*High Level Design –* ***Aspire***

*Overflow*

**Document ID: Author(s):** Aurora

**Version:** 0.1

**Date:** April 18,2022

 **Confidential**

**Copyright notice:** Copyright © 1996-2004 Aspire Systems. All rights reserved. The contents on this document are not to be reproduced or duplicated in any form or kind, either in part or full, without written consent from Aspire Systems

Table of Contents

1. **Introduction iii**
2. **Design Principles iii**
3. **Non-Functional Considerations for the Design iii**
   1. Execution speed for Uploading and viewing articles **iii**
   2. Execution speed for Uploading and viewing articles **iii**
   3. Security...............................................................................................................**iii**
   4. Performance........................................................................................................ **iii**
4. **System Architecture iv**
5. **Deployment Architecture**............................................................................................**.iv**
6. **Logical View.**...............................................................................................................**v**

6.1 Admin Flow.......................................................................................................... **v**

6.2 Reviewer Flow...................................................................................................... **v**

6.3 User Flow.............................................................................................................**vi**

**7. Component Interactions** **viii**

**8. Database Diagram …**......................................................................................................**ix**

**9. Database Dictionary.**......................................................................................................**xi**

**10. Deployment Environment** …..........................................................................................**xx**

**11. Modification History**.....................................................................................................**xx**

1. **Introduction**

This document provides a detailed design of the system. It starts with the design principle used, list the non-functional requirements and a detail description about the subsystem or components which comprise the system and their interactions.

**Design Principles**

The principles based on which the design is developed are

* + Increase Reusability
  + Solid Principles
  + DRW (Don’t Reinvent the Wheel)
  + DRY (Don’t Repeat Yourself)
  + SOC (Separation of Concerns)

**Non-Functional Considerations for the Design**

* **Execution speed for uploading and viewing the articles**

The Main objective of the aspire overflow is to upload an article to the system and each user can view that article in the minimal amount of Time.

* **Execution speed for uploading and viewing queries in the forum**

The Other main objective of the aspire overflow is to upload a query to the forum, so that other user can view that query and answer to that query.

* **Security**

Authentication and Authorization plays the important role in the system, So the Registered users can log into the system and access the articles and queries.

* **Performance**

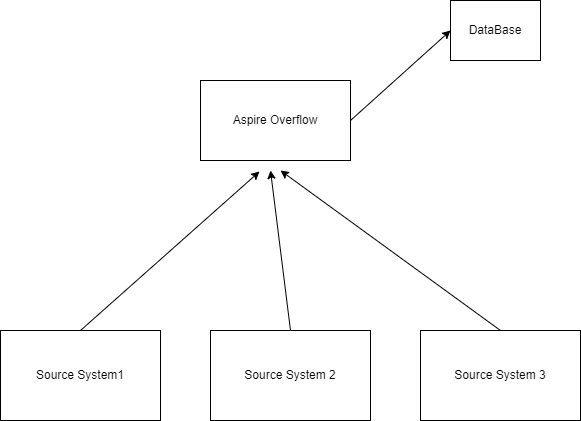
For every User action, the system will be more responsive in a very minimal amount of time.

**System Architecture**

The Aspire Overflow can be divided into the following subsystems

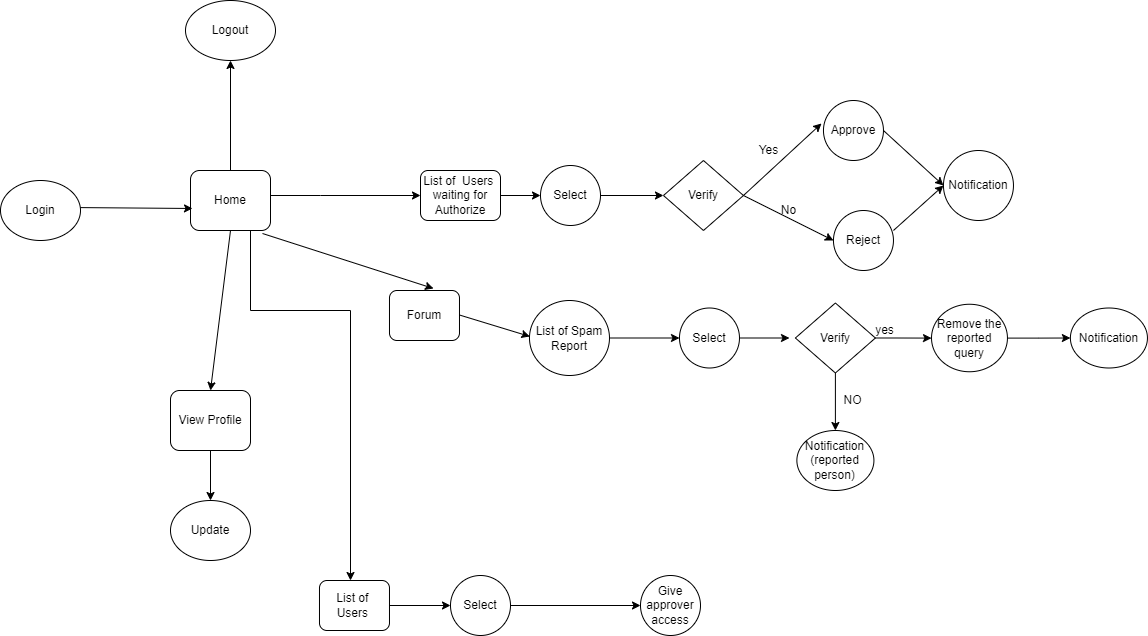
* A subsystem for logging in, create a new user, edit the details etc.
* In our system, we have 3 different user types.
  + **User** : In User Flow the User can create, Edit, Search, read, delete, comment the articles and create, search, read, comment the query.
  + **Admin** : In Admin Flow, the admin can approve the User who can access our system and gives specific users a reviewer control and manages the spams.
  + **Reviewer** : In reviewer Flow, The reviewer can select an article and review it and publishes it if it is Ok else Reject the article.

**Deployment architecture**

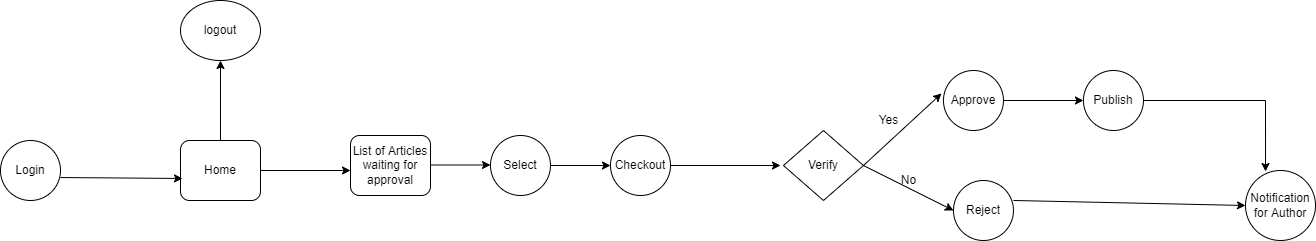


**Logical View**

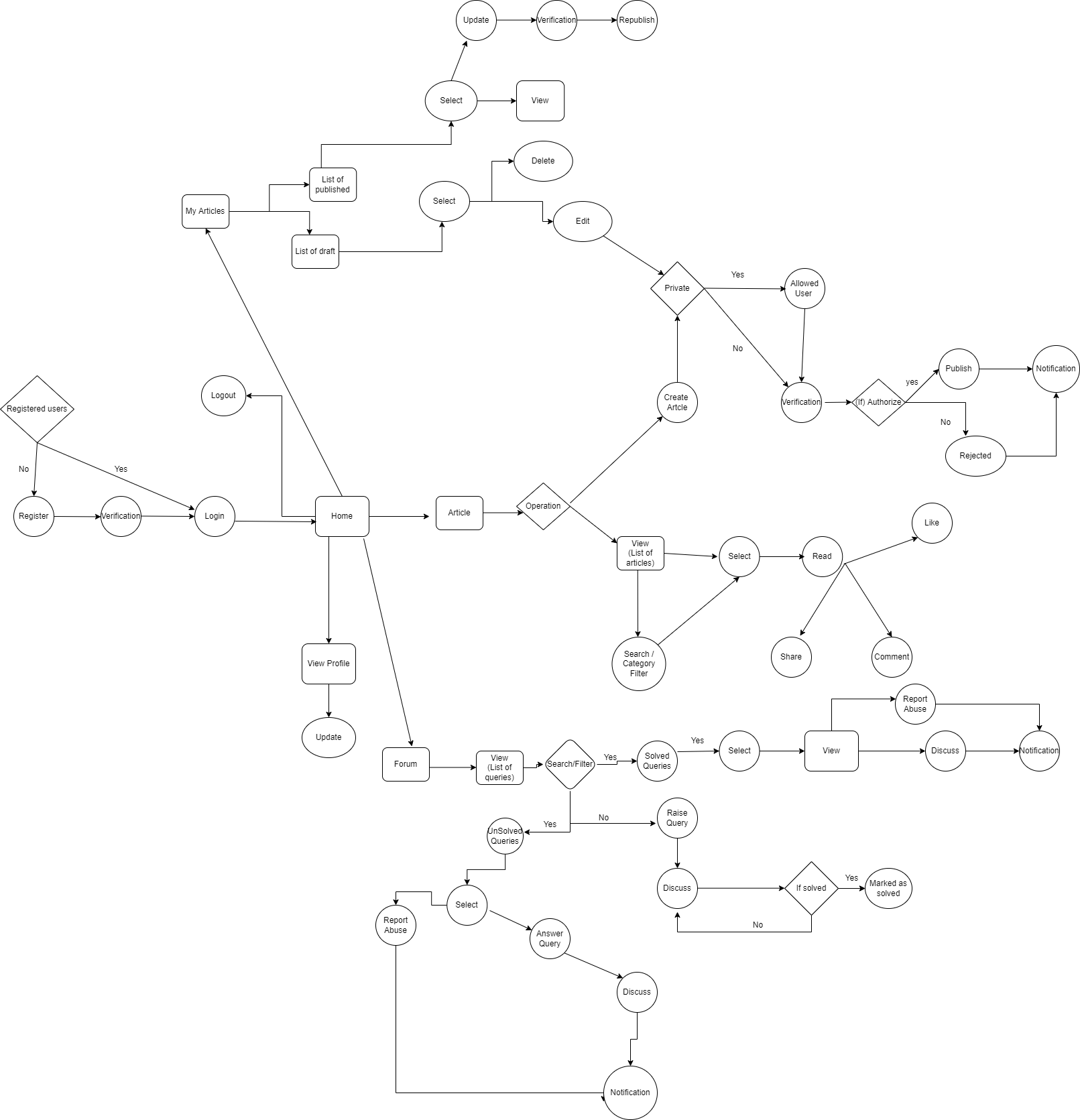
**Admin Flow**



**Reviewer Flow**



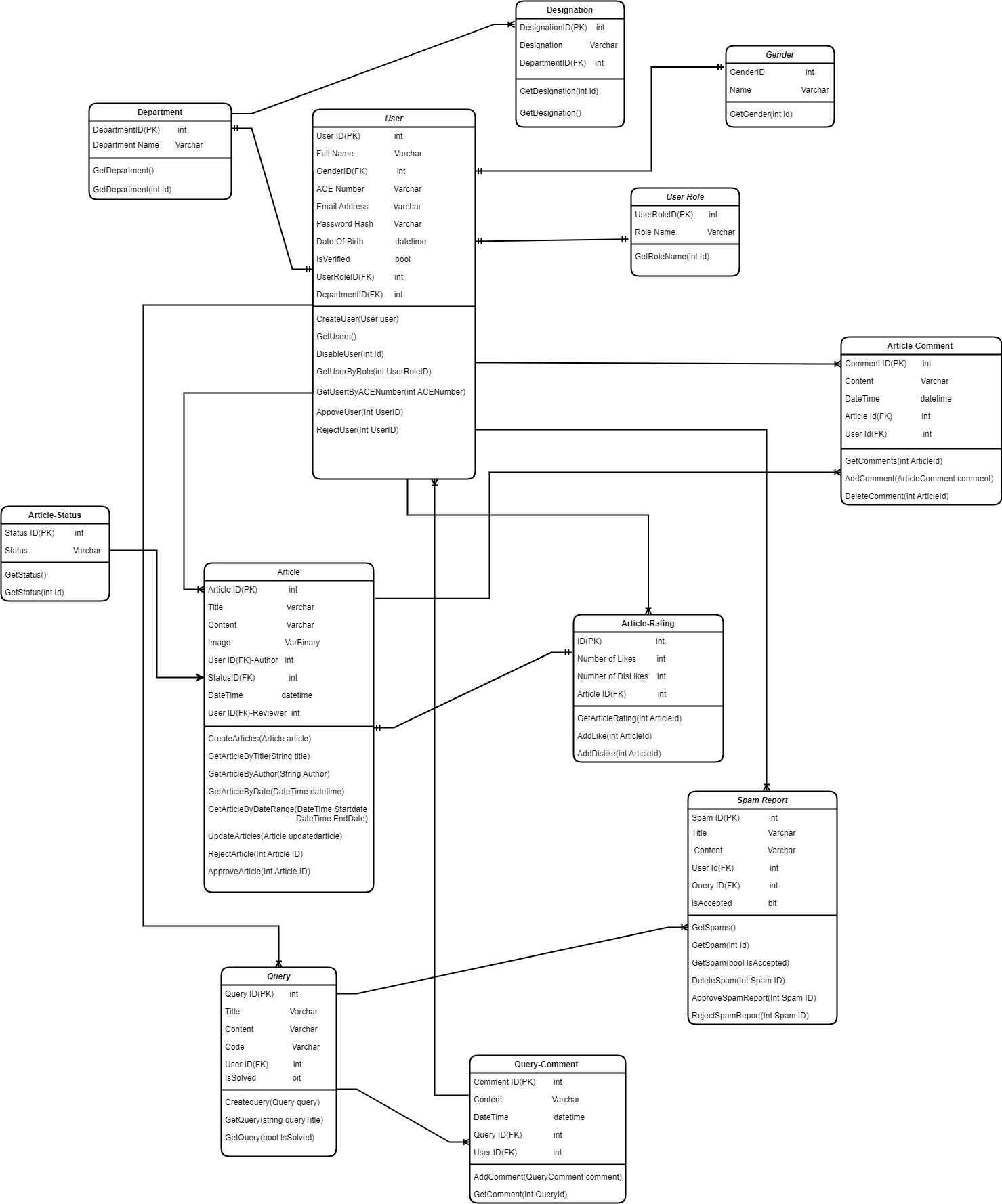
**User Flow**



**Component Interactions**

|  |  |  |
| --- | --- | --- |
| **Component** | **Collaborative Component** | **Level of Interaction** |
| Quartz Scheduler | Email Component | High level interaction |

**Database Diagram**



**Database Dictionary**

Second Normal form is used to normalize the database. Subsets of data that apply to multiple rows of a table are removed and are placed in separate tables. Relationships between these new tables and their predecessors are created through the use of foreign keys.

**Table 1 : Department**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data type** | **Constraints** | **Description** |
| DepartmentID | varchar(10) | Primary key | Unique ID for each Department |
| DepartmentName | varchar(20) | Not null | Name of the department |

**Table 2 : Designation**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data type** | **Constraints** | **Description** |
| DesignationID | varchar(10) | Primary key | Unique ID for each Designation |
| DesignationName | varchar(20) | Not null | Name of the designation |
| DepartmentID | varchar(10) | Foreign key | The ID of the department for which this designation will come under |

**Table 3 : Gender**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data type** | **Constraints** | **Description** |
| GenderID | varchar(10) | Primary key | Unique ID for each gender |
| GenderName | varchar(20) | Not null | Name of the gender |

**Table 4 : User Role**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data type** | **Constraints** | **Description** |
| UserRoleID | varchar(10) | Primary key | Unique ID for each User Role |
| RoleName | varchar(20) | Not null | Name of the Role |

**Table 5 : Article Status**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data type** | **Constraints** | **Description** |
| StatusID | varchar(10) | Primary key | Unique ID for each status of the article |
| Status | varchar(20) | Not null | Name of the status |

**Table 6 : User**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data type** | **Constraints** | **Description** |
| UserID | Varchar(10) | Primary key | Unique ID for each User |

|  |  |  |  |
| --- | --- | --- | --- |
| FullName | varchar(20) | Not null | Name of the User |
| GenderID | Varchar(10) | Foreign key | Gender ID to identify the gender of the specific user |
| ACE Number | Varchar(30) | Not null | ACE Number of the specific User |
| Email Address | varchar(50) | Not null | Email Address for the user |
| Password Hash | varchar(20) | Not null | Password which is given by the user during registration is hashed |
| Date of Birth | date | Not null | The date of birth of the user |
| IsVerified | bit | Not null | Determines whether the user is approved to login |
| UserRoleID | varchar(10) | Foreign key | ID to determine the Role of the User |
| DepartmentID | varchar(10) | Foreign key | ID to specify the department of the user |

**Table 7 : Articles**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data type** | **Constraints** | **Description** |
| ArticleID | integer | Primary key | Unique Id for all the articles created |
| Title | varchar(20) | Not null | Title for the specific Article |
| Content | Varchar() | Foreign key | The content or body of the article |
| Image | image | Not null | Any images that are inserted in the article |
| UserId -Author | integer | Foreign key | The user-id of the author who created the specific article |
| StatusID | integer | Foreign key | The specific StatusID for the article |
| DateTime | datetime | Foreign key | The published Date and Time of the article will be described |
| UserID- Reviewer | integer | Foreign key | The user-id of the reviewer who reviewed the article |

Table 8 : Article Rating

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data type** | **Constraints** | **Description** |
| ID | integer | Primary key | Unique Id for all the articles Published articles |
| Number of Likes | integer |  | The Number of likes for a specific article will be specified |
| Number of Dislikes | integer |  | The Number of Dislikes for the specific article will be specifies |
| ArticleID | integer | Foreign key | The ArticleID of the specific article will be specified |

**Table 9: Article Comment**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data type** | **Constraints** | **Description** |
| CommentID | varchar(10) | Primary key | Unique Id for all the comments for a specific article will be specified |
| Content | text | Not null | The content of the comment will be displayed |
| DateTime | datetime | Not null | The commented date and time will be specified |
| ArticleID | varchar(10) | Foreign key | The ArticleID of the specific article for which the comment has been raised |
| UserID | varchar(10) | Foreign key | The user-id of the user who has commented for the specific article |

**Table 10 : Query**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data type** | **Constraints** | **Description** |
| QueryID | varchar(10) | Primary key | Unique Id for each query |
| Title | varchar(20) | Not null | Title for the specific query |
| Content | Varchar() | Foreign key | The content or body of the query |

|  |  |  |  |
| --- | --- | --- | --- |
| Code | varchar() |  | Any code that are attached in the query |
| UserId -Author | varchar(10) | Foreign key | The user-id of the user who created the specific query |
| IsSolved | bit | Not null | Determines whether the query has been solved or not |

**Table 11 :Query Comment**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data type** | **Constraints** | **Description** |
| CommentID | varchar(10) | Primary key | Unique Id for all the comments for a specific query will be specified |
| Content | text | Not null | The content of the comment will be displayed |
| DateTime | datetime | Not null | The commented date and time will be specified |
| QueryID | varchar(10) | Foreign key | The QueryID of the specific query for which the comment has been raised |
| UserID | varchar(10) | Foreign key | The user-id of the user who has commented for the specific query |

# **Deployment Environment**

The Aspire overflow is deployed in a central server. Whoever needs to access the article and queries can log on to the system and use it.

**Modification History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Author(s)** | **Change Description** | **Version** |
| 18  April,2022 | Aurora | Created Document | 0.1 |
|  |  |  |  |