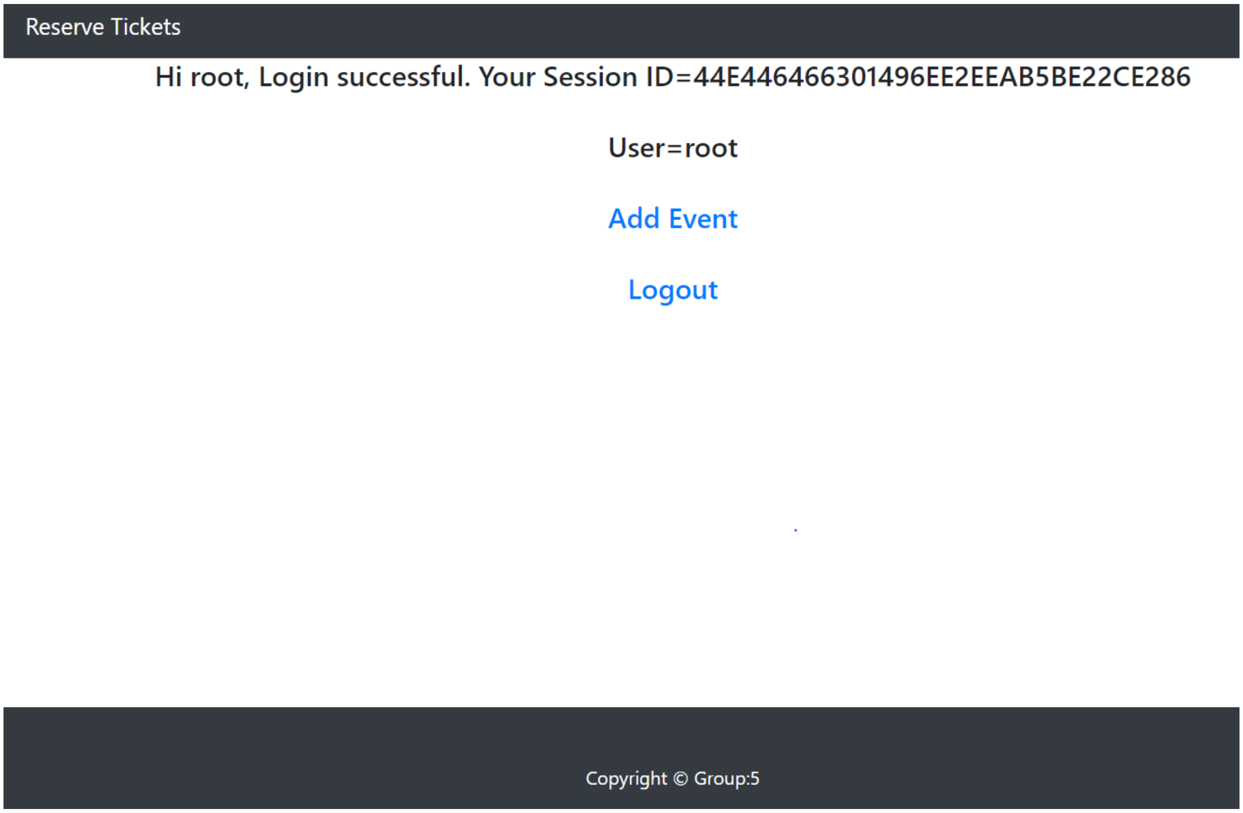

LoginController
+doGet(request: HttpServletRequest, responce: HttpServletResponce)
+doPost(request: HttpServletRequest, responce: HttpServletResponce)
Login
-username: String
-password: String
+getUsername(): String
+setUsername(username: String)
+getPassword(): String
+setPassword(password: String)
SearchController
+doPost(request: HttpServletRequest, responce: HttpServletResponce)
StudentDaoImpl
+login
+validateStudent(login: Login)
Student
-email: String
-userpassword: String
+getEmail(): String
+getUserPassword(): String
+setUserPassword(password: String)
+setEmail(email: String)
DetailsController
+doGet(request, responce)
EventsDaoImp
+getCurrentEvents(): List<Event>
+addEvent(event: Event)
+getRequiredEvent(eventTopic: String): Event
Cart
-eventsSelected: List<Event>
-totalCost: Double
+getEventsSelected(): List<Event>
+setEventsSelected(eventsSelected): List<Event>
+getTotalCost(): Double
+setTotalCost(totalCost: Double)
CartDaoImp
-cart: Cart
+getEventsCount(): Int
+deleteCartItem(evetToBeDeleted: Event)
+addCartItem(evetToBeadded: Event)
+getCartItem(): List<Event>
+calculateOrderTota()
+getOrderTotal(): Double
+checkOut(student: Student, cart: Cart)
Event
-eventID: Int
-topic: String
-eventType: String
-eventDate: Date
-location: String
-price: Float
-eventTime: String
-description: String
+getEventId(): Int
+setEventId(eventId: Int)
+getTopic(): String
+setTopic(topic: String)
+getEvent(): String
+setEventType(eventType: String)
+getEventDate(): Date
+setEventDate(eventDate: Date)
+getLocation(): String
+setLocation(location: String)
+getPrice(): Float
+setPrice(price: Float)
+getEventTime(): String
+setEventTime(eventTime: String)
+getDescription(): String
+setDescription(description: String)
CartController
+doGet(request: HttpServletRequest, responce: HttpServletResponce)
+doPost(request: HttpServletRequest, responce: HttpServletResponce)
+uses
1
1
1
+validate
1
1
+uses
1
+searches
*
1
1
1
+validate
1
+display
*
1
1
1
+fetches
1
+display
1
1
+contains
1
*
+manage
*
+uses
1
1
+resolve
1
+addevents
1
1
+implements
1
1

## 7. Screenshots



Hello! Thu Jun 21 00:07:18 CDT 2018 succeeded

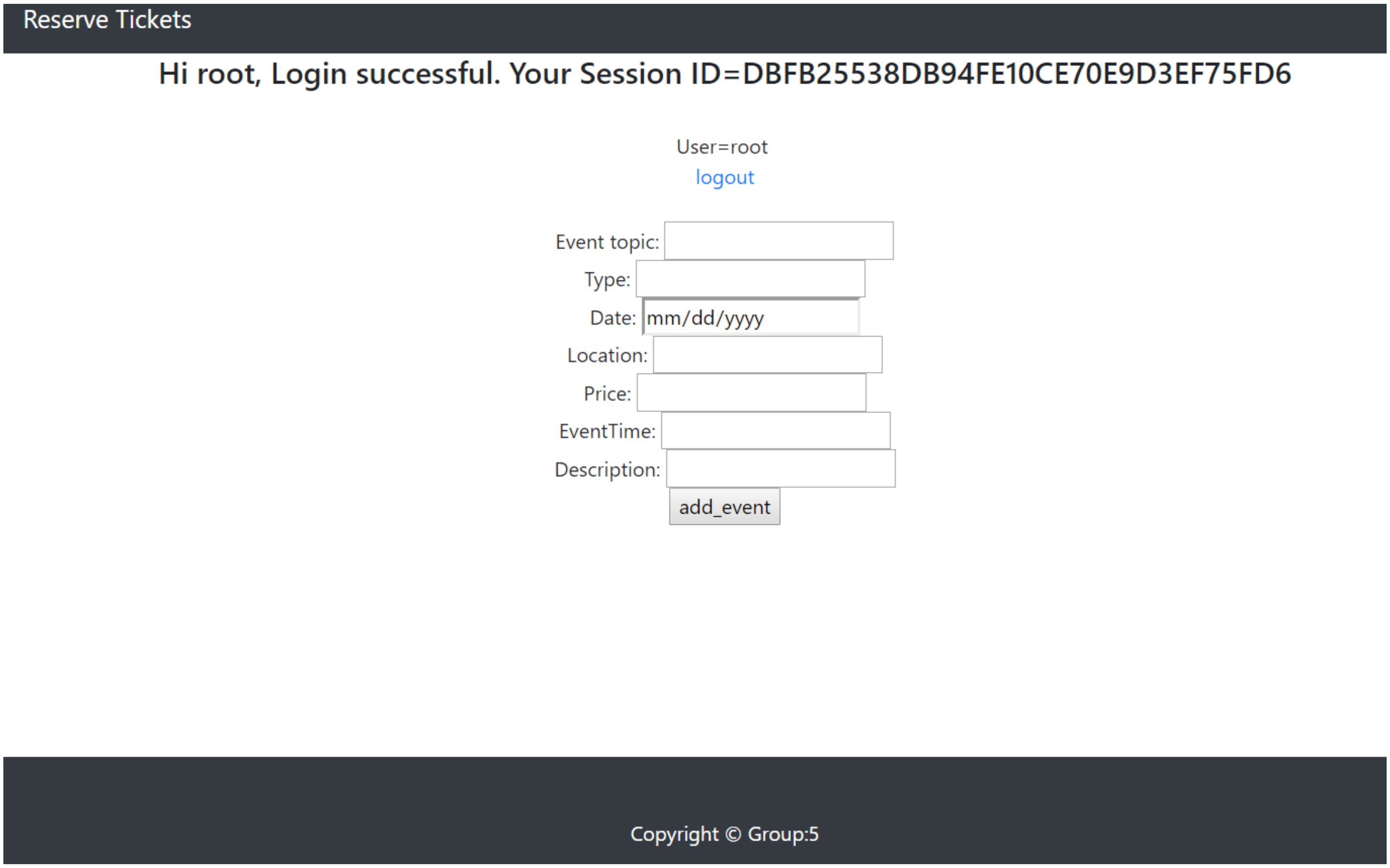
Copyright © Group:5



Hi root, Login successful. Your Session ID=44E446466301496EE2EEAB5BE22CE286

User=root

Copyright © Group:5



Reserve Tickets

## Hi root, Login successful. Your Session ID=DBFB25538DB94FE10CE70E9D3EF75FD6

User=root

logout

Event topic:

Type:

Date: mm/dd/yyyy

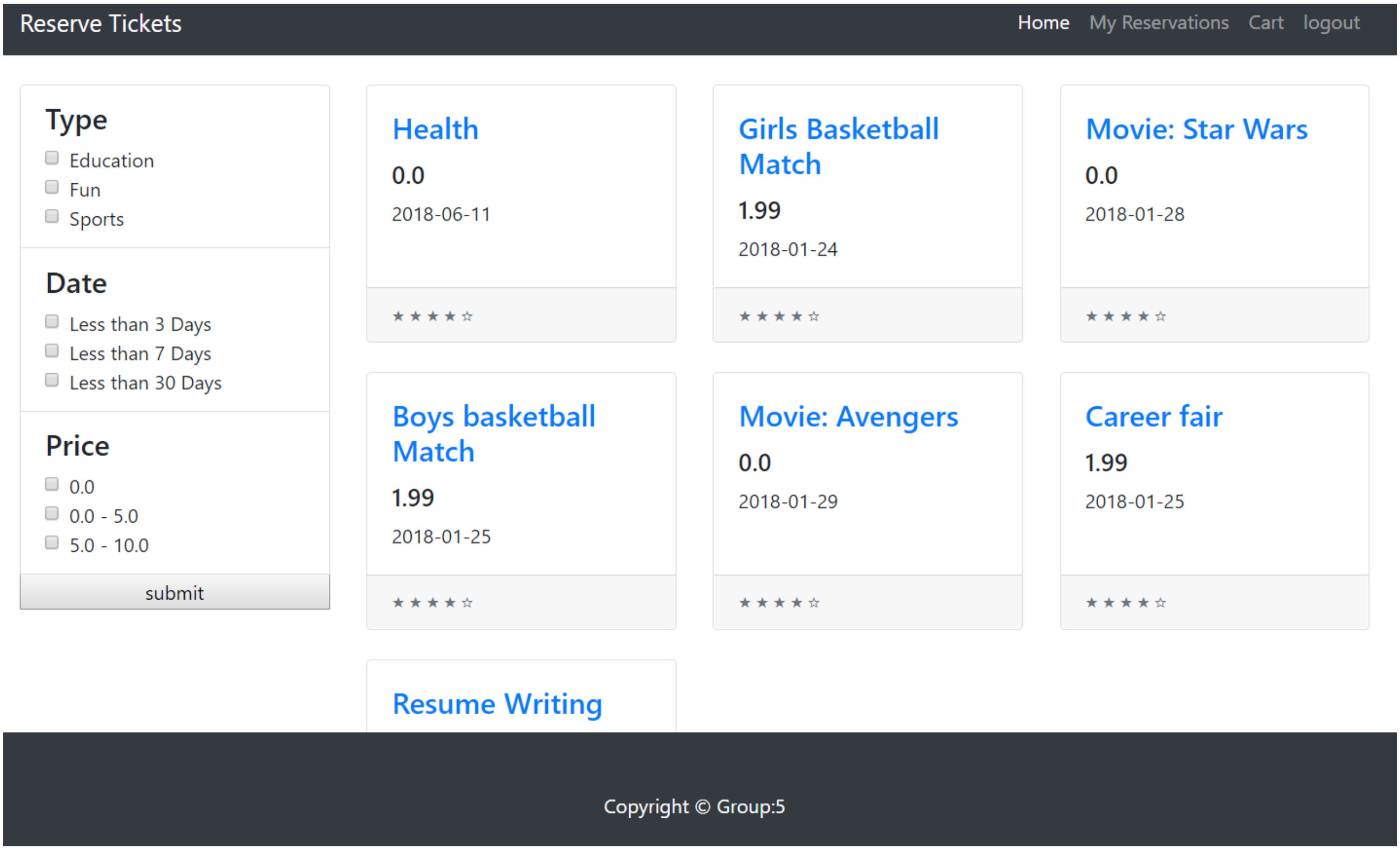
Location:

Price:

EventTime:

Description:

Copyright © Group:5



**Type**

- □ Education
- □ Fun
- □ Sports

**Date**

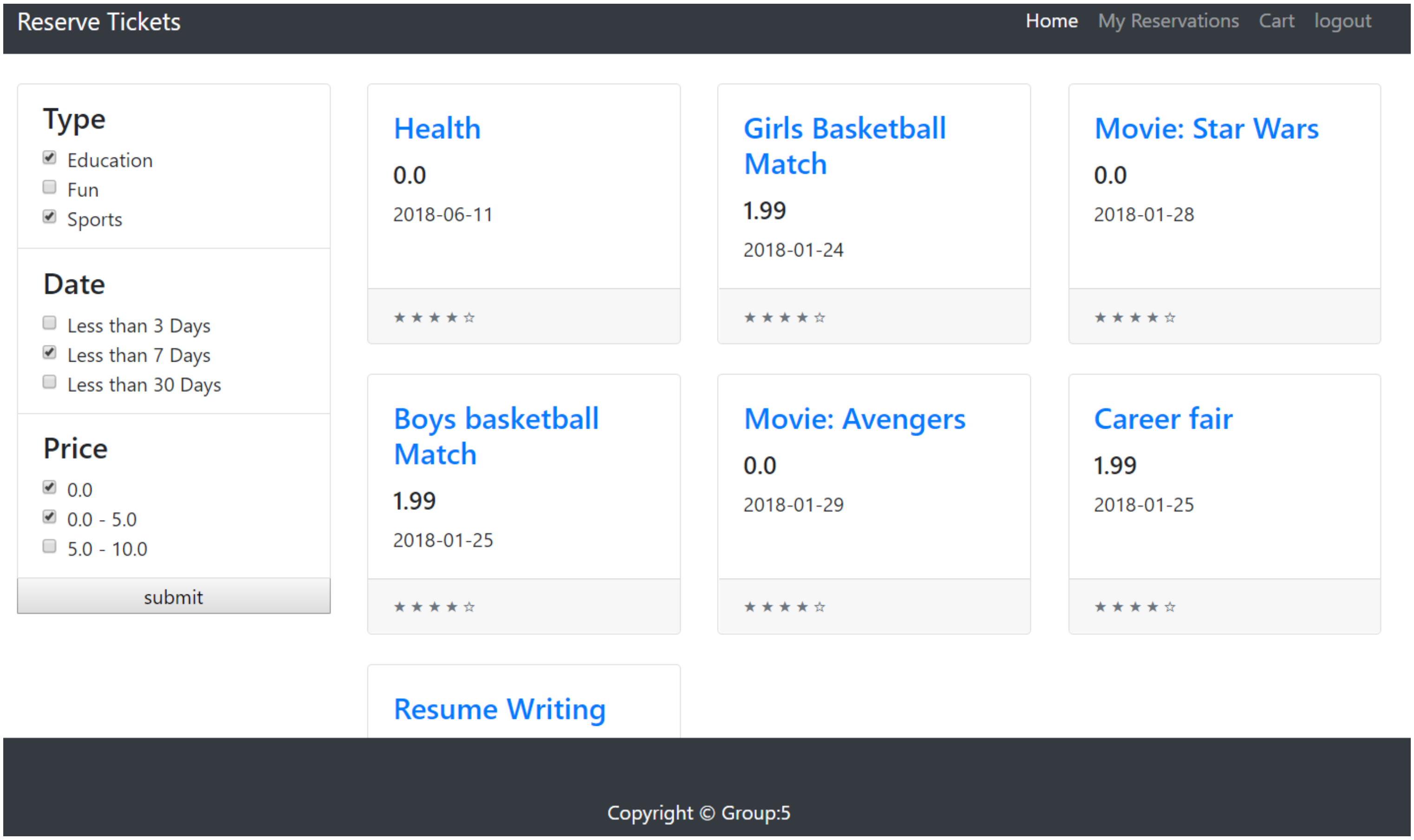
- □ Less than 3 Days
- □ Less than 7 Days
- □ Less than 30 Days

**Price**

- □ 0.0
- □ 0.0 - 5.0
- □ 5.0 - 10.0

| Event | Price | Date | Rating |
|---|---|---|---|
| Health | 0.0 | 2018-06-11 | ★ ★ ★ ★ ☆ |
| Girls Basketball Match | 1.99 | 2018-01-24 | ★ ★ ★ ★ ☆ |
| Movie: Star Wars | 0.0 | 2018-01-28 | ★ ★ ★ ★ ☆ |
| Boys basketball Match | 1.99 | 2018-01-25 | ★ ★ ★ ★ ☆ |
| Movie: Avengers | 0.0 | 2018-01-29 | ★ ★ ★ ★ ☆ |
| Career fair | 1.99 | 2018-01-25 | ★ ★ ★ ★ ☆ |
| Resume Writing | | | |

Copyright © Group:5



Type

☑ Education
☐ Fun
☑ Sports

Date

☐ Less than 3 Days
☑ Less than 7 Days
☐ Less than 30 Days

Price

☑ 0.0
☑ 0.0 - 5.0
☐ 5.0 - 10.0

Health
0.0
2018-06-11
★★★★☆

Girls Basketball Match
1.99
2018-01-24
★★★★☆

Movie: Star Wars
0.0
2018-01-28
★★★★☆

Boys basketball Match
1.99
2018-01-25
★★★★☆

Movie: Avengers
0.0
2018-01-29
★★★★☆

Career fair
1.99
2018-01-25
★★★★☆

Resume Writing

## 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.

**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.<br>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 (Admin) | 1. Admin enters email address and password to login.<br>2. Admin information is fetched from the database. | Validates the admin and admin dashboard is displayed. | Admin can login and the dashboard are displayed. |

| | | | | |
|---|---|---|---|---|
| 3. | Manage Events (Admin) | 1. Admin can add and edit the current list of 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 by selecting multiple filters at a time. |
| 5. | Manage Cart | 1. Student can add required events to the cart.<br>2. A checkout button is present to reserve the tickets. | Reserved events to be added in the database after student adds them to the cart and press checkout. | The cart displays all the selected events by the student and once checkout is done a thank you page is displayed confirming the reservation. |

## 9. Plan: Next Iterations

Iteration 3

Iteraiton1 + Iteration2 + Optimized UI + A receipt is generated at the end of the checkout. The students will receive an email confirmation after the reservation of the event. Database of the students and the current events are modified according to the changed specifications. Unit testing and integration testing is done on the final product.

## 10. Code

https://github.com/Akshay1022/ooad_project