

**Content Management for Broadcasting Organizations   
COMB**

**Group Members:**

**Deepti Evani**

**Mannat Nanda**

**Suchindranath Hegde**

**Vyasaraj Vajramani**

**Table of Contents**

[**1. Problem Description** 4](#_Toc499848815)

[1.1 Business Process 4](#_Toc499848816)

[1.2 Current platform issues 4](#_Toc499848817)

[1.3 Proposed Solution 4](#_Toc499848818)

[1.4 Advantages of Proposed Solution 5](#_Toc499848819)

[1.5 Proposed Tables 5](#_Toc499848820)

[**2. Scope of Database** 6](#_Toc499848821)

[2.1 Department Table 6](#_Toc499848822)

[2.2 Role Table 6](#_Toc499848823)

[2.3 Staff Table 7](#_Toc499848824)

[2.4 StaffContact Table 7](#_Toc499848825)

[2.5 Issue Table 8](#_Toc499848826)

[2.6 Assignment Table 8](#_Toc499848827)

[2.7 Content Table 9](#_Toc499848828)

[2.8 JobRate Table 9](#_Toc499848829)

[2.10 PositionRate Table 10](#_Toc499848830)

[2.11 User Table 10](#_Toc499848831)

[**3. Data Validations** 11](#_Toc499848832)

[Department table 11](#_Toc499848833)

[Role table 11](#_Toc499848834)

[Staff table 11](#_Toc499848835)

[StaffContact table 12](#_Toc499848836)

[Issue table 12](#_Toc499848837)

[Assignment table 12](#_Toc499848838)

[Content table 13](#_Toc499848839)

[JobRate table 13](#_Toc499848840)

[JobCharge table 13](#_Toc499848841)

[PositionRate table 13](#_Toc499848842)

[**4. ER Diagram** 14](#_Toc499848843)

[**5. Database Design – Relational Schema** 15](#_Toc499848844)

[**6. Menu & Data Input Screens** 17](#_Toc499848845)

[Login Page 17](#_Toc499848846)

[Main Page 18](#_Toc499848847)

[Add Department 18](#_Toc499848848)

[Add Role 19](#_Toc499848849)

[Add Job Rate 19](#_Toc499848850)

[Add Position Rate 20](#_Toc499848851)

[Add Issue 20](#_Toc499848852)

[Add Assignment 21](#_Toc499848853)

[Add content 21](#_Toc499848854)

[**7. Reports** 22](#_Toc499848855)

[Contact Handbook 22](#_Toc499848856)

[Administrative Staff Details 22](#_Toc499848857)

[Staff count by Department 23](#_Toc499848858)

[Staff Count by Role 23](#_Toc499848859)

[Assignment Summary Report 24](#_Toc499848860)

[Issue Content Expenditure 24](#_Toc499848861)

[Staff Contribution by role 25](#_Toc499848862)

[**8. Contributions** 26](#_Toc499848863)

[Deepti 26](#_Toc499848864)

[Mannat 26](#_Toc499848865)

[Suchindranath 26](#_Toc499848866)

[Vyasaraj 26](#_Toc499848867)

# **1. Problem Description**

## **1.1 Business Process**

Mercury is a student newspaper with bi-weekly issues of its newsletter distributed on campus. It is a student run organization which has students working as staff in various roles and various student content contributors like writers, photographers and infographic designers.

The readership of the newsletter is on a upward swing and contributors working for the newspaper are now more than ever. Content production is a vital step in the process of putting out a newspaper. In order to achieve this, pitch meetings are conducted once every 2 weeks where the content for the upcoming issue of the newspaper is discussed. Writers, photographers and designers pitch story ideas that they would be interested in working on.

Once the potential list of articles for the coming issue have been discussed and short listed, editor opens up the floor for discussion and further allocates each task to one among the contributors. Then there would be follow ups and allocation changes as we inch towards publication deadline.

All the above-mentioned activities are being performed through MS Excel. As a result, the information is organized in an incoherent way on an excel sheet. Multiple updates to the sheet is human error-prone and there are no tools for tracking and report generation at any point of time.

We are looking at solving these problems by proposing a database management system which looks at the problems holistically and design a DBMS platform which not only caters to the current business problems at the Mercury but can be leveraged by any broadcasting organization looking for a tool to organize, plan and track content for their issues.

## **1.2 Current platform issues**

* Inefficient and human error prone.
* Multiple excel files need to be maintained after updates, gives rise to concurrency issues.
* Information on the excel file is not up-to-date and synchronized.
* Large amount of maintenance required.
* No tools to pull up reports giving out meaningful information.
* No proper organization of data and process flow.
* No constructs to maintain historical data or archives.

## **1.3 Proposed Solution**

With the increasing number of contributors, the excel method of management is getting cumbersome. A major chunk of time is spent on searching and retrieving data that needs to be updated. We are looking at an effective way of organizing data that needs to be stored. Our suggestion is to streamline all processes involved, which runs into several tabs on an excel document and convert them into meaningful tables. This provides modularity to data and simplifies data generation and retrieval. This platform we are designing would provide effortless ways to update the tables with information and once the data is in the system it is always concurrent.

## **1.4 Advantages of Proposed Solution**

* **Modularity of Data** – Clear demarcation of data stored in the form of tables. The data between tables can be used based on the relationships established, giving further meaning and functionality downstream.
* **Scalability** – Implementing a DBMS system would be able to support vast amounts of data effectively and efficiently. We can always add further tables based on evolving functionalities/processes.
* **Concurrency** – End users need not maintain multiple copies of the excel file, make changes and then at a later point of time consolidate into a single file. With a DBMS approach, flexibility is provided to the staff to manage data i.e. add, update, delete, insert data concurrently.
* **Search & retrieval of data** - Over time a substantial amount of data would be collected. Querying a table rather than searching endlessly in an excel file would be easy & convenient.
* **Archival of data** – Instead of fretting about the way in which excels need to be stored for future use, DBMS platform solves this issue with its inherent capability of data archival.
* **Report Generation** – SQL provides us with tools that will help in generation of meaningful and actionable reports from data already generated.

## **1.5 Proposed Tables**

* **Department –** DepartmentID, DepartmentName, Description
* **Role –** RoleID, RoleName, Description
* **Staff –** StaffID, FirstName, LastName, DOB, Gender, SSN, HireDate, TerminationDate, Department, Role
* **StaffContact –** StaffContactID, StaffID, HomePhone, WorkPhone, MobilePhone, Email, AddressLine01, AddressLine02, City, State, ZipCode
* **JobRate –** RateID, JobType, Description, Rate
* **JobCharge –** Assignment, Content, Rate
* **PositionRate –** PositionID, Staff, Rate
* **Issue -** IssueID,IssueNumber, IssueMonth, IssueYear, DateOfIssue, IssueOwner, NrOfCopies
* **Assignment -** AssignmentID, Issue, StartDate, DueDate, Description, Comments, PushtoFinal
* **Content -** ContentID, Assignment, ContentType, ContentURL, Description, Selected(Y/N), Contributor

# **2. Scope of Database**

## **2.1 Department Table**

This table contains the details of the departments in the organization. Each record in the table is identified by the unique key Department ID. The departmental information is consumed downstream when hiring new employees to an organization. Newly hired employees are mapped to a department which is present in this table.

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute Name** | **Description** | **Data Type(size)** | **Remarks** |
| DepartmentID | Identify individual department | INTEGER | Primary Key |
| DepartmentName | Name of the department | VARCHAR(30) | Required |
| Description | Description of department and its function | VARCHAR(255) | Required |

## **2.2 Role Table**

This table contains the details of the roles in the organization. Each record in the table is identified by the unique key Role ID. This information is consumed downstream when hiring new employees to an organization. Newly hired employees are mapped to a role present in this table.

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute Name** | **Description** | **Data Type(size)** | **Remarks** |
| RoleID | Identify individual roles | INTEGER | Primary Key |
| RoleName | Name of the role | VARCHAR(30) | Required |
| Description | Description of role and its privileges | VARCHAR(255) | Required |

## **2.3 Staff Table**

This table consists of the basic information of employees in the organization. Each record in the table is identified by the unique key Staff ID. Information regarding DOB, gender, hire/termination date are recorded in this table.

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute Name** | **Description** | **Data Type(size)** | **Remarks** |
| StaffID | Identify individual roles | INTEGER | Primary Key |
| FirstName | First name of the employee/staff | VARCHAR(30) | Required |
| LastName | Last name of the employee/staff | VARCHAR(30) | Required |
| DOB | Date of Birth | DATE | Required |
| Gender | Gender of employee/staff | VARCHAR(30) | Required |
| SSN | SSN of employee | VARCHAR(30) | Optional |
| HireDate | Date of Hire | DATE | Required |
| TerminationDate | Date of Termination | DATE | Optional |
| Department | Department of the employee/staff | INTEGER | Required : FK references Department |
| Role | Role of the employee/staff | INTEGER | Required : FK references Role |

## **2.4 StaffContact Table**

This table consists of the contact details of employees in the organization. Each record in the table is identified by the unique key StaffContact ID. Information phone numbers, address, email etc are recorded in this table.

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute Name** | **Description** | **Data Type(size)** | **Remarks** |
| StaffContactID | Identifies individual contact records | INTEGER | Primary Key |
| StaffID | Identifies staff with these contact details | INTEGER | Required: FK references Staff |
| HomePhone | Home Phone of employee | VARCHAR(30) | Optional |
| WorkPhone | Work Phone of employee | VARCHAR(30) | Optional |
| MobilePhone | Mobile Phone of employee | VARCHAR(30) | Required |
| Email | Email ID of employee | VARCHAR(50) | Required |
| AddressLine01 | Address Line 01 | VARCHAR(255) | Required |
| AddressLine02 | Address Line 02 | VARCHAR(255) | Optional |
| City | City of employee | VARCHAR(30) | Required |
| State | State of employee | VARCHAR(30) | Required |
| ZipCode | ZipCode of employee | VARCHAR(30) | Required |

## **2.5 Issue Table**

This table consists of the details of Issues created for publication. Each record in the table is identified by the unique key Issue ID. Information regarding issues like issue month, date of issue, phone numbers, address, email etc are recorded in this table.

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute Name** | **Description** | **Data Type(size)** | **Remarks** |
| IssueID | Identifies individual Issue of newspaper | INTEGER | Primary Key |
| IssueOwner | Identifies staff who owns this issue | INTEGER | Required : FK references Staff |
| IssueMonth | Month of the issue | VARCHAR(30) | Required |
| IssueYear | Year of issue | VARCHAR(10) | Required |
| IssueNo | Identifies the issue no in a month | INTEGER | Required |
| DateOfIssue | Date of publication of issue | DATE | Optional |
| NrOfCopies | Number of copies published | INTEGER | Optional |

## **2.6 Assignment Table**

This table consists of the details of Assignments mapped to an issue. Each record in the table is identified by the unique key Assignment ID. Information regarding assignment like start/due dates, description, push\_to\_final etc are recorded in this table.

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute Name** | **Description** | **Data Type(size)** | **Remarks** |
| AssignmentId | Identifies individual assignments | INTEGER | Primary Key | |
| Issue | Identifies the issue for which content is prepared | INTEGER | Required : FK references Issue | |
| StartDate | Date on which assignment is created | DATE | Required | |
| DueDate | Date on which assignment is due | DATE | Required | |
| Description | Description of the assignment | VARCHAR(255) | Optional | |
| Comments | Comments to be updated | VARCHAR(255) | Optional | |
| Push\_To\_Final | Status of assignment completion | CHAR | Optional | |

## **2.7 Content Table**

This table consists of the details of Content generated for a Assignment/Issue. Each record in the table is identified by the unique key Content ID. Information regarding Content like content type, content url, selected(yes/no) etc are recorded in this table.

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute Name** | **Description** | **Data Type(size)** | **Remarks** |
| ContentID | Identifies individual content | INTEGER | Primary Key |
| Assignement | Identifies the assignment to which content is prepared | INTEGER | Required : FK references Assignment |
| ContentType | Type of content being generated | VARCHAR(30) | Required |
| ContentURL | Link where content is uploaded | VARCHAR(255) | Required |
| Description | Description of the assignment | VARCHAR(255) | Optional |
| Selected(Y/N) | Flag to identify whether content approved | CHAR | Required |
| Contributor | Comments to be updated | VARCHAR(255) | Required: FK references Staff |

## **2.8 JobRate Table**

This table consists of the details of rates charged for the type of Content produced for a particular Assignment/Issue. Each record in the table is identified by the unique key JobRate ID. Information regarding Content like content type, rate, description etc are recorded in this table.

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute Name** | **Description** | **Data Type(size)** | **Remarks** |
| RateID | Identifies individual rates per role | INTEGER | Primary Key |
| JobType | Type of job/assignment | VARCHAR(30) | Required |
| Description | Description of the type of job and rate | VARCHAR(255) | Optional |
| Rate | Rate chargeable for the JobType | INTEGER | Required |

**2.9 JobCharge Table**

This table consists of the costs to the organization deduced by the number of contents selected for publishing for a particular Assignment/Issue. Each record in the table is identified by the unique key JobCharge ID. Information regarding charges Assignment, Staff and Rate are recorded in this table.

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute Name** | **Description** | **Data Type(size)** | **Remarks** |
| Assignment | Assignment on which charged | INTEGER | Primary Key: FK references Assignment |
| Content | Content for which charged | INTEGER | Primary Key: FK references Content |
| Rate | Rate at which job charged | INTEGER | Required : FK references JobRate |

## **2.10 PositionRate Table**

This table consists of the costs to the organization for fixed positions of Administration department. Each record in the table is identified by the unique key PositionRate ID. Information regarding charges Staff and Rate are recorded in this table.

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute Name** | **Description** | **Data Type(size)** | **Remarks** |
| PositionID | Uniquely identifies the rates for positions | INTEGER | Primary Key |
| Staff | StaffID on the position | INTEGER | Required : FK references Staff |
| Rate | Rate chargeable for the Position | INTEGER | Required |

## **2.11 User Table**

This table consists of the login details of users with access to the application. Each record in the table is identified by the unique key PositionRate ID. Information regarding charges Staff and Rate are recorded in this table.

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute Name** | **Description** | **Data Type(size)** | **Remarks** |
| UserID | Uniquely identifies the user details for login | INTEGER | Primary Key |
| UserName | Username of the user | VARCHAR(30) | Required |
| UserLogin | Login ID of the user | VARCHAR(30) | Required |
| Password | Password of the user | VARCHAR(30) | Required |
| UserSecurity | Defines user security level | VARCHAR(30) | Required |

# **3. Data Validations**

This section details the data validations performed on the attributes of database tables. We find this imperative as this helps in maintaining the integrity of data. This is achieved by controlling the input of records by checking on values for attributes.

## **Department table**

|  |  |
| --- | --- |
| **Attribute Name** | **Validation** |
| DepartmentID | Auto generated |
| DepartmentName | Only characters |
| Descrption | None |

## **Role table**

|  |  |
| --- | --- |
| **Attribute Name** | **Validation** |
| RoleID | Auto generated |
| RoleName | Only characters |
| Description | None |

## **Staff table**

|  |  |
| --- | --- |
| **Attribute Name** | **Validation** |
| StaffID | Auto generated |
| FirstName | Only characters |
| LastName | Only characters |
| DOB | Only Dates |
| Gender | LOV - Male, Female |
| SSN | Only numbers |
| HireDate | Only Dates |
| TerminationDate | Only Dates :  TerminationDate > HireDate |
| Department | LOV - Department table |
| Role | LOV - Role table |

## **StaffContact table**

|  |  |
| --- | --- |
| **Attribute Name** | **Validation** |
| StaffContactID | Auto generated |
| StaffID | Staff details from Staff table |
| HomePhone | Only Numbers |
| WorkPhone | Only Numbers |
| MobilePhone | Only Numbers |
| Email | None |
| AddressLine01 | None |
| AddressLine02 | None |
| City | Only Characters |
| State | Only Characters 2 character long |
| ZipCode | Only Numbers |

## **Issue table**

|  |  |
| --- | --- |
| **Attribute Name** | **Validation** |
| IssueID | Auto generated |
| IssueOwner | LOV from Staff table |
| IssueMonth | Only Characters |
| IssueYear | Only Numbers |
| IssueNo | Only Numbers |
| DateOfIssue | Only Date |
| NrOfCopies | Only Numbers |

## **Assignment table**

|  |  |
| --- | --- |
| **Attribute Name** | **Validation** |
| AssignmentId | Auto generated |
| Issue | LOV from Issue table |
| StartDate | Only Dates |
| DueDate | Only Dates |
| Description | None |
| Comments | None |
| Push\_To\_Final | Check box |

## **Content table**

|  |  |
| --- | --- |
| **Attribute Name** | **Validation** |
| ContentID | Auto generated |
| Assignement | LOV from Assignment table |
| ContentType | LOV - Article, Graphics, Photograph |
| ContentURL | None |
| Description | None |
| Selected(Y/N) | Check box |
| Contributor | LOV from Staff table |

## **JobRate table**

|  |  |
| --- | --- |
| **Attribute Name** | **Validation** |
| RateID | Auto generated |
| JobType | LOV - Writer, Designer, Photographer |
| Description | None |
| Rate | Only Numbers |

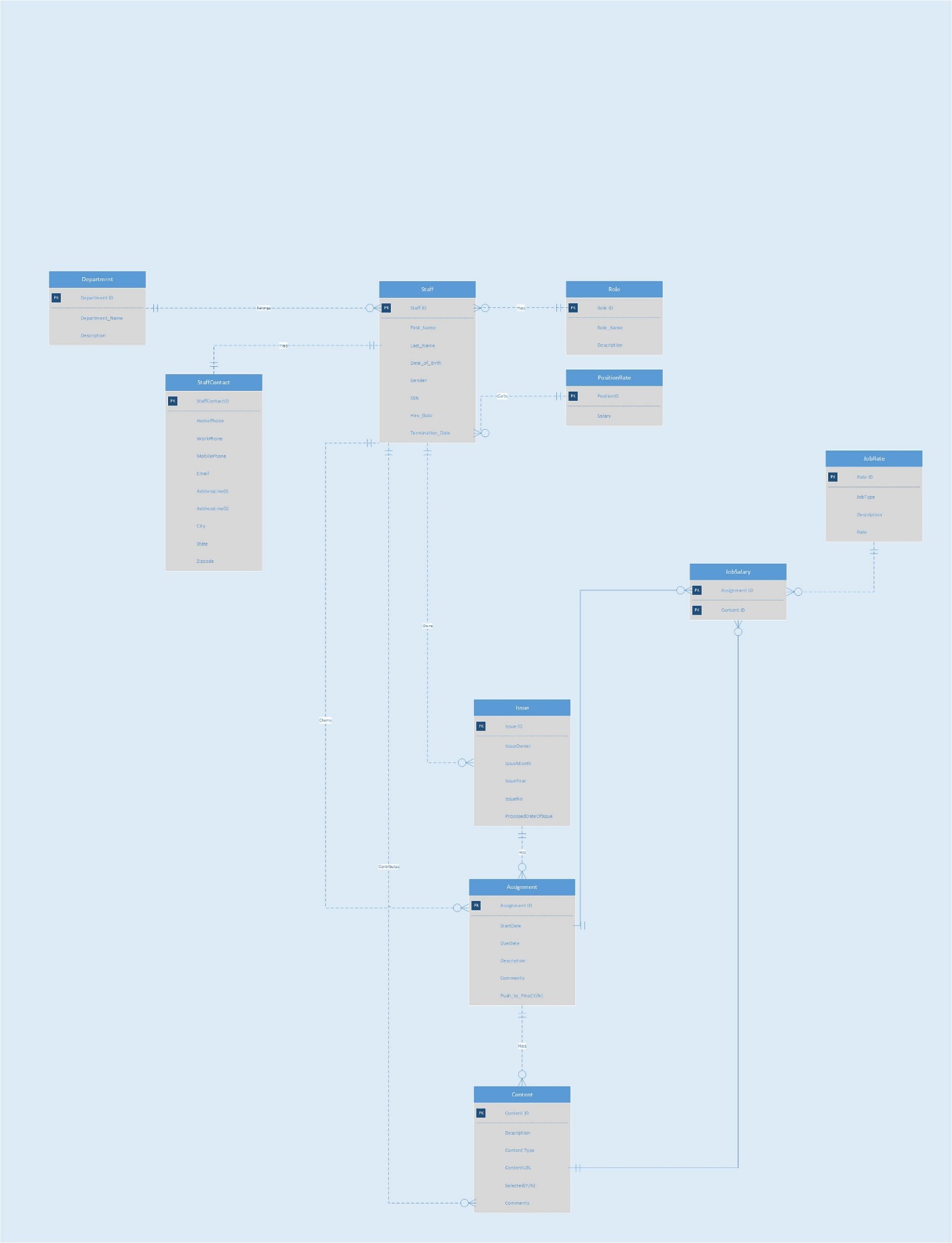
## **JobCharge table**

|  |  |
| --- | --- |
| **Attribute Name** | **Validation** |
| Assignment | LOV from Assignment table |
| Content | LOV from Content table |
| Rate | LOV from JobRate table |

## **PositionRate table**

|  |  |
| --- | --- |
| **Attribute Name** | **Validation** |
| PositionID | Auto generated |
| Staff | LOV from Staff table |
| Rate | Only Numbers |

# **4. ER Diagram**



# **5. Database Design – Relational Schema**

CREATE TABLE Department

(DepartmentID INTEGER,

DepartmentName VARCHAR(30) CONSTRAINT DptNameRqd NOT NULL,

Descrption VARCHAR(255),

CONSTRAINT PKDepartment PRIMARY KEY (DepartmentID)

)

CREATE TABLE Role

(RoleID INTEGER,

RoleName VARCHAR(30) CONSTRAINT RoleNmRqd NOT NULL,

Descrption VARCHAR(255),

CONSTRAINT PKRole PRIMARY KEY (RoleID)

)

CREATE TABLE Staff

(StaffID INTEGER,

FirstName VARCHAR(30) CONSTRAINT StaffFNameReqd NOT NULL,

LastName VARCHAR(30) CONSTRAINT StaffLNameReqd NOT NULL,

DOB DATE CONSTRAINT StaffDOBReqd NOT NULL,

Gender VARCHAR(30) CONSTRAINT StaffGenderReqd NOT NULL,

SSN VARCHAR(30),

HireDate DATE CONSTRAINT StfHDate NOT NULL,

TerminationDate DATE,

StaffDept INTEGER,

StaffRole INTEGER,

CONSTRAINT PKStaff PRIMARY KEY (StaffID),

CONSTRAINT FKStaffDept FOREIGN KEY (StaffDept) REFERENCES Department,

CONSTRAINT FKRole FOREIGN KEY (StaffRole) REFERENCES Role,

)

CREATE TABLE StaffContact

(StaffContactID INTEGER,

Staff INTEGER CONSTRAINT SCStaffReqd NOT NULL,

HomePhone VARCHAR(30),

WorkPhone VARCHAR(30),

MobilePhone VARCHAR(30) CONSTRAINT SCMobReqd NOT NULL,

Email VARCHAR(50) CONSTRAINT SCEmailReqd NOT NULL,

AddressLine01 VARCHAR(255) CONSTRAINT SCAdrLn1Reqd NOT NULL,

AddressLine02 VARCHAR(255),

City VARCHAR(30) CONSTRAINT SCCityReqd NOT NULL,

State VARCHAR(30) CONSTRAINT SCStateReqd NOT NULL,

ZipCode VARCHAR(30) CONSTRAINT SCZipReqd NOT NULL,

CONSTRAINT PKStaffContact PRIMARY KEY(StaffContactID),

CONSTRAINT FKStaff FOREIGN KEY (Staff) REFERENCES Staff

)

CREATE TABLE Issue

(IssueID INTEGER,

IssueOwner INTEGER CONSTRAINT FKIssueClmn NOT NULL,

IssueMonth VARCHAR(30) CONSTRAINT IssueMonthReqd NOT NULL,

IssueNo VARCHAR(2) CONSTRAINT IssueNoReqd NOT NULL,

DateOfIssue DATE,

NrOfCopies INTEGER,

CONSTRAINT PKIssue PRIMARY KEY (IssueID),

CONSTRAINT FKIssue FOREIGN KEY (IssueOwner) REFERENCES Staff

)

CREATE TABLE Assignment

(AssignmentId INTEGER,

Issue INTEGER CONSTRAINT FKAsgClmn NOT NULL,

StartDate DATE CONSTRAINT AssSDate NOT NULL,

DueDate DATE CONSTRAINT AssDDate NOT NULL,

Descrption VARCHAR(255),

Comments VARCHAR(255),

AssignmentStatus VARCHAR(30) CONSTRAINT AssStatus NOT NULL,

CONSTRAINT PKAssignment PRIMARY KEY (AssignmentID),

CONSTRAINT FKAssignment FOREIGN KEY (Issue) REFERENCES Issue

)

CREATE TABLE Content

(ContentID INTEGER,

Assignment INTEGER CONSTRAINT FKCntClmn NOT NULL,

ContentType VARCHAR(30) CONSTRAINT CntType NOT NULL,

ContentURL VARCHAR(255) CONSTRAINT CntUrl NOT NULL,

Descrption VARCHAR(255),

Selected CHAR,

Comments VARCHAR(255),

CONSTRAINT PKContent PRIMARY KEY (ContentID),

CONSTRAINT FKContent FOREIGN KEY (Assignment) REFERENCES Assignment

)

CREATE TABLE JobRate

(RateID INTEGER,

JobType VARCHAR(30) CONSTRAINT JobRateTypeReqd NOT NULL,

Descrption VARCHAR(255),

Rate INTEGER CONSTRAINT JobRateReqd NOT NULL,

CONSTRAINT PKJobRate PRIMARY KEY (RateID)

)

CREATE TABLE JobCharge

(Assignment INTEGER,

Content INTEGER,

Rate INTEGER CONSTRAINT JCRateReqd NOT NULL,

CONSTRAINT PKJobCharge PRIMARY KEY (Assignment, Content),

CONSTRAINT FKAsgJC FOREIGN KEY (Assignment) REFERENCES Assignment,

CONSTRAINT FKCntJC FOREIGN KEY (Content) REFERENCES Content,

CONSTRAINT FKRateJC FOREIGN KEY (Rate) REFERENCES JobRate

)

CREATE TABLE PositionRate

(PositionID INTEGER,

Staff INTEGER CONSTRAINT PRStfReqd NOT NULL,

Rate INTEGER CONSTRAINT PRRateReqd NOT NULL,

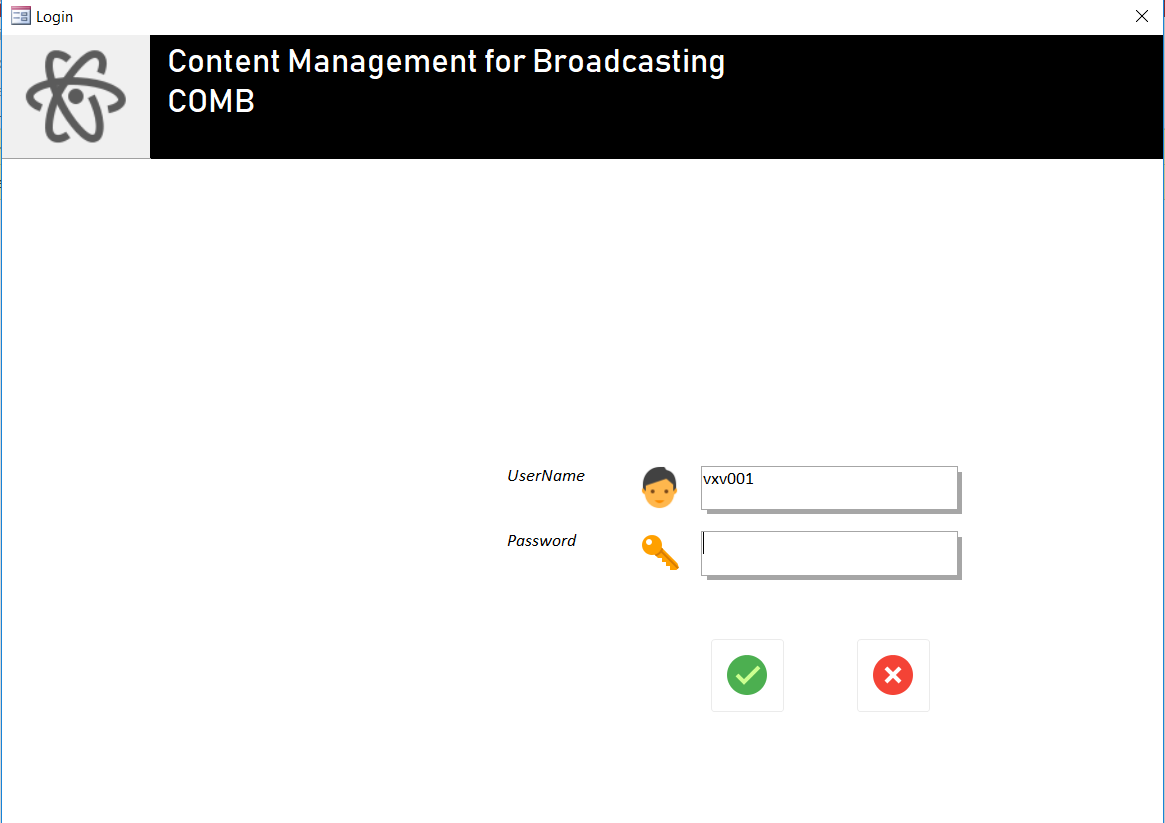
CONSTRAINT PKPosRate PRIMARY KEY (PositionID),

CONSTRAINT FKPosRate FOREIGN KEY (Staff) REFERENCES Staff

)

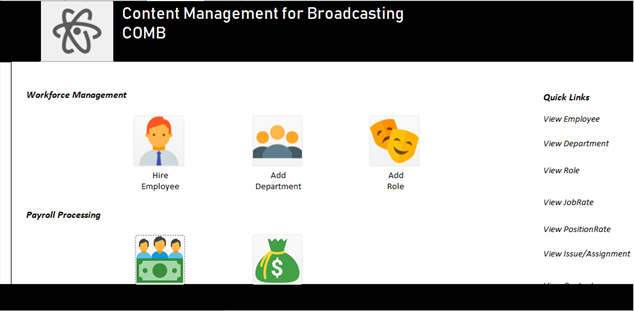
# **6. Menu & Data Input Screens**

## **Login Page**

****

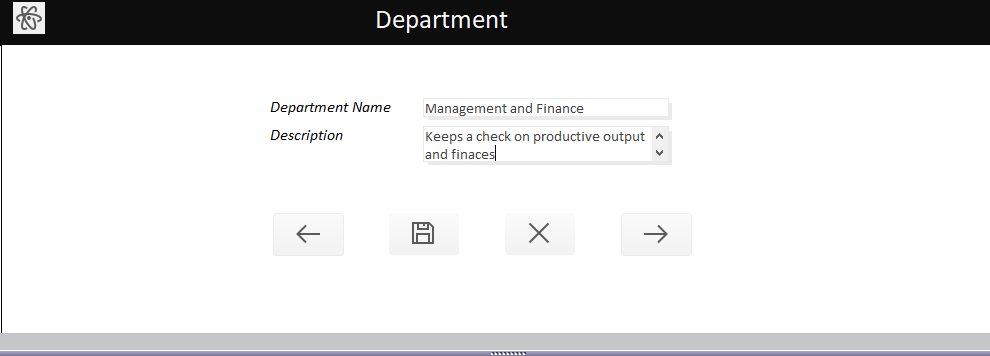
On launching the access file, this is the first page displayed. On successfully keying in the credentials and clicking on the ‘OK’ button, the control is re-directed to the main page of the application. On hitting ‘CANCEL’ the application is closed.

## **Main Page**



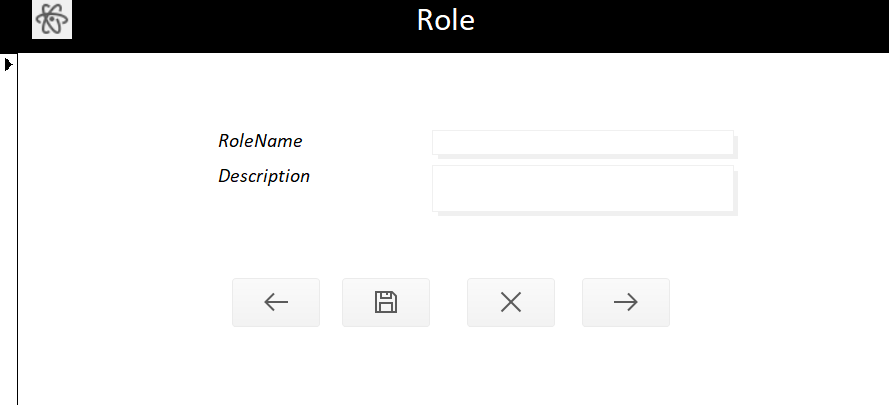
From this page, all the forms and reports can be launched. We have designed the UI in such a way that we have different forms for adding data into tables and viewing them.

## **Add Department**



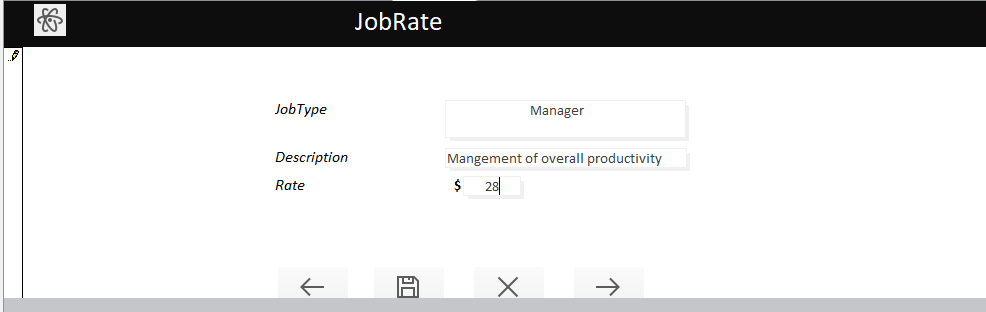
This screen is used to add a new department and describe the functionality of the newly added department.

## **Add Role**



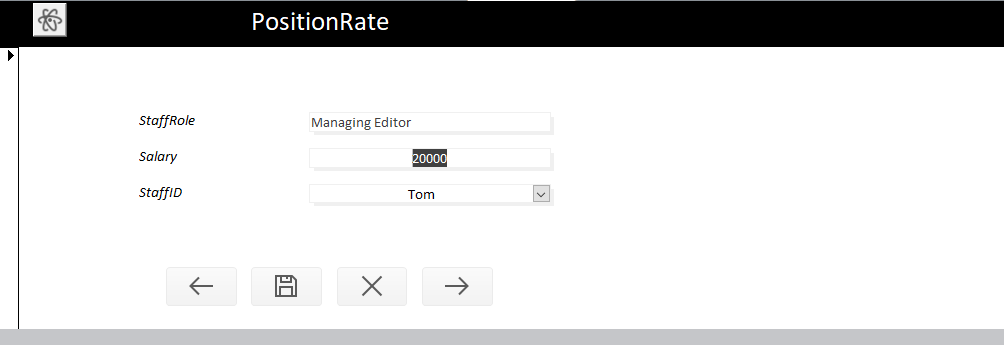
This screen is used to add a new role and describe the functionality of the newly added role.

## **Add Job Rate**



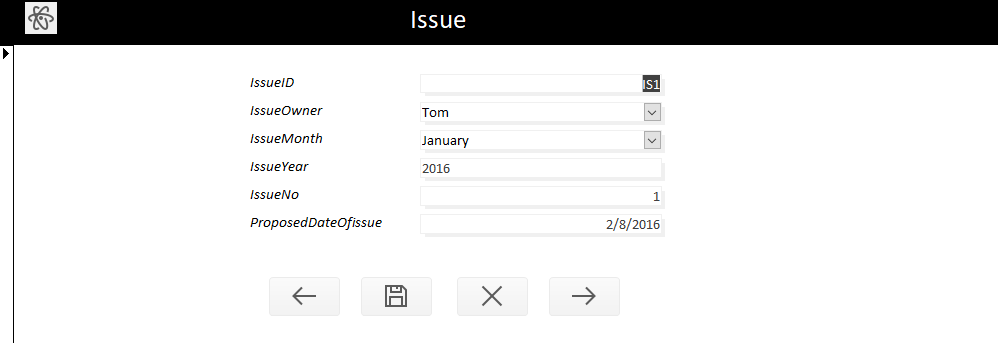
This screen is used to add a job rate (payment per hour) of a specific job.

## **Add Position Rate**



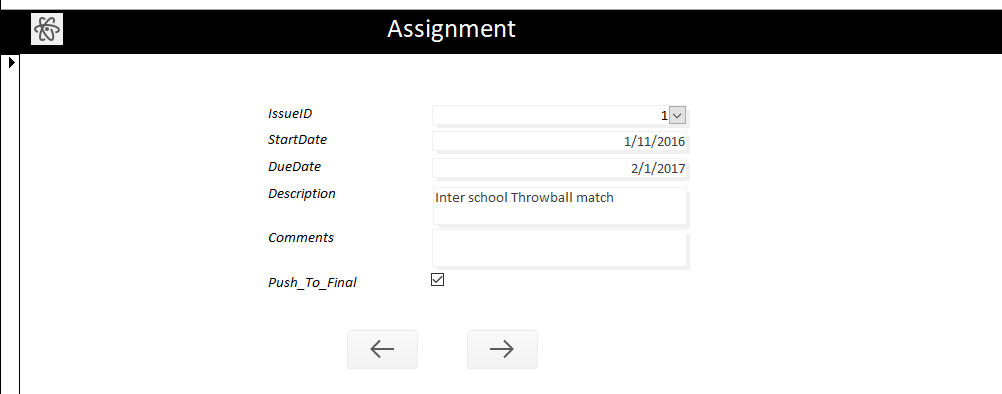
This screenshot specifies the payment made per hour to a specific staff role.

## **Add Issue**



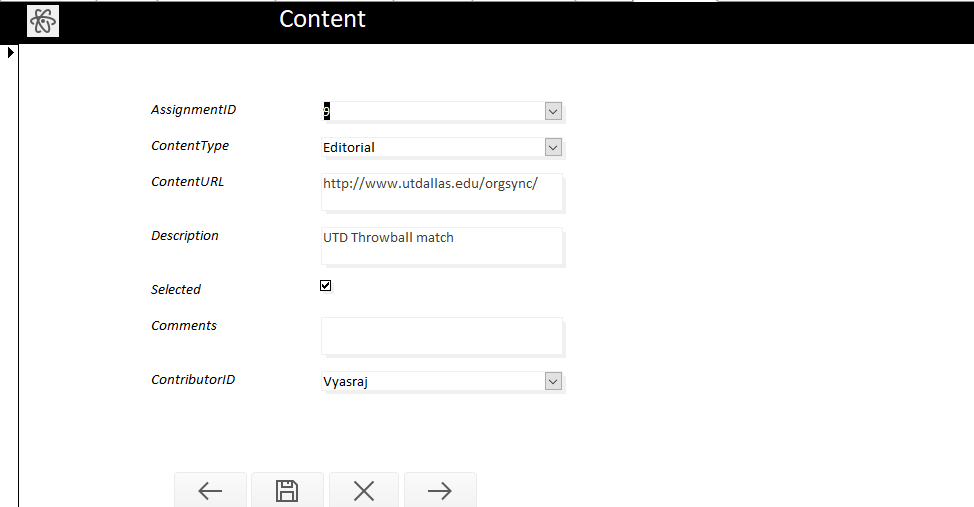
This screen is used to add an issue with its details.

## **Add Assignment**



This screen is used to add an assignment and details with it.

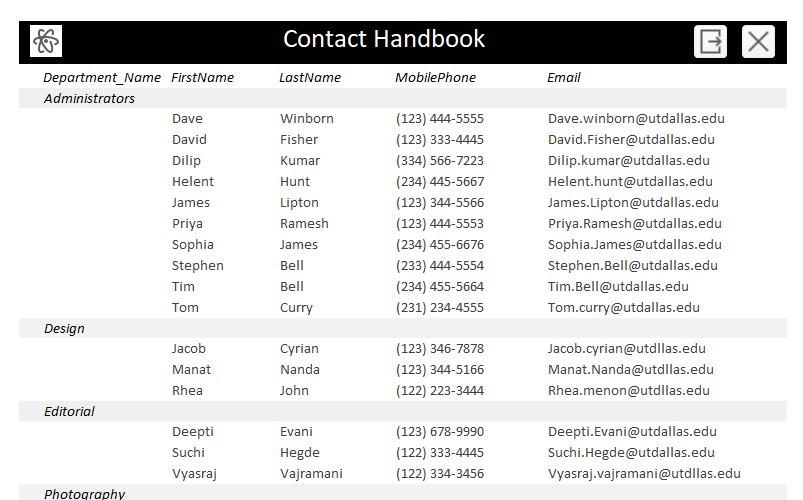
## **Add content**



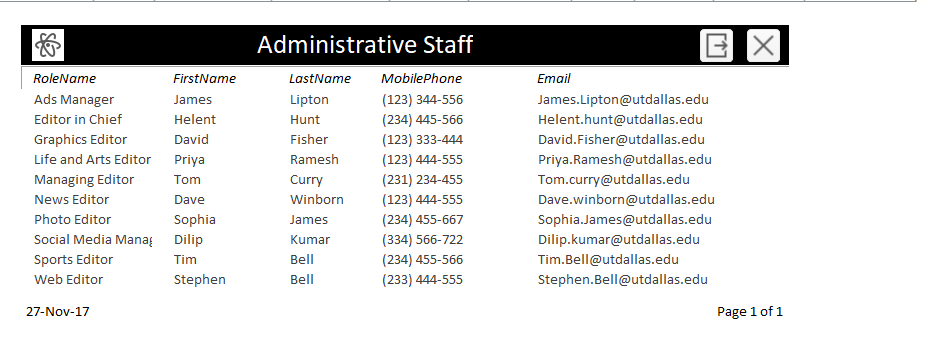
This page adds the content type and description along with other details

# **7. Reports**

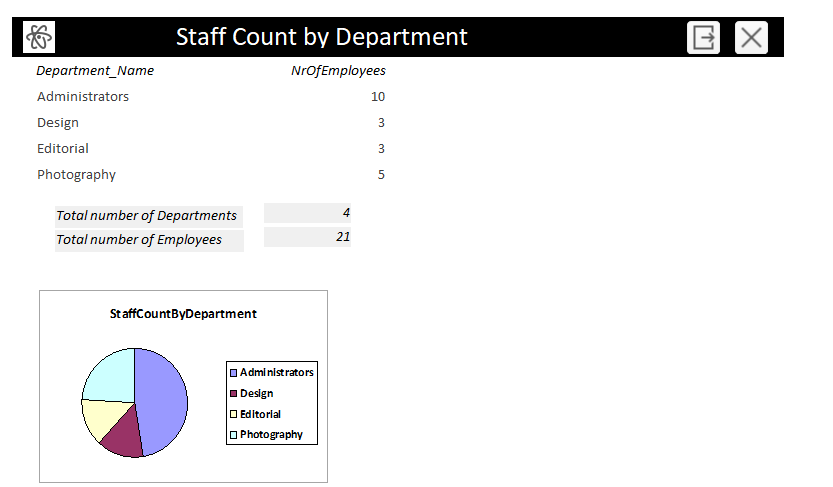
## **Contact Handbook**



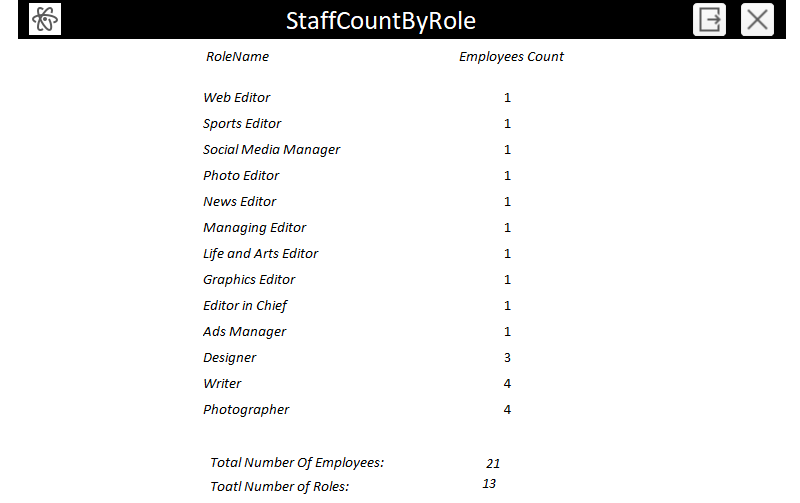
## **Administrative Staff Details**



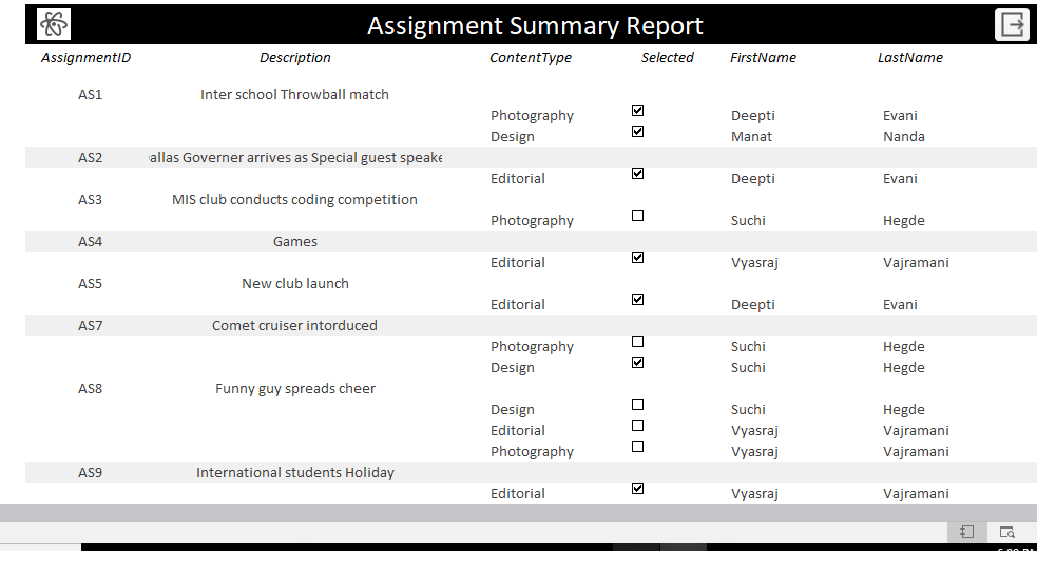
## **Staff count by Department**



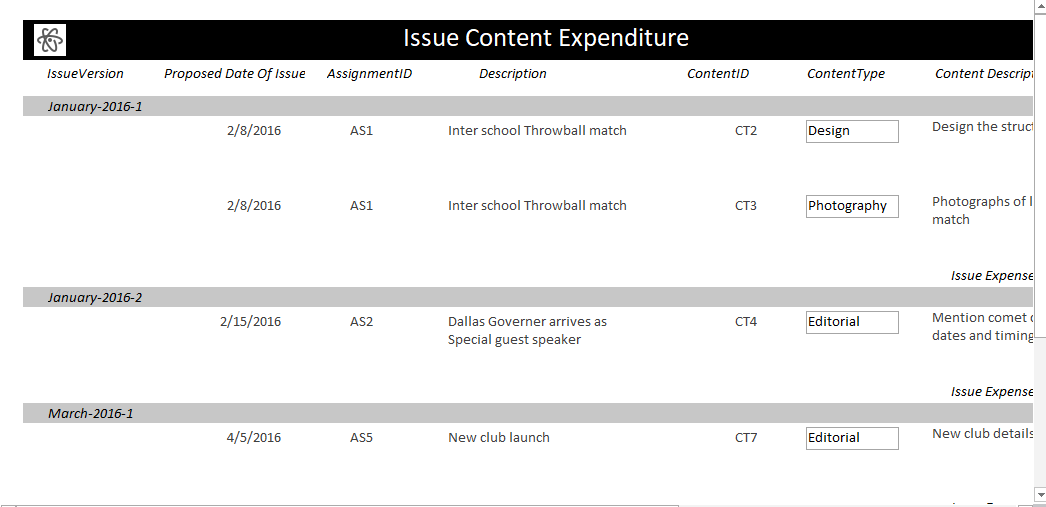
## **Staff Count by Role**



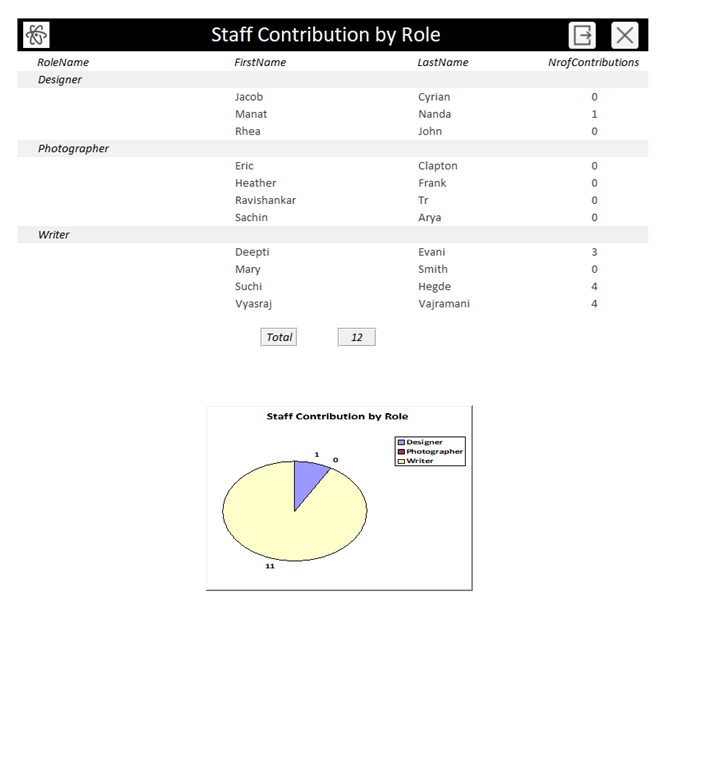
## **Assignment Summary Report**



## **Issue Content Expenditure**



## **Staff Contribution by role**



# **8. Contributions**

**Deepti** - Oversaw the Design right from the nascent stages and ensured that there was no redundancy in term of design and structure. Also, came up with and implemented ideas that added to the overall value of the project such as the Login page.

**Mannat** - Took on the herculean task of populating the tables with data and carried out the required testing to cover any gaps in system design and structure in areas such as data validation and made suggestion to improve the design further.

**Suchindranath** - Instrumental in transforming the proposal into an entity relationship diagram. Oversaw table and form level validations as deemed appropriate post creation of tables. Created and embellished the forms further to create a user-friendly interaction and a hassle-free user interaction.  Instrumental in designing macros for forms and reports which brought in a lot of novelty in terms of functionality and aesthetics.

**Vyasaraj** - Facilitated interaction and follow-up meeting with the Web Editor at the Mercury to understand the day-to-day functioning at the Newsletter, which assisted us in bringing our idea to fruition. Proposed and implemented the reports by building customer specific queries.  Conceptualized the design of UI, selection of a color palette and overall functional flow of UI to ensure better user experience.