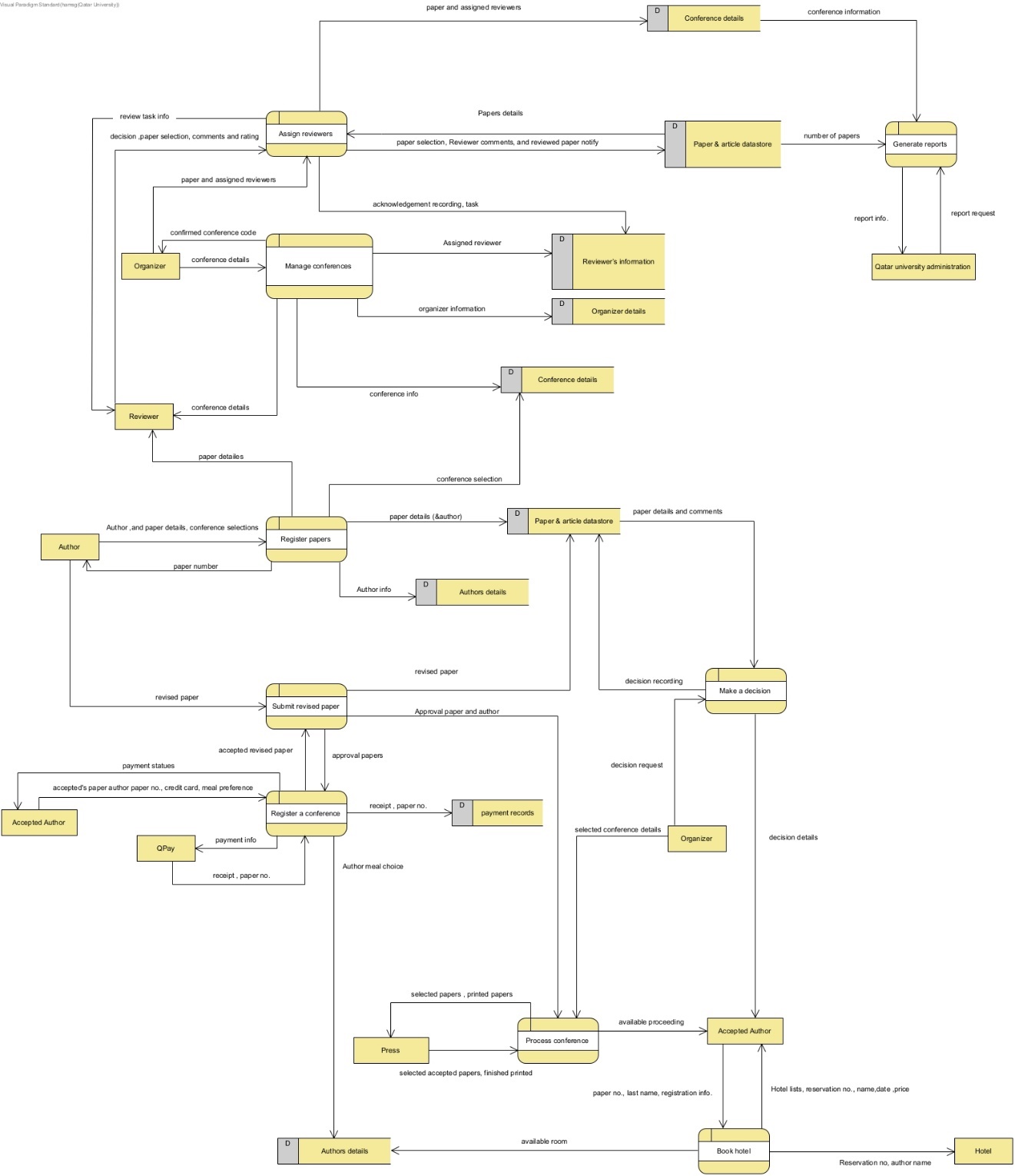
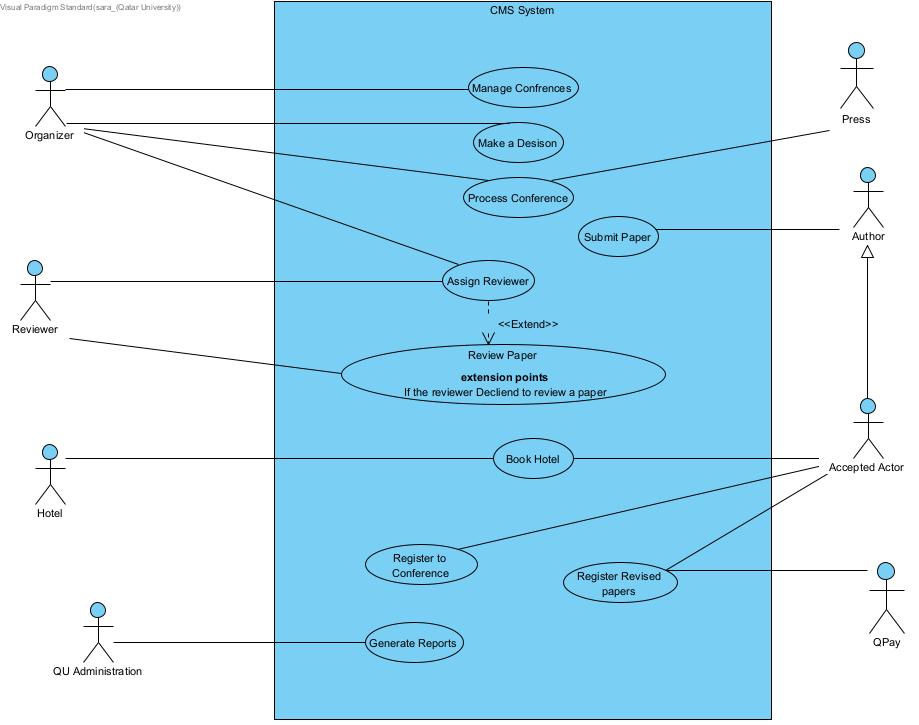
**Task 1 (part a):**

**A data flow diagram (DFD) of the system: Include major processes, data storage/data files, data flows, and external entities (*15%*):**

****

**Task 1 (part b):**

**A use case diagram of the system (*15%*)*:***

****

**Task 1 (part c):**

**Explain which of these two you prefer best to analyze a system and why:**

We would choose to use the use case diagram while constructing data flow and use case diagrams to analyze the Conference Management System (CMS). The nature of the system's functioning and user interactions is the cause of this preference. The system's features (submitting papers, managing conferences, evaluating papers, booking rooms, and generating reports) are best illustrated through use case diagrams since they show how different players (organizers, authors, reviewers, and administrators) interact with them. They give these interactions a clear visual representation, which makes it simpler to comprehend how the system behaves from the standpoint of the user.

Data flow diagrams, on the other hand, are frequently more suited for illustrating the movement of data and processes within a system, which might be more difficult to illustrate in this situation due to the variety of user actions and interactions. Use case diagrams enable a more straightforward depiction of the user roles and how they interact with the system's capabilities thanks to their actor-centered approach. This clarity is also helpful when dealing with stakeholders because it makes sure that everyone knows how the system works and how users interact with it.

**Task 2:**

**Develop *use case specifications* for the most complex and key 4 use cases selected from your use case diagram.**

# Use cases specification.

|  |  |  |
| --- | --- | --- |
| **Use case Id:** UC001 | **Manage Conference** | |
| **Brief Description** | This use case describes the process of an organizer registering a conference with the system. The system checks if a conference with the same name already exists, records the conference details, associates reviewers with the conference, generates a conference code, and confirms the registration to the organizer. | |
| **Primary actors** | Organizer | |
| **Trigger(s)** | The organizer logs in to register for a conference. | |
| **Preconditions:**  The organizer must be logged in.  The organizer's details must be available in the system.  Reviewer details and their expertise may or may not be already available in the system. | | |
| **Post-conditions:**   * The new conference was successfully registered in the system. * Reviewers were associated with the conference. * The organizer received a confirmation message with the conference code. | | |
| **Normal Scenario** | | |
| Actor Action | | System Response |
| 1.The organizer logs in to the system. | | 2. validates the organizer's login credentials (see 2a) |
| 3. The organizer enters conference details | | 4. checks if a conference with the same name already exist (see 4a,b) |
|  | | 5. records the new conference details and associates them with the organizer. |
|  | | 6. generates a unique conference code for the newly registered conference. |
|  | | 7. sends a confirmation message to the organizer, including the conference code and registration details. |
| **Alternative flows:**  2.a. If the organizer enters incorrect login credentials, displays an error message and the use case terminates.  4.a.If a conference with the same name does not exist, proceed to the next step.  4.b. If a conference with the same name does not exist, return to step 3 | | |

|  |  |  |
| --- | --- | --- |
| **Use case Id:** UC002 | **Register for conference** | |
| **Brief Description** | This use case registers accepted authors for a conference, verifying paper acceptance, collecting payment information, meal preferences, and processing payments through QPay. | |
| **Primary actors** | Accepted Author, QPay | |
| **Trigger(s)** | Author provides the paper number and credit card details | |
| **Preconditions:**   * Author is logged in * Author’s paper has been accepted * Author submitted the revised paper | | |
| **Post-conditions:**   * Author’s meal preferences stored. * Author informed about payment outcome. | | |
| **Normal Scenario** | | |
| Actor Action | | System Response |
| 1. Author provides the paper number and credit card details | | 1. Check if the paper number corresponds to an accepted paper. ( see 2.a. ) |
| 1. Author selects their meal preferences for the conference. | | 1. Store the meal preference. |
|  | | 1. Forward card details and paper number to the QPay. |
| 1. QPay processes the payment and sends the payment outcome. | |  |
|  | | 1. Inform the author the payment was successful. (see 7.a.) |
| **Alternative flows:**  **2.a.** If the paper number does not exist or is not accepted, generate an error message.  7.a. If the payment is not accepted, send an error message. | | |

|  |  |  |
| --- | --- | --- |
| **Use case Id:** UC003 | **Assign Reviewer** | |
| **Brief Description** | It manages the process of assigning reviewers to the submitted paper for the conference. | |
| **Primary actors** | Organizer, Reviewer | |
| **Trigger(s)** | The organizer assign reviewer to a submitted paper. | |
| **Preconditions:**  Have the conference deadline.  Have the submitted papers. | | |
| **Post-conditions:**  The reviewer assigned to papers.  The rating and comments were attached with the submitted papers. | | |
| **Normal Scenario** | | |
| Actor Action | | System Response |
| 1. The organizer assign reviewer to a submitted paper. | | 2. Retrieve the conference and the submitted paper details. |
|  | | 3. Search for the reviewers who have less than three papers to review and assign them to the papers. |
|  | | 4. Notify the author about the assigned task and the deadline to submit the feedback and the review result. |
| 5. The reviewer send an acknowledgment. | | 6. Records the acknowledgment and the reviewer’s details. |
| 7. The reviewer selects a conference to see the assigned paper. | | 8. Showes the paper’s abstract. |
| 9. The reviewer send her/ his option (accept or decline). | | 10. Record the reviewer comments and rating.  See (10.a) |
|  | | 11. Mark the paper as reviewed with the review date. |
| **Alternative flows:**  10.a. If the reviewer declines the assigned task, release the reviewer from the list, and find another for the paper. | | |

|  |  |  |
| --- | --- | --- |
| **Use case Id:** UC005 | **Submit Paper** | |
| **Brief Description** | It manages submitting the papers by author | |
| **Primary actors** | Author | |
| **Trigger(s)** | Author provides his details | |
| **Preconditions:** There is a conference that their deadline didn’t pass , Paper not submitted to another conference | | |
| **Post-conditions: A paper submitted to a conference** | | |
| **Normal Scenario** | | |
| Actor Action | | System Response |
| 1-Author provides his details | | 2-log in the author see(2 .a) |
|  | | 3.find author details and find all the conferences and display them |
| 4. Author select the conference | | 5.record the selection |
| 6.Author provides paper details (paper name, names of all authors, abstract of the paper, keywords) and upload the paper. | | 6.Store all the paper details with the selected conference recorded earlier. |
|  | | 7. Produce a paper number for the author |
|  | | 8.Register the paper see(8 .a) |
| **Alternative flows:**  **2 .a if there is records for this author the system creates a login name and password for the author.**  **8 .a if this is the fourth paper for the author in the same conference the system terminates the session without registering the paper.** | | |

**Task 3 part(a):**

**Complete the *design class diagram* of the entire system with all required classes, their attributes and types, relationships (aggregation, generalization, association) with multiplicity where applicable, methods with major parameters, and visibility of attributes and methods.**

**A diagram of a computer flowchart

Description automatically generated**

**Task 3 part(b):**

**Explain briefly how you applied design principles in your class diagram. In your explanation, give examples from your class diagram.**

1. **Inheritance**:

Example: The Organizer, Author, Reviewer, and Administrator classes all inherit from the base User class. This inheritance allows them to inherit common user-related attributes (the user and password) and methods (log in).

1. **Encapsulation**:

The class diagram shows that attributes such as userId, username, and password are encapsulated within the User class. Access to these attributes is restricted through methods like login (username, password) to ensure secure and controlled access. This encapsulation helps maintain data integrity and security.

1. **Association**:

The class diagram illustrates various associations between classes to represent the relationships described in the system's requirements.

* The association between the Organizer and Conference classes is denoted as "Organizes (1) Conference" This indicates that an organizer can be associated with one or more conferences.
* Includes (1 to n) Venue,
* HasReviewers (0 to n) Reviewer,
* AcceptsPapers (0 to n) Paper,
* GeneratesProceedings (1) Proceedings

1. **Multiplicity:**

The "Reviews (2 to 3) Paper" association with the Reviewer class indicates that a reviewer can be assigned to review a minimum of 2 and a maximum of 3 papers for a particular conference.

**Any assumptions you made regarding the system description must be explained:**

**Comments regarding assigning methos and attributes:**

1. Author: any author can register for a conference, but only the accepted Author (isAccept) can make reservation and choose a meal.
2. Assigning paper no. is done by the controller class and not the author.
3. Assigning reviewers to review papers is handled by the controller class.
4. A reviewer can accept a paper and the acknowledgment is sent automatically, otherwise it considers declined.
5. For accepetConditional:String

For the paper accepted conditionally, we first check the isAccepted if it is true,

We read the accepetConditional attribute to get the condition, if it is null this paper is accepted, otherwise the String is representing the condition of acceptance.

1. Consider conference no. and code to be same to reduce redundancies.
2. Assumed that all attributes have setter and getter.