



EventHub

Use Case diagram & Description

PRESENTED BY

MAMBER 1: M. AHMAD HASSAN (LEADER)

MAMBER 2: AHMAD MASOOD

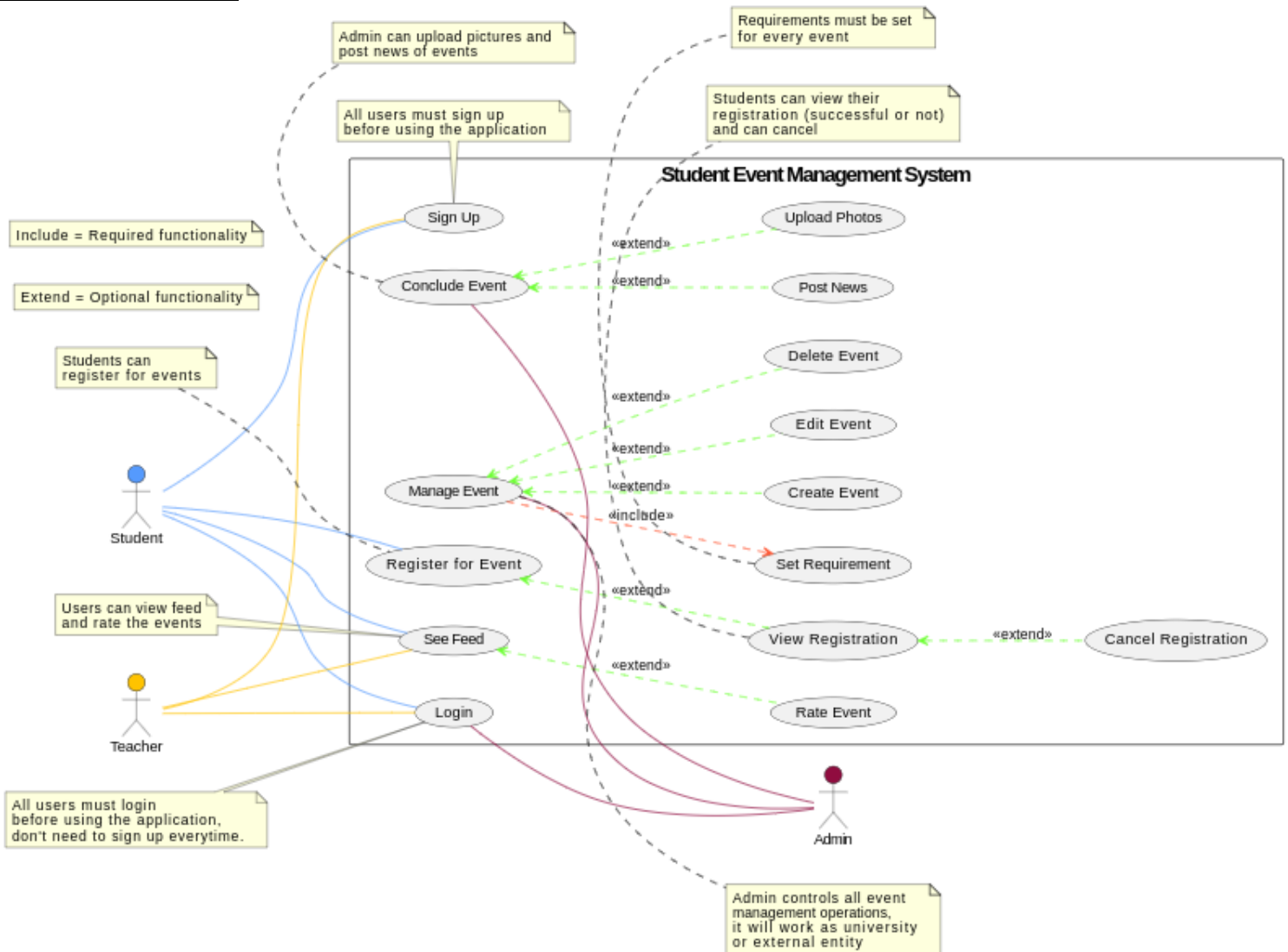
MEMBER 3: M. WALEED

MEMBER 4: TALHA WAINS

SUPERVISED BY

Dr. ONAIZA MAQBOOL

Use Case Diagram:



GITHUB LINK:

<https://github.com/Talha-AppDev/SC-G-15-Case-Study-5.git>

USECASE DESCRIPTION:

BY: M. Waleed

Use Case 1: Conclude Event

Primary Actor:

- Admin (Event Organizer)

Stakeholders and Interests:

- **Student:**
 - Wants to see event outcomes, news, and photos, and provide feedback.
- **Department:**
 - Wants a record of completed events for documentation and future planning.
- **Event Organizer (Admin):**
 - Wants to officially mark the event as completed and share updates.

Preconditions:

- The event must have taken place.
- The admin must be logged into the system.

Success Guarantee (Postconditions):

- The event status is updated to "Completed."
- Photos and news are uploaded for students and teachers to view.
- The system allows students and teachers to provide ratings and feedback.

Inputs:

- Event ID
- Post-event summary
- Key highlights
- Photos/images
- Event news updates

Outputs:

- Event marked as "Completed."
- Uploaded images and news displayed to students and teachers.
- Rating and feedback section enabled for users.

Main Success Scenario (Basic Flow):

1. The admin logs into the system.
2. The admin navigates to the event management section.
3. The admin selects the complete event.
4. The system prompts post-event details (summary, key highlights, photos).
5. The admin uploads event news, images, and key outcomes.
6. The system updates the event status to "Completed."
7. The system enables the rating and feedback feature for students and teachers.

Alternate Scenario (Extensions):

- ***a. System Failure at Any Time:**
 - The software saves existing data automatically.
 - When the system is reopened, it resumes from the last step saved.
- **4a. The admin does not have all the post-event details:**
 - The system allows saving progress and updating later.
- **5a. Upload fails due to file size or format issues:**
 - The system notifies the admin and prompts for a compatible file.

Special Requirements:

- The application should be developed in **Java**.
- Student registration should be easily visible, using a readable font size (e.g., 12pt).

Frequency of Occurrence:

- Nearly continuous.

Open Issues:

- Open for feedback to improve the use of case text.
 - Consider additional input and output details for better usability.
-

BY: M. AHMAD HASSAN

Use Case 2: Register for Event

Primary Actor:

- Student

Stakeholders and Interests:

- **Student:**
 - Wants to register easily for events they are eligible for and view their registration details.
- **Department:**
 - Wants an organized system to manage student participation.
- **Event Organizer:**
 - Wants to monitor student registrations and ensure participant limits are maintained.

Preconditions:

- The student must be logged into the system.
- The event must be open for registration.

Success Guarantee (Postconditions):

- The student is successfully registered for the event.
- The system updates the event's participant list.

Inputs:

- Event ID
- Student ID
- Eligibility verification data
- Registration confirmation request

Outputs:

- Confirmation message of successful registration
- Updated participant list
- Error message (if registration fails due to eligibility or full capacity)

Main Success Scenario (Basic Flow):

1. The student logs into the system.
2. The student navigates the events section and selects an event.
3. The system displays event details and eligibility criteria.
4. The student confirms eligibility and proceeds with registration.
5. The system registers the student and updates the participant list.
6. The student receives confirmation of successful registration.

Alternate Scenario (Extensions):

- ***a. System Failure at Any Time:**
 - The software saves existing registration data automatically.
 - When the system is reopened, it resumes from the last step saved.
- **1a. Student is not eligible for the event:**
 - The system displays error messages and does not allow registration.
- **5a. Event registration limit is reached:**
 - The system notifies the student that registration is full and prevents further registration.

Special Requirements:

- The application should be developed in **Java**.
- Student registration should be easily visible, using a readable font size (e.g., 12pt).

Frequency of Occurrence:

- Nearly continuous.

Open Issues:

- Open for feedback to improve the use of case text.
- Consider additional input and output details for better usability.

BY: AHMAD MASOOD

Use Case 3: Manage Event

Primary Actor:

- Admin

Stakeholders and Interests:

Admin:

- Needs to create, update, delete events, and set event requirements efficiently.

Inputs:

- Event details (name, description, location, date, time, participant limit)
- Edited event details
- Confirmation for deletion
- Requirement details (prerequisites, maximum participants)

Outputs:

- Confirmation of event creation
- Updated event details
- Deletion confirmation
- Error messages if required fields are missing or unauthorized access is attempted

Preconditions:

- The admin must be logged into the system.
- Only Admins can create, edit, delete, and set requirements for events.

Postconditions:

- The event is successfully created, updated, or deleted.
- Event requirements are correctly configured and stored.

Main Success Scenario (Basic Flow):

1. The admin navigates to the "*Manage Event*" section
2. The system displays options for managing events
3. The admin selects "*Create Event*"
4. The system prompts for event details (name, description, location, date, time, participant limit, etc.)
5. The admin enters the required details and submits the form
6. The system validates the input and saves the event
7. The system confirms event creation and adds it to the event list
8. The admin selects an existing event to "*Edit*"
9. The system displays the event's details
10. The admin modifies the necessary fields and submits changes
11. The system validates and updates the event information
12. The admin selects an event to "*Delete*"
13. The system prompts confirmation.
14. Upon confirmation, the system removes the event.
15. The admin selects "*Set Event Requirements*"
16. The system displays available requirement fields (e.g., prerequisites, maximum participants, category).
17. The admin enters or updates the requirements.
18. The system validates and saves the event requirements.

Alternate Scenario:

- ***a. System Failure at Any Time:**
 - The software saves existing data automatically.
 - When the system is reopened, it resumes from the last step saved.
- **05a:** If required fields are missing, the system prompts the admin to complete them.
- **10a:** If required fields are missing, the system prompts the admin to complete them.

Special Requirements:

- The application should be developed in **Java**.
- Event details should be stored securely.

Frequency of Occurrence:

- Nearly continuous.

Open Issues:

- Open for feedback to improve the use of case text.
- Consider additional input and output details for better usability.

BY: TALHA WAINS

Use Case 4: See Feed

Use Case Title:

- View Event Feed (with Extended Event Rating)

Primary Actors:

- Student
- Teacher

Stakeholders & Interests:

• Students & Teachers:

- Can see the event updates (news and photos).
- Provide feedback by rating events.

• Admin:

- Upload event-related news and images.
- Ensure the event feed is uploaded with related photos & news.

Preconditions:

- The user is authenticated (logged in).
- The Admin has successfully uploaded event updates (news and/or photos) so that content is available for display.

Postconditions:

- The user successfully views the feed, photos and news of event.
- Any rating submitted by the user is stored and the event's overall rating is updated accordingly.

Input:

User Action: The user selects the "View Event Feed" option (triggering the use case).

Optional Rating Input: If the user interacts with the rating feature, provide a rating value (e.g., via stars or emojis).

Output:

Displayed Event Feed: The system shows a feed with event news/photos.

Rating Confirmation: If a user rating is submitted, the system records it and immediately updates the event's overall rating.

Main Success Scenario (Basic Flow):

1. The user logs into the system using valid credentials.
2. The user navigates to the "Event Feed" section by clicking on the option in the main menu.
3. The system displays the event feed with available content and provides an option to rate events.
4. The user gives rating (e.g. by star system or emojis method).
5. The system saves the submitted rating by the user.
6. The system immediately updates the event feed by doing calculations to update the rating and displays the new overall rating.

Alternate Flow:

- ***a. System Failure at Any Time:**
 - The software saves existing registration data automatically.
 - When the system is reopened, it resumes from the last step saved.
- **3a.** If no event content is available (i.e., the admin has not uploaded content), the system displays a message such as "No event updates available!"
- **3b.** The system disables (unclickable) the rating option.

Exception Flows:

Exception Flow 1 – Unauthenticated Access:

- **1a.** If the user is not logged in, the system prevents access to the event feed and rating features.
- **1b.** The system directs the user to Sign Up first.

Special Requirements:

- Only authenticated users can access the event feed & give rating.
- The event feed will display the event content that has been uploaded by the Admin.
- Ratings must be saved, and the updated event rating must be visible to users immediately.

Frequency of Occurrence:

- Opened multiple times as users routinely check for event updates & submit ratings.

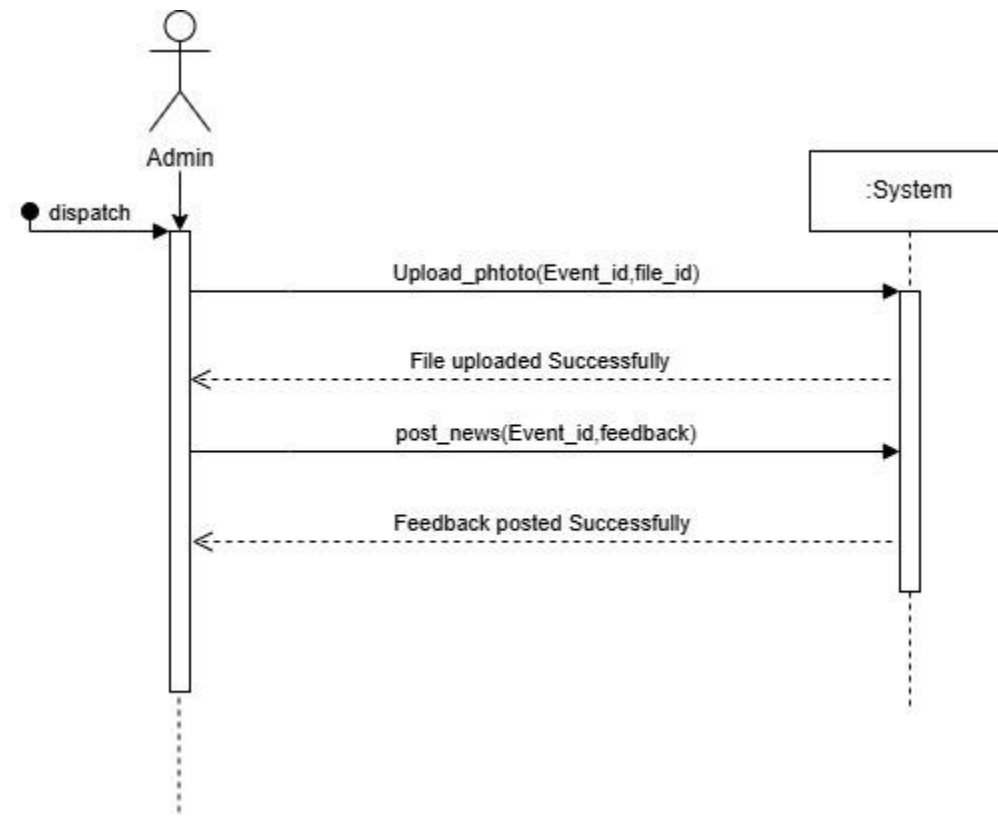
Open Issues:

- Open for feedback to improve the use of case text.
- Consider additional input and output details for better usability.

Class System Sequence Diagram:

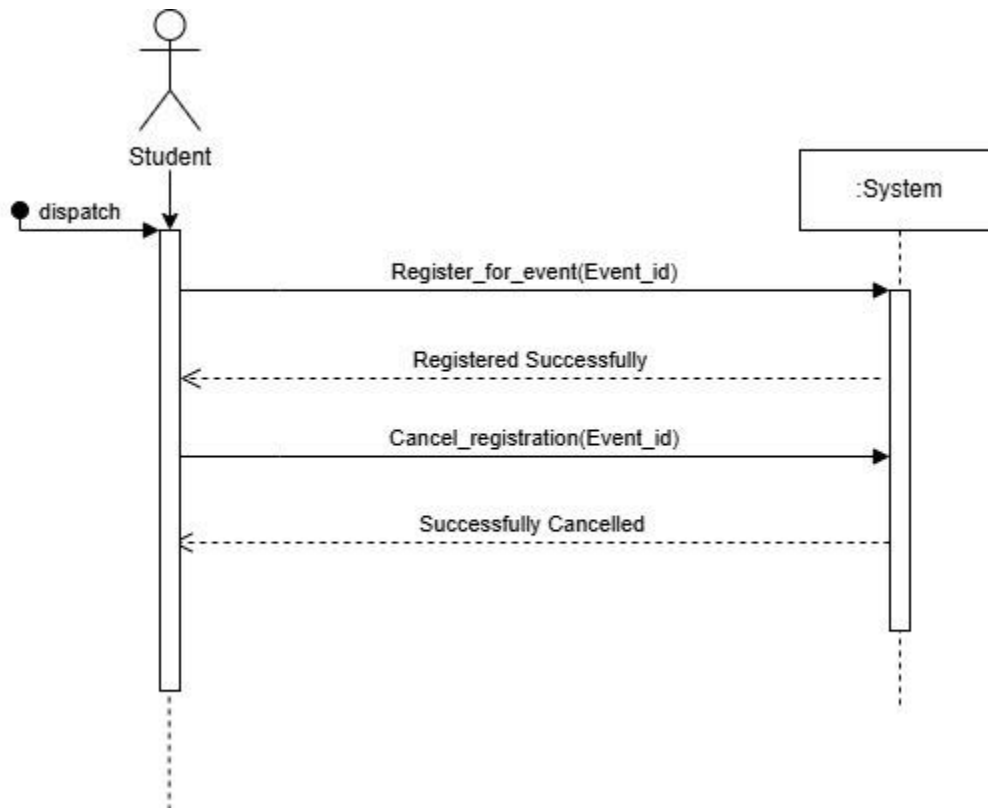
BY: M. Waleed

Use Case: Conclude Event



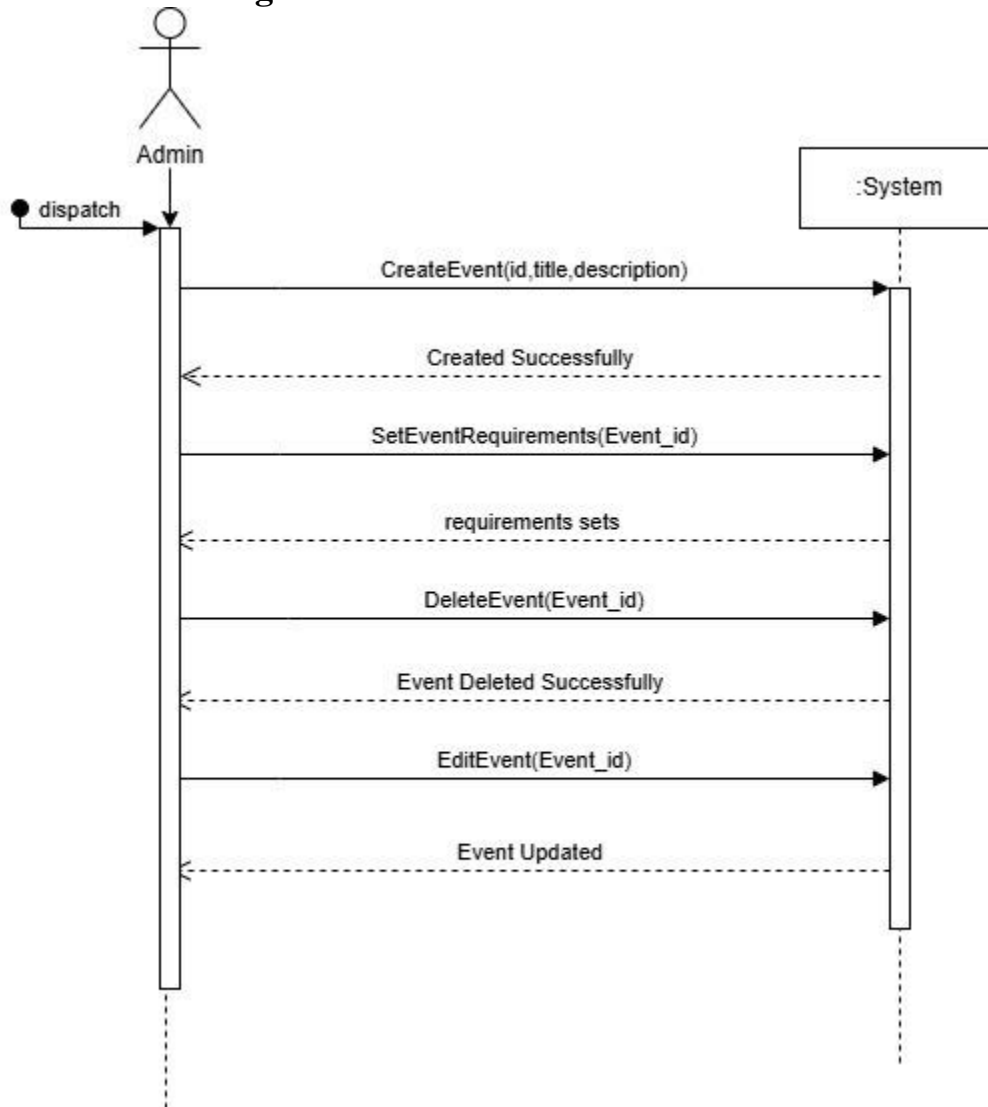
BY: M. AHMAD HASSAN

Use Case: Register for Event



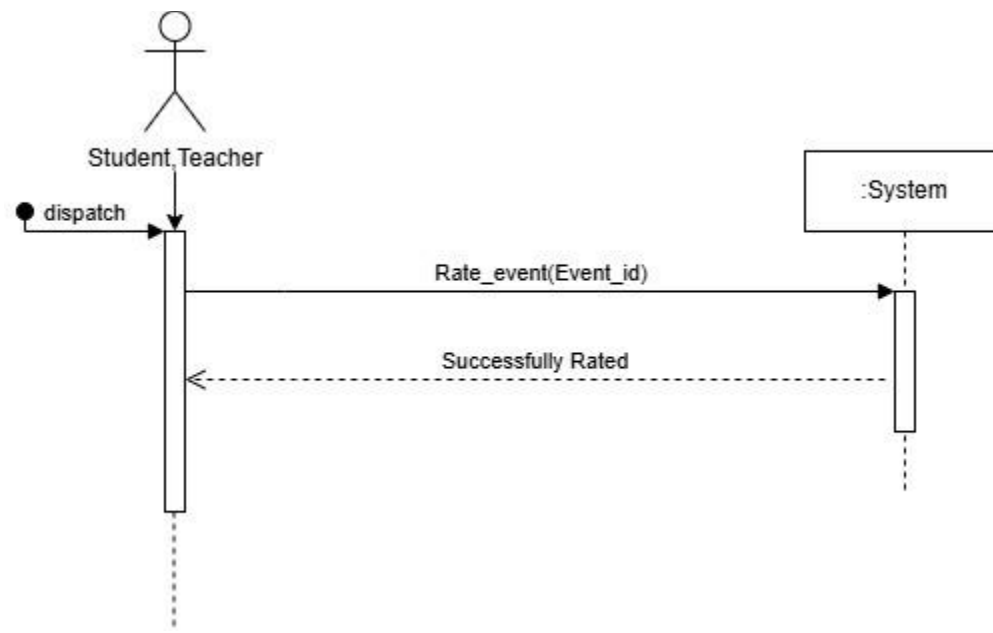
BY: AHMAD MASOOD

Use Case: Manage Event



BY: TALHA WAINS

Use Case: See Feed



Domain Model:

