**Name of the Project(Batch Name/No)**

| Version Number | Date | Author/Owner | Description of Change |
| --- | --- | --- | --- |
|  |  |  |  |

**NOTE:**

**Version no is increased for every deliverable. And increased further for adding or deleting features.**

**Eg.,**

|  |  |  |  |
| --- | --- | --- | --- |
| Version Number | Date | Author/Owner | Description of Change |
| 01 | 12/12/2022 | XYZ | Problem statement , Features identification, Project plan . |
| 02 | 1/1/2023 | XYZ | Project Plan changed |
| 03 | 15/1/2023 | XYZ | Conceptual design |

**:**

**:**

**The contents of the document should include the following**

1. **Problem Statement & Requirements Definition**

***SAMPLE LIBRARY MANAGEMENT SYSTEM:***

***We decided to create a database that can help organize a library. It would be ideal for a library upgrading from a card or paper system where they have a card for each piece of media and for each customer or patron. It would also benefit a library where all employee records are still on a paper basis as well including all hire paperwork and payroll information such as salary. As you can imagine for a paper system, it makes figuring out how many books are overdue, or how many people owe fees, take quite a while to figure out for the employees. Patrons also have no control over the system.***

The purpose of this database is to automate and replace the current card and paper system. All tasks previously recorded on paper or cards will be integrated into the new system. For example, based on due dates, librarians can run reports to see who has late books (checked out media report), who owes fees for late books or damaged books (cost report) and much more. It will only take a few seconds to run the report as opposed to going through all of the cards by hand, saving the librarians hours a week.

The library patrons will also have added usability. Potentially there could be computers in the library for their use to see what books they currently have checked out, and what they have checked out in the past.. The last main feature that would be new to patrons is that they would have web access to the library to reserve books. It will be a simple form with the same search capabilities that allows them to submit a request. After the request, the librarians could then pull the books and place them on hold on a first come first served basis. Other capabilities include being able to renew a book online. Each book can only be renewed once before it is considered to be late.

Lastly, the library manager will have the same access as the librarian as well as some added features. Employee records will all be stored in the database, replacing the paper system. He will also be able to order new media (not an actual function of the database) and add it to the system. He will then have more reporting capabilities including a usage report that shows how often something is used to help determine what types of media should be ordered and in what quantities.

Hopefully with all of the changes, it would make the library run much more smoothly. With less time spent filling out paperwork, more time can be devoted to serving the customers and maintaining the potentially vast amounts of media held in the library. The system will hopefully also be more reliable and user friendly to everyone. While it will require some training for employees, once the system is in place, the benefits will greatly outweigh the costs of implementing the system since using the computer to do most daily tasks will be much faster than a paper based system.

1. **Project features identified**

|  |  |  |
| --- | --- | --- |
| **Feature ID** | **Feature name** | **Description** |
| **Eg. T01** | **Login** | **Authentication and on success takes you to your dashboard** |
|  |  |  |

1. **Software and hardware details**

|  |  |
| --- | --- |
| **Platform** |  |
| **Frontend/console** |  |
| **Backend/server** |  |
| **Database** |  |
| **Programming Language : Frontend** |  |
| **Backend/server: programming Language** |  |

1. **Project Plan**

**Eg.,**

Chart, timeline, waterfall chart

Description automatically generated

1. **Specify a Google drive link for sharing all your RS documents and future deliverables**

**================= END OF DELIVERABLE 1 =================**

1. **CONCEPTUAL DESIGN PHASE:**

**1. Entity Relationship Model**

Diagram

Description automatically generated

**2.Object Model Diagram (optional)**

Diagram

Description automatically generated

**=================== END OF DELIVERABLE 2 =================**

**7. LOGICAL DESIGN PHASE:**

1. **Relational Database Schema**

Graphical user interface, application

Description automatically generated

**From diagram above write down all functional dependencies.**

Eg.,

orderid à custid, productid

1. **Normalization**

Include the below for all the tables.

Table Name: Author

State: 3NF

Reason: No multivalued attributes, No partial dependencies, No transitive dependency

All tables are expected to be in 3NF. Optionally you can choose higher normal forms also. Proper justification for choosing a higher normal form has to be provided.

*Note: Incase if there is any table not in 3NF, it has to be normalized to 3NF and changes made should be projected separately. Changes should also be updated in version table*

1. **CREATE DATA DICTIONARY**

**Table Definitions and Data Contents**

Table

Description automatically generated

**===================== END OF DELIVERABLE 3 =================**

**8.PHYSICAL DESIGN PHASE**

NOTE:

Please note the following points w.r.t to deliverable 4 or physical design phase submissions.

1. Each group member should be given a feature or a task in your group.

2. Each member should work on their feature and submit the Relational schema or Model diagram with FDs stated.

3. Since each of you must learn logical design and thereby your individual Relational schema/Relational model (RM) work for the feature allocated will ensure that all of you work and learn.

4. create a folder "DELIVERABLE 4" and in it submit all your individual deliverable4 file which has your allocated feature’s Implemented as SQL code and it should be stored as "<yourname>.sql".

1. SQL Statements

CREATE TABLE [dbo].[Author](

[Fname] [nvarchar](50) NOT NULL, [Minit] [nchar](1) NOT NULL, [Lname] [nvarchar](50) NOT NULL, [DOB] [date] NOT NULL,

CONSTRAINT [PK\_\_Author\_M\_\_EFC884A033D4B598] PRIMARY KEY CLUSTERED

(

[Fname] ASC,

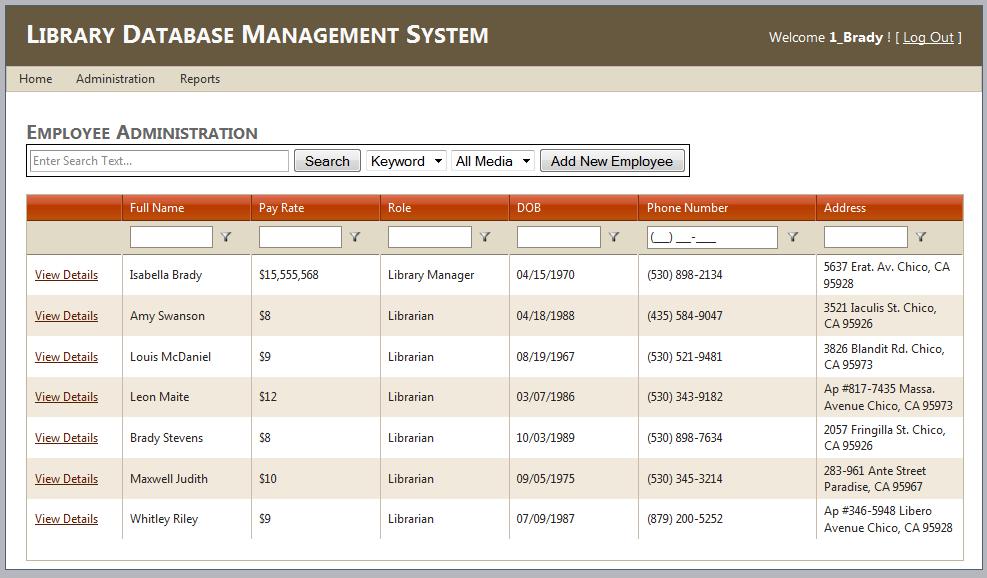
[Lname] ASC, [DOB] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY =OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

1. **Stored Procedures/ Triggers**

Include code for stored procedures/triggers if applicable

1. **User Interface and Database connectivity**
2. Your application should access DB through ODBC or Backend using appropriate technology and demonstrate the DB working.



1. **Using UI themes and project code**

Front end or console based logic or code

**===================== END OF DELIVERABLE 4 =================**

**Summary and statistics:**

**For 3 tier systems:**

For Backend:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Feature or functionality | API endpoint name | input | output |
|  |  |  |  |  |

For DB

DB objects used in your project

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Stored procedure |  |  |  |  |
| Triggers |  |  |  |  |
| JOBS |  |  |  |  |
| Tables |  |  |  |  |
| Views |  |  |  |  |
| Primary Indexes |  |  |  |  |
| Clustered Index |  |  |  |  |
| Secondary Indexes |  |  |  |  |
| Multilevel Indexes |  |  |  |  |
|  |  |  |  |  |

For Frontend

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Feature or functionality | API used | Input passed to API | Output  From API |
|  |  |  |  |  |

**For 1 or 2 tier systems:**

For DB

DB objects used in your project

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Stored procedure |  |  |  |  |
| Triggers |  |  |  |  |
| JOBS |  |  |  |  |
| Tables |  |  |  |  |
| Views |  |  |  |  |
| Primary Indexes |  |  |  |  |
| Clustered Index |  |  |  |  |
| Secondary Indexes |  |  |  |  |
| Multilevel Indexes |  |  |  |  |
|  |  |  |  |  |

Application End

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Feature or functionality | Function name | Input passed to function | Output  From function |
|  |  |  |  |  |

**12.**

REFERENCES:

1. For Library management System: mattben.info/media/pdf/370\_ImplementationPhase.pdf.

**13.** A DEMO VIDEO OF YOUR PROJECT for MAX 5 mins.

* Walk me through the features of the project
* Walk me through the code (ie., back and front end). Also, how the UI accesses the DB through SERVER.
* Finally, Application test cases, expected and actual behaviours and values obtained.

**==================== FINAL DELIVERABLE =================**

**DUE Dates :**

**Deliverable 1: 14th AUG 2023 RS DOC**

**Deliverable 2: 29th AUG 2023 CD**

**Deliverable 3: 9th SEPT 2023 LD**

**Deliverable 4: 20th OCT 2023 PROJECT FINAL CODE EXECUTABLE**

**Deliverable 5: 31st OCT 2023 FINAL COMPLETED DOCUMENT**

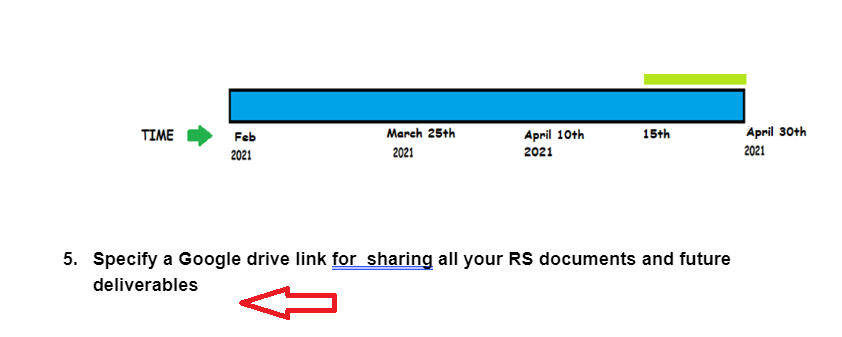
**Checklist for FINAL DOCUMENT for submission:**

* **The FINAL COMPLETED DOCUMENT named as**

**<your bitsid>\_<projectname>.pdf**

* **Make sure all 5 deliverables are in gdrive folder link.**
* **It should be accessible by me.**
* **This GDRIVE folder link is mentioned in 1 st page**

**Under the 5 item denoted by red arrow.**



* **Project code both front and back end are in gdrive folder.**
* **Demo video in gdrive for 5 mins max.**

**Once all are there in FINAL COMPLETED DOCUMENT, Submit this only document on taxila portal.**

==========================XXXXXX============================