1. Simplyfying the use case diagram but providing the absolute required fields

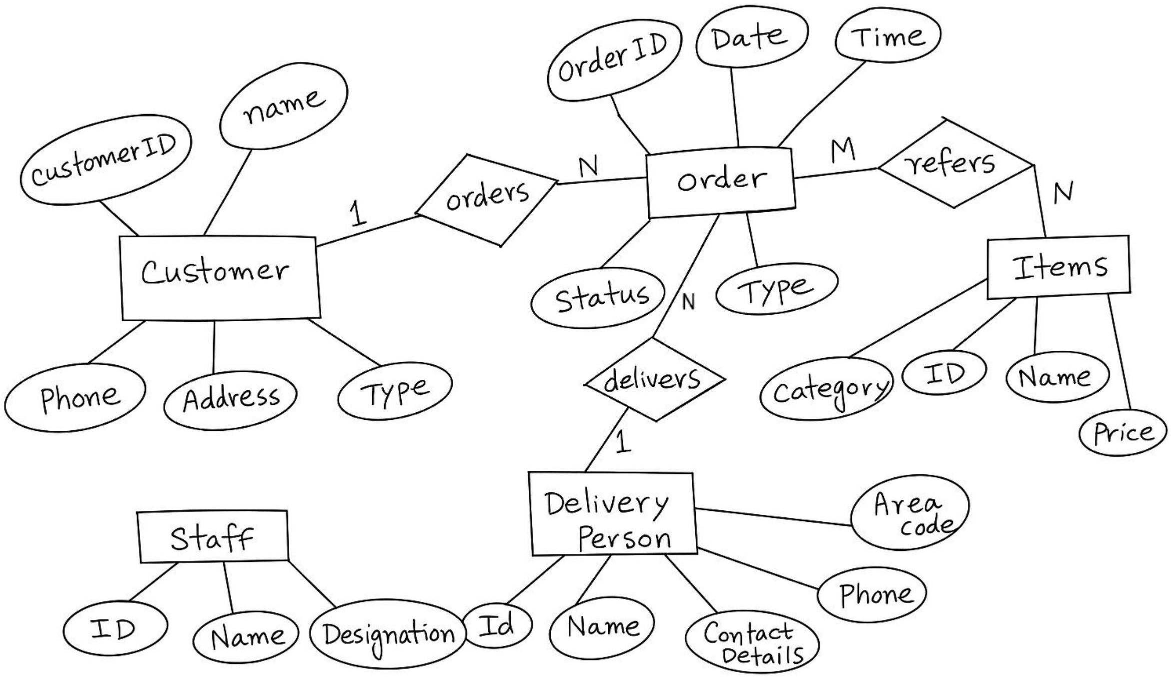
2. srs doc depends on use case diagram modifying according to that

3. Adding input output to srs

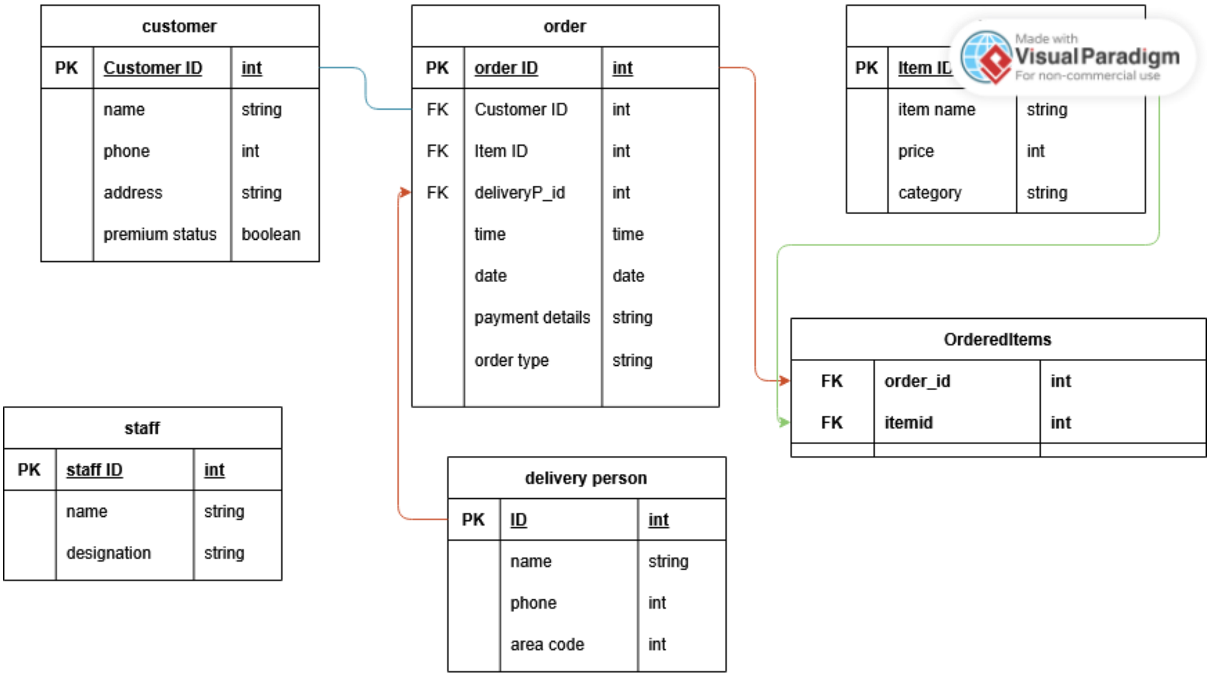
4. Modifying the er diagram

Problem 1:

Er Diagram:



Relational Schema:



customer		
PK	Customer ID	int
	name	string
	phone	int
	address	string
	premium status	boolean

order		
PK	order ID	int
FK	Customer ID	int
FK	Item ID	int
FK	deliveryP_id	int
	time	time
	date	date
	payment details	string
	order type	string

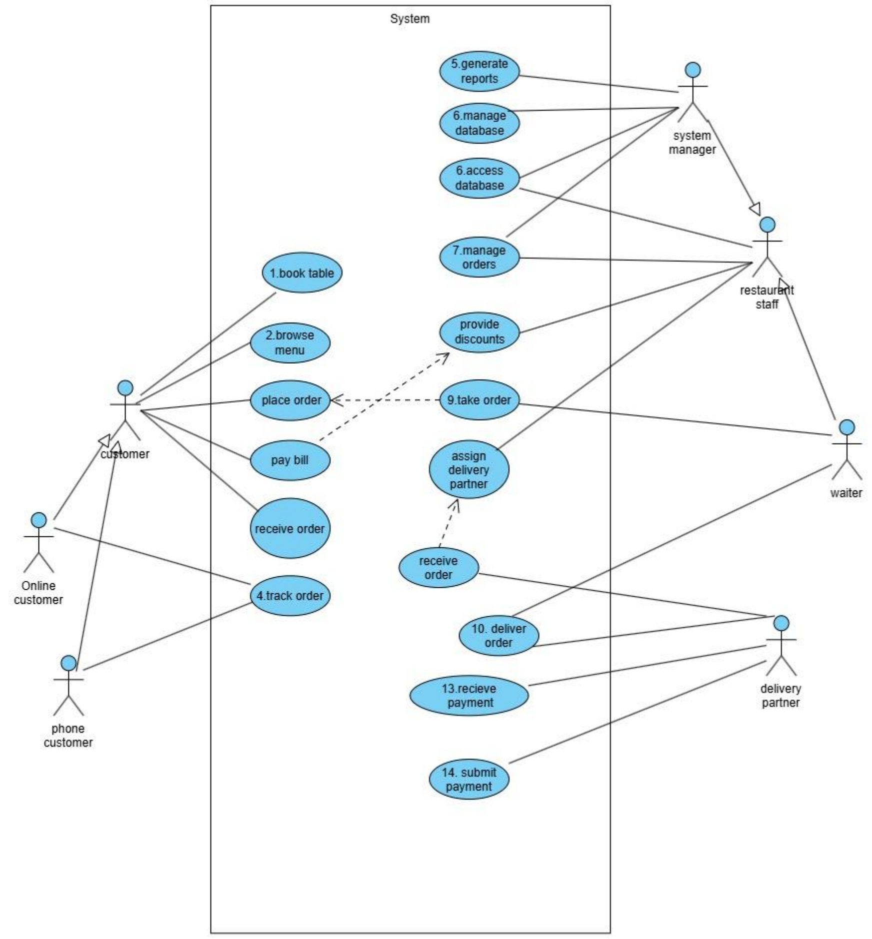
PK	Item ID	
	item name	string
	price	int
	category	string

OrderedItems		
FK	order_id	int
FK	itemid	int

staff		
PK	staff ID	int
	name	string
	designation	string

delivery person		
PK	ID	int
	name	string
	phone	int
	area code	int

Use Case Diagram:



Functional Requirements (SRS)

1. Food Menu Maintenance

- **FR1.1:** The system shall allow an administrator to add new food or beverage items.

    - **Input:** Item name, category, price.
    - **Output:** Confirmation message with Item ID.
- **FR1.2:** The system shall allow an administrator to update existing item details.
    - **Input:** Item ID, updated name/category/price.
    - **Output:** Confirmation of successful update.
- **FR1.3:** The system shall allow an administrator to delete items no longer offered.
    - **Input:** Item ID.
    - **Output:** Confirmation of successful deletion.

2. Placing Orders

- **FR2.1:** The system shall allow customers to place orders online, by phone, or at the premises.
    - **Input:** Customer credentials (login or phone number), selected items and quantities, order type (online/phone/in-premises).
    - **Output:** Confirmation message with Order ID and order summary.
- **FR2.2:** The system shall validate that selected items exist in the menu.
    - **Input:** List of Item IDs and quantities.
    - **Output:** Validation success or error indicating invalid Item ID.

3. Order Recording

- **FR3.1:** The system shall record details of each order.
    - **Input:** Order ID, Customer ID, list of Item IDs with quantities, date, time, order type.
    - **Output:** Stored order record in database.
- **FR3.2:** The system shall record payment details for each order.
    - **Input:** Order ID, payment mode, payment transaction details.

    - **Output:** Stored payment record linked to Order ID.

4. Delivery Partner Assignment

- **FR4.1:** The system shall assign a delivery partner to each delivery order based on area code.
    - **Input:** Order ID, customer address (area code).
    - **Output:** Assignment confirmation with Delivery Person ID.
- **FR4.2:** The system shall prevent assignment of a delivery partner outside their area code.
    - **Input:** Delivery Person ID, customer area code.
    - **Output:** Error if area codes do not match.

5. Delivery Recording

- **FR5.1:** The system shall record delivery details for each order.
    - **Input:** Delivery ID, Order ID, Delivery Person ID, delivery date and time, delivery status (Pending/Completed/Cancelled).
    - **Output:** Stored delivery record in database.

6. Customer Management

- **FR6.1:** The system shall allow registration of new customers.
    - **Input:** Name, phone, address, email (optional).
    - **Output:** Confirmation with Customer ID.
- **FR6.2:** The system shall maintain and update customer premium status.
    - **Input:** Customer ID, premium status flag (true/false).
    - **Output:** Confirmation of status change.
- **FR6.3:** The system shall retrieve customer order history.

- **Input:** Customer ID.
- **Output:** List of past orders with dates, items, totals.

7. Staff Management

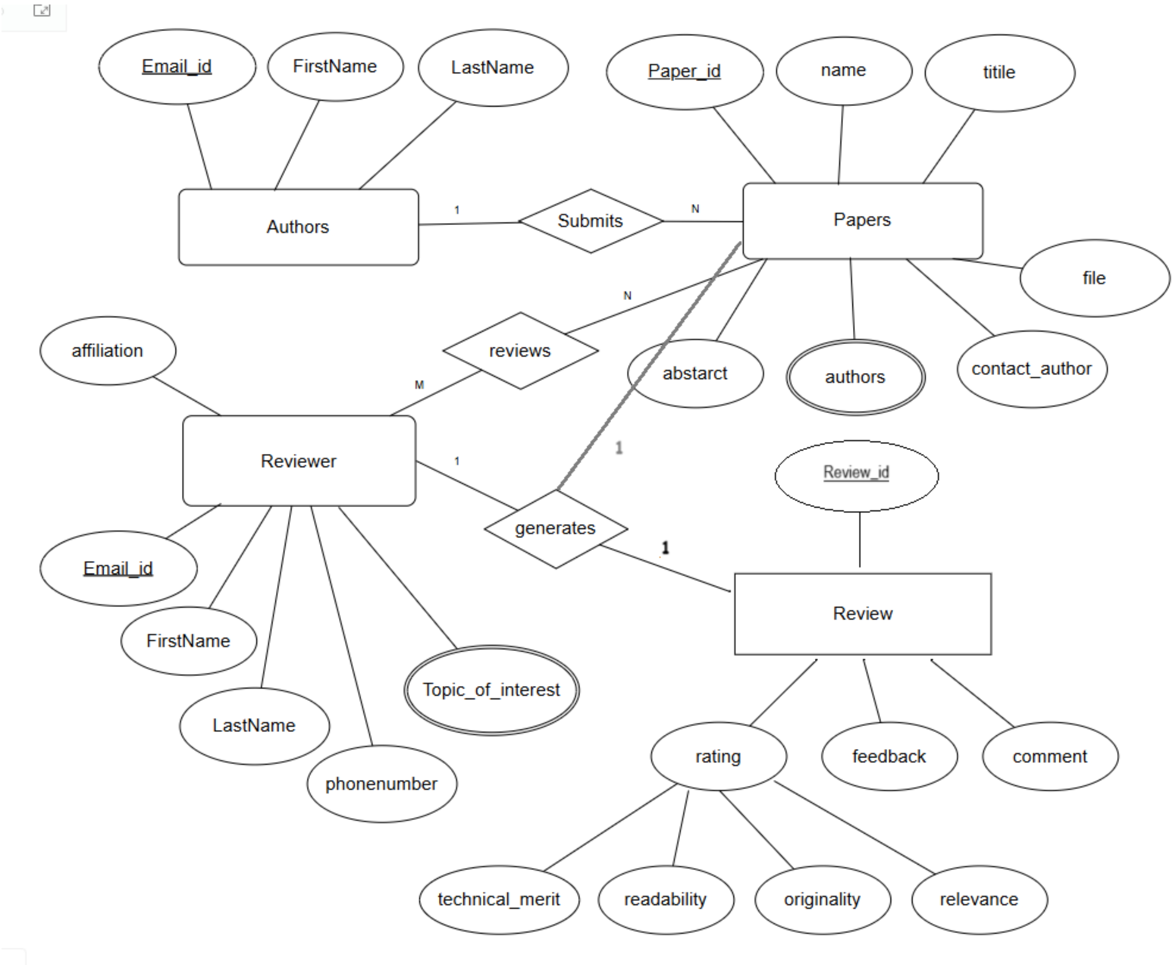
- **FR7.1:** The system shall allow an administrator to add or update restaurant staff records.
  - **Input:** Staff ID (for updates), name, role (e.g., Manager, Waiter, Delivery Coordinator).
  - **Output:** Confirmation message with Staff ID.
- **FR7.2:** The system shall allow assignment of delivery partners to area codes by staff.
  - **Input:** Staff ID, Delivery Person ID, area code.
  - **Output:** Confirmation of assignment.

8. Report Generation

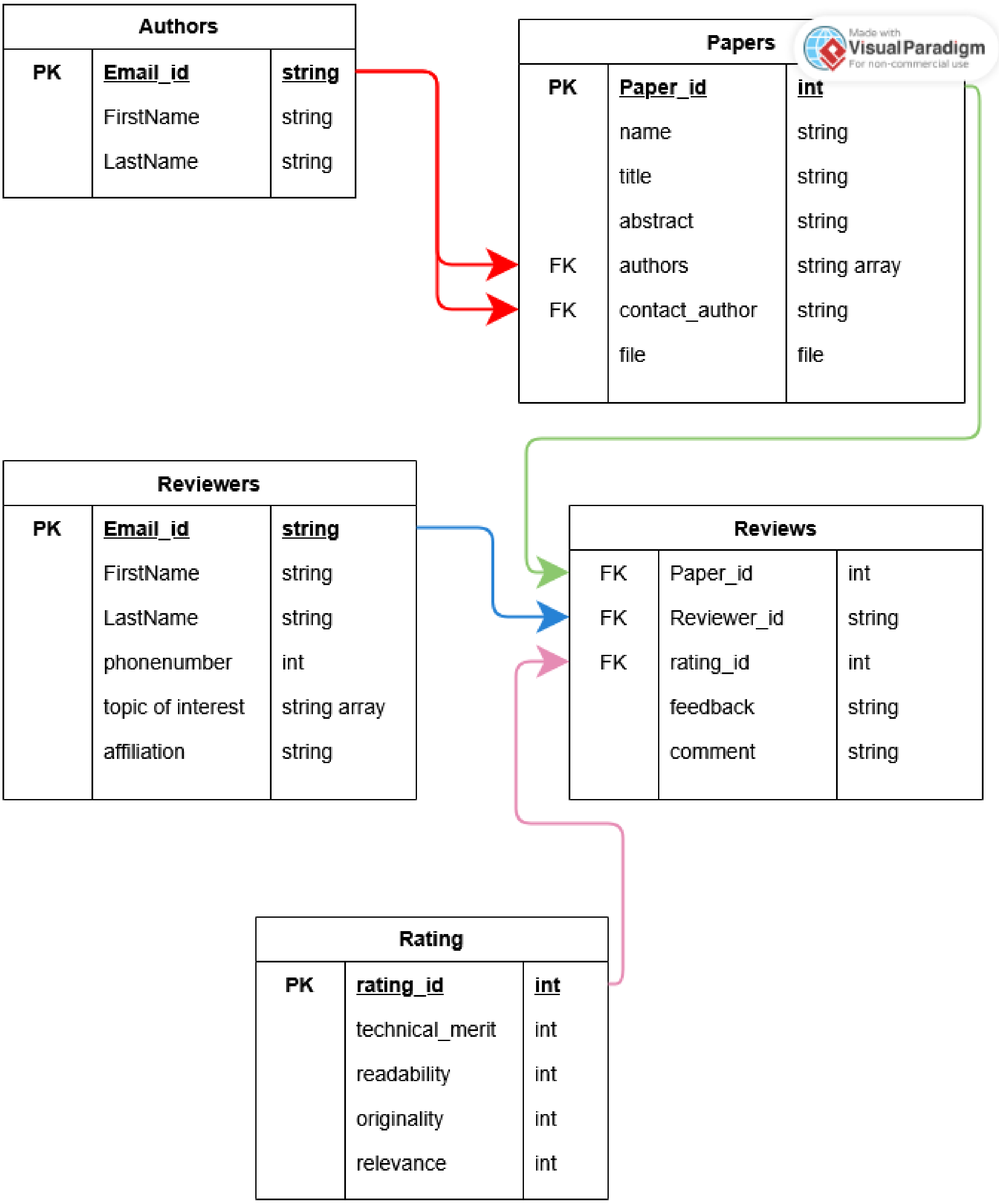
- **FR8.1:** The system shall generate daily/weekly/monthly reports on orders and revenues.
  - **Input:** Date range, report type (orders/revenue/delivery performance).
  - **Output:** report in tabular or graphical format.

Problem 2:

Er Diagram:



Relational Schema:



Authors

PK	**Email_id**	**string**
	FirstName	string
	LastName	string

Papers

PK	**Paper_id**	**int**
	name	string
	title	string
	abstract	string
FK	authors	string array
FK	contact_author	string
	file	file

Reviewers

PK	**Email_id**	**string**
	FirstName	string
	LastName	string
	phonenumber	int
	topic of interest	string array
	affiliation	string

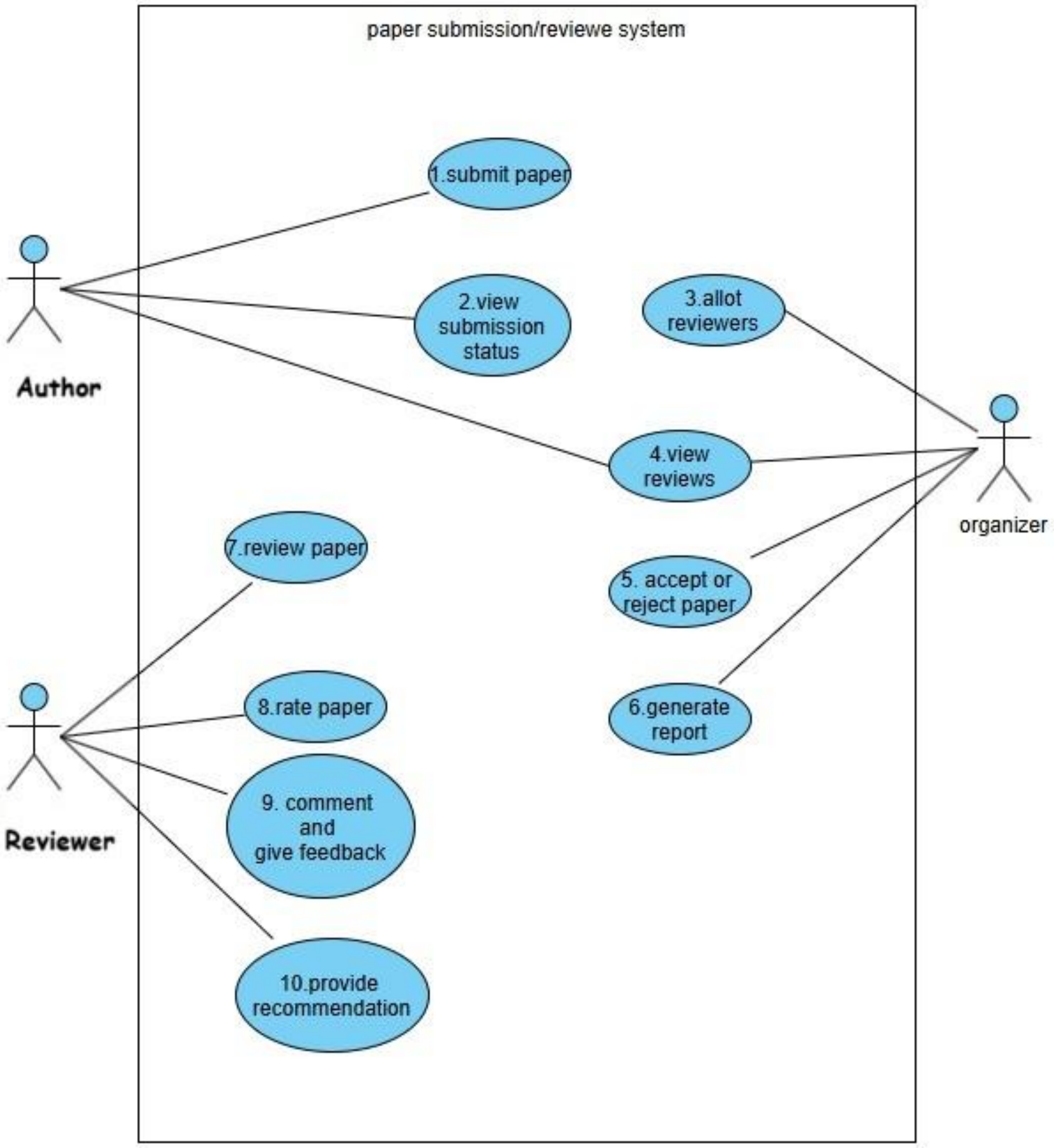
Reviews

FK	Paper_id	int
FK	Reviewer_id	string
FK	rating_id	int
	feedback	string
	comment	string

Rating

PK	**rating_id**	**int**
	technical_merit	int
	readability	int
	originality	int
	relevance	int

Use Case Diagram:



Functional Requirements (SRS):

1. Paper Submission

- **FR1.1**: The system shall allow authors to submit papers in a specified format.
  - **Input**: Paper file (PDF/DOC), title, abstract, keywords.
  - **Output**: Confirmation message with Paper ID.

- **FR1.2**: The system shall store submitted papers in a secure repository.
  - **Input**: Metadata and file from FR1.1.
  - **Output**: Stored document in database with access control.

---

2. View Submission Status

- **FR2.1**: The system shall allow authors to view the status of their submitted papers.
  - **Input**: Author login credentials or Paper ID.
  - **Output**: Status (Under Review / Accepted / Rejected).
- **FR2.2**: The system shall display decision details when available.
  - **Input**: Paper ID.
  - **Output**: Decision and feedback (if accepted/rejected).

---

3. Allot Reviewers

- **FR3.1**: The system shall allow the organizer to assign reviewers to submitted papers.
  - **Input**: Paper ID, selected Reviewer IDs.
  - **Output**: Assignment confirmation.
- **FR3.2**: The system shall validate reviewer eligibility.
  - **Input**: Paper-author metadata.
  - **Output**: Eligibility status or conflict warning.

---

4. View Reviews

- **FR4.1**: The system shall allow organizers and authors to view review feedback.
  - **Input**: Paper ID.
  - **Output**: Review ratings, comments, and recommendations.

- **FR4.2**: The system shall maintain reviewer anonymity.
  - **Input**: Review submission metadata.
  - **Output**: Anonymized display of reviewer details.

---

5. Accept or Reject Paper

- **FR5.1**: The system shall allow organizers to accept or reject papers.
  - **Input**: Paper ID, decision (Accept/Reject), optional comment.
  - **Output**: Notification to author with decision and comments.

---

6. Generate Report

- **FR6.1**: The system shall generate summary reports.
  - **Input**: Date range, filtering criteria (e.g., status).
  - **Output**: Downloadable report (PDF/CSV).

---

7. Review Paper

- **FR7.1**: The system shall allow reviewers to access papers assigned to them.
  - **Input**: Reviewer login, Paper ID.
  - **Output**: Downloadable paper document.
- **FR7.2**: The system shall log the review status.
  - **Input**: Reviewer marks as "Reviewed".
  - **Output**: Review completion log.

---

8. Rate Paper

- **FR8.1**: The system shall let reviewers rate papers.
  - **Input**: Numerical scores on defined metrics (e.g., originality, clarity).

  - **Output**: Stored ratings associated with paper and reviewer.

---

9. Comment and Give Feedback

- **FR9.1**: The system shall allow textual feedback from reviewers.
  - **Input**: Free-text comments.
  - **Output**: Displayable feedback visible to author/organizer.
- **FR9.2**: Feedback can be marked public or private.
  - **Input**: Visibility flag.
  - **Output**: Filtered display based on user role.

---

10. Provide Recommendation

- **FR10.1**: The system shall allow final recommendation from reviewer.
  - **Input**: Recommendation value (Accept / Reject / Revise).
  - **Output**: Stored and visible to the organizer.