

Higher Nationals

Internal verification of assessment decisions – BTEC (RQF)

INTERNAL VERIFICATION – ASSESSMENT DECISIONS			
Programme title	BTEC HND in Computing		
Assessor	Mrs. Ishani Jayasuriya	Internal Verifier	
Unit(s)	Unit 04: Database Design & Development		
Assignment title	Database Solution for Quiet Attic Films		
Student's name	Thakshila Jayasinghe		
List which assessment criteria the Assessor has awarded.	Pass	Merit	Distinction
INTERNAL VERIFIER CHECKLIST			
Do the assessment criteria awarded match those shown in the assignment brief?	Y/N		
Is the Pass/Merit/Distinction grade awarded justified by the assessor's comments on the student work?	Y/N		
Has the work been assessed accurately?	Y/N		
Is the feedback to the student: Give details: • Constructive? • Linked to relevant assessment criteria?	Y/N Y/N		

• Identifying opportunities for improved performance?	Y/N		
• Agreeing actions?	Y/N		
Does the assessment decision need amending?	Y/N		
Assessor signature		Date	
Internal Verifier signature		Date	
Programme Leader signature (if required)		Date	

Confirm action completed			
Remedial action taken Give details:			
Assessor signature		Date	
Internal Verifier signature		Date	
Programme Leader signature (if required)		Date	

Higher Nationals - Summative Assignment Feedback Form

Student Name/ID			
Unit Title	Unit 04: Database Design & Development		
Assignment Number	1	Assessor	
		Date	

Submission Date		Received 1st submission	
Re-submission Date		Date Received 2nd submission	

Assessor Feedback:

LO1 Use an appropriate design tool to design a relational database system for a substantial problem

Pass, Merit & Distinction Descriptors

P1 ☐ M1 ☐ D1 ☐

LO2 Develop a fully functional relational database system, based on an existing system design

Pass, Merit & Distinction Descriptors

P2 ☐ P3 ☐ M2 ☐ M3 ☐ D2 ☐

LO3 Test the system against user and system requirements.

Pass, Merit & Distinction Descriptors

P4 ☐ M4 ☐ D2 ☐

LO4 Produce technical and user documentation.

Pass, Merit & Distinction Descriptors

P5 ☐ M5 ☐ D3 ☐

Grade:	Assessor Signature:	Date:
---------------	----------------------------	--------------

Resubmission Feedback:

Grade:	Assessor Signature:	Date:
---------------	----------------------------	--------------

Internal Verifier's Comments:

Signature & Date:

* Please note that grade decisions are provisional. They are only confirmed once internal and external moderation has taken place and grades decisions have been agreed at the assessment board.

Assignment Feedback**Formative Feedback: Assessor to Student****Action Plan****Summative feedback**

Feedback: Student to Assessor			
Assessor signature		Date	
Student signature		Date	

**Pearson Higher Nationals in
Computing**

Unit 04: Database Design & Development
Assignment 01

General Guidelines

1. A Cover page or title page – You should always attach a title page to your assignment. Use previous page as your cover sheet and make sure all the details are accurately filled.
2. Attach this brief as the first section of your assignment.
3. All the assignments should be prepared using a word processing software.
4. All the assignments should be printed on A4 sized papers. Use single side printing.
5. Allow 1” for top, bottom, right margins and 1.25” for the left margin of each page.

Word Processing Rules

1. The font size should be **12** points, and should be in the style of **Time New Roman**.
2. **Use 1.5 line spacing**. Left justify all paragraphs.
3. Ensure that all the headings are consistent in terms of the font size and font style.
4. Use **footer function in the word processor to insert Your Name, Subject, Assignment No, and Page Number on each page**. This is useful if individual sheets become detached for any reason.
5. Use word processing application spell check and grammar check function to help editing your assignment.

Important Points:

1. It is strictly prohibited to use textboxes to add texts in the assignments, except for the compulsory information. e.g.: Figures, tables of comparison etc. Adding text boxes in the body except for the before mentioned compulsory information will result in rejection of your work.
2. Carefully check the hand in date and the instructions given in the assignment. Late submissions will not be accepted.
3. Ensure that you give yourself enough time to complete the assignment by the due date.
4. Excuses of any nature will not be accepted for failure to hand in the work on time.
5. You must take responsibility for managing your own time effectively.
6. If you are unable to hand in your assignment on time and have valid reasons such as illness, you may apply (in writing) for an extension.
7. Failure to achieve at least PASS criteria will result in a REFERRAL grade.
8. Non-submission of work without valid reasons will lead to an automatic REFERRAL. You will then be asked to complete an alternative assignment.
9. If you use other people's work or ideas in your assignment, reference them properly using HARVARD referencing system to avoid plagiarism. You have to provide both in-text citation and a reference list.
10. If you are proven to be guilty of plagiarism or any academic misconduct, your grade could be reduced to A REFERRAL or at worst you could be expelled from the course

Student Declaration

I hereby, declare that I know what plagiarism entails, namely to use another's work and to present it as my own without attributing the sources in the correct form. I further understand what it means to copy another's work.

1. I know that plagiarism is a punishable offence because it constitutes theft.
2. I understand the plagiarism and copying policy of Edexcel UK.
3. I know what the consequences will be if I plagiarise or copy another's work in any of the assignments for this program.
4. I declare therefore that all work presented by me for every aspect of my program, will be my own, and where I have made use of another's work, I will attribute the source in the correct way.
5. I acknowledge that the attachment of this document signed or not, constitutes a binding agreement between myself and Pearson, UK.
6. I understand that my assignment will not be considered as submitted if this document is not attached to the assignment.

E205866**Student's Signature:***(Provide E-mail ID)***12/10/2023****Date:***(Provide Submission Date)*

Higher National Diploma in Computing

Assignment Brief

Student Name /ID Number	E205866
Unit Number and Title	Unit 4: Database Design & Development
Academic Year	2022/23
Unit Tutor	Mrs.Ishani Jayasuriya
Assignment Title	Data base system for Quiet Attic Films
Issue Date	08/02/2023
Submission Date	10/12/2023
IV Name & Date	

Submission format

Part 1: The submission should be in the form of an individual written report written in a concise, formal business style using single spacing and font size 12. You are required to make use of headings, paragraphs and subsections as appropriate, and all work must be supported with research and referenced using Harvard referencing system. Please also provide in-text citation and bibliography using Harvard referencing system. The recommended word limit is 3,000–3,500 words, although you will not be penalised for exceeding the total word limit.

Part 2: The submission should be in the form of a fully functional relational database system demonstrated to the Tutor; and an individual written report (please see details in Part 1 above).

Part 3: The submission should be in the form of a witness statement of the testing completed by the Tutor; technical documentation; and a written report (please see details in Part 1 above).

Unit Learning Outcomes:

	<p>LO1 Use an appropriate design tool to design a relational database system for a substantial problem.</p> <p>LO2 Develop a fully functional relational database system, based on an existing system design.</p> <p>LO3 Test the system against user and system requirements.</p> <p>LO4 Produce technical and user documentation.</p>
	Assignment Brief and Guidance:

Assignment brief

Quiet Attic Films is a film production company based in London, England who specialize in making short information films and advertisements for television. They want you to design and implement a database that meets the requirements for their data. These requirements are specified in this scenario and the examples of paper documents kept by the company shown below.

Below are samples of the paper records currently kept by the company.

Initial Production Sheet at outset of a production

Production ID	Client	Locations	Production Type	Staff	Properties	Number of Days
2	Epom Motors	1. Greenwich Park, London 2. Windsor Castle Grounds	Advertisement	2 x Camera Crew 1 x Runner 1 x Producer 3 x Actors	Featured Vehicle for both locations. Wrecked Car for location 1.	5
6	Ministry of Agriculture, Fisheries and Food	1. River Cam, Cambridge 2. Quiet Attic Studio	Information Film	2 x Camera Crew 1 x Runner 1 x Producer 1 x Voice Actor	None	1
7	Ministry of Agriculture, Fisheries and Food	1. St James's Park, London	Training Film	2 x Camera Crew 1 x Runner 1 x Producer 1 x Voice Actor	Tractor	2

Quiet Attic Films organize their data around the concept of a 'production'. A production is specified as being for a particular client; but note that a client might have more than one production at any time. A production will take place at one or more locations. A production will also use a number of, what are called, properties, which might be anything from an actual property like a building, to costumes or small items of any sort. It is important to keep

a record of which properties are required at which location.

There should also be a record kept of the staff types that are assigned to productions.

NOTE: The above data is not properly normalised. Note repeating group data in location, properties and staff. Note that properties are shown with links to locations.

Standard Costs by Staff Type

Staff Type	Fee per production per day (£)
Camera Crew	100.00
Runner	25.00
Actor	200.00
Voice Actor	100.00
Producer	550.00

Property Types

Type	Examples
Vehicle	Car, Tractor, Boat etc.
Furniture	Chair, Table etc.
Building	Suburban House, Inner City House etc.

Activity 1

Identify the user and system requirements to design a database for the above scenario and design a relational database system using conceptual design (ER Model) by including identifiers (primary Key) of entities and cardinalities, participations of relationships.

Convert the ER Model into logical database design using relational database model including primary keys foreign keys and referential Integrities.

It should contain at least five interrelated tables. Check whether the provided logical design is normalised. If not, normalize the database by removing the anomalies.

(Note: -It is allowed to have your own assumptions and related attributes within the scope of the case study given)

Design set of simple interfaces to input and output for the above scenario using Wireframe or any interface-designing tool. Evaluate the effectiveness of the given design (ERD and Logical design) in terms of the identified user and system requirements.

Activity 2

Develop a relational database system according to the ER diagram you have created (Use SQL DDL statements). Provide evidence of the use of a suitable IDE to create a simple interface to insert, update and delete data in the database. Implement proper security mechanisms in the developed database and evaluate the database solution developed in terms of its effectiveness with relevance to the user and system requirements identified, system security mechanisms (EX: -User groups, access permissions) and the maintenance of the database. Suggest improvements for any identified problems.

Assess the usage of the below SQL statements with the examples from the developed database to prove that the data extracted through them are meaningful and relevant to the given scenario.

Select/ Where / Update / Between / In / Group by / Order by / Having

Activity 3

3.1 Provide a suitable test plan to test the system against user and system requirements. provide relevant test cases for the database you have implemented. Assess how the selected test data can be used to improve the effectiveness of testing.

Note: - Learner needs to give expected results in a tabular format and screenshots of the actual results with the conclusion

3.2 Get independent feedback on your database solution from the non-technical users and some developers (use surveys, questioners, interviews or any other feedback collecting method) and make a separate conclusion from the feedbacks.

Activity 4

Produce technical and user documentation for a fully functional system, including data flow diagrams showing movement of data through the system, and flowcharts describing how the system works. Evaluate the developed database by suggesting future enhancements to ensure the effectiveness of the system.

Acknowledgment!

I'm really appreciative to Database Design & Development
subject's outstanding lecturer
Head of Computing Department Mrs. Ishani Jayasuriya
for Ended our semester syllabus on time & also guiding
this Assignment properly.

Also acknowledged are all of the Lecturers, Management, &
The General Manager of the Esoft Metro campus Kandy
Mr. Dimuthu Thammitage.

Specially Thanks for Pearson UK. My parents and friends
were especially helpful to me throughout this period.
I appreciate your aid all!

Table of Contents

Activity 01.....	19
1.1 Identify the user and system requirements to design a database for the above scenario.	19
1.2 Design a relational database system using conceptual design (ER Model) by including identifiers (primary Key) of entities and cardinalities, participations of relationships.	24
1.3 Convert the ER Model into logical database design using relational database model including primary keys foreign keys and referential Integrities.	29
1.4 Check whether the provided logical design is normalised. If not, normalize the database by removing the anomalies.....	30
1.5 Design set of simple interfaces to input and output for the above scenario using Wireframe or any interface-designing tool.....	33
Activity 02.....	39
2.1 Develop a relational database system according to the ER diagram you have created (Use SQL DDL statements).....	39
2.2 Provide evidence of the use of a suitable IDE to create a simple interface to insert, update and delete data in the database.	42
2.3 Implement proper security mechanisms in the developed database and evaluate the database solution developed in terms of its effectiveness with relevance to the user and system requirements identified, system security mechanisms (EX: -User groups, access permissions).....	45
2.4 The maintenance of the database. Suggest improvements for any identified problems.	48
Activity 03.....	55
3.1 Provide a suitable test plan to test the system against user and system requirements.	55
3.2 provide relevant test cases for the database you have implemented.....	59
3.3 Assess how the selected test data can be used to improve the effectiveness of testing.	66
3.4 Get independent feedback on your database solution from the non-technical users and some developers (use surveys, questioners, interviews or any other feedback collecting method) and make a separate conclusion from the feedbacks.	66
Activity 04.....	71
4.1 Produce technical and user documentation for a fully functional system.....	71
4.2 Including data flow diagrams showing movement of data through the system, and flowcharts describing how the system works.	83
4.3 Evaluate the developed database by suggesting future enhancements to ensure the effectiveness of the system.	85
References	86

Table of Figure

Figure 1 Database.....	19
Figure 2 Entity RM.....	25
Figure 3 RM.....	26
Figure 4 ER Diagram.....	28
Figure 5 Logical DB.....	29
Figure 6 Login Form.....	33
Figure 7 Main Menu.....	34
Figure 8 Production DF.....	34
Figure 9 Properties DF.....	35
Figure 10 Client DF.....	35
Figure 11 Location DF.....	36
Figure 12 Staff DF.....	36
Figure 13 SQL Server.....	39
Figure 14 Client Table.....	40
Figure 15 Location & Staff Table.....	41
Figure 16 Properties Table.....	41
Figure 17 Production Table.....	41
Figure 18 Visual Studio.....	42
Figure 19 Login Form.....	43
Figure 20 Main Menu.....	43
Figure 21 Production DF.....	43
Figure 22 Properties DF.....	44
Figure 23 Client DF.....	44
Figure 24 Location DF.....	44
Figure 25 Staff DF.....	45
Figure 26 Database Security.....	46
Figure 27 Select & Where Statement.....	50
Figure 28 And Or & Not.....	51
Figure 29 Update Statement.....	52
Figure 30 Between Statement.....	52
Figure 31 IN statement.....	52
Figure 32 Group by & Having Statement.....	53
Figure 33 Order By Statement.....	53
Figure 34 Level of Testing.....	55
Figure 35 Feedback.....	66
Figure 36 Feedback Form.....	67
Figure 37 Feedback Responses.....	69
Figure 38 Documentation Types.....	71
Figure 39 Use case Diagram.....	77
Figure 40 Class Diagram.....	78
Figure 41 Sequence Diagram.....	79
Figure 42 Gantt Chart.....	80
Figure 43 Flow Chart.....	83

Table 1 Normalization	30
Table 2 Test Plan	58
Table 3 Test Case foe Login.....	59
Table 4 Test case for Main Menu.....	60
Table 5 Test case for PD	61
Table 6 Test case for PD	62
Table 7 Test case for CD	63
Table 8 Test case for SD	64
Table 9 Test case for LD	65

1. User requirements: are plain language statements and diagrams that outline the services that users should be able to obtain from the system as well as the limitations that must be met. The user requirements might range from general descriptions of the necessary system features to specific, in-depth explanations of the functionality of the system. These're the user requirements of above scenario.

- 1. User Authentication**
- 2. Client Management**
- 3. Location Management**
- 4. Staff Management**
- 5. Production Management**

2. System requirements: these are more thorough explanations of the features, services, and limitations of the software system. The system requirements document, also known as a functional specification, ought to specify precisely what needs to be put into practice. It might be covered in the agreement between the software developers and the system buyer. Functional and non-functional requirements are two common categories for software system requirements:

1. **Functional requirements:** These are declarations of the functions the system ought to perform, the way it ought to respond to certain inputs, and the way it ought to operate in specific circumstances. In some circumstances, the system's prohibited behaviors may also be stated clearly in the functional requirements.
2. **Non-functional requirements:** These are limitations on the features or services the system provides. These consist of standards-imposed limitations, time constraints, and development process constraints. Non-functional requirements frequently relate to the overall system rather than specific features or services.

(Collegenote, 2023).

1. User Authentication

(Functional Requirements)

- i. **User Registration:** Users It should be possible for users to register for an account using a special username and password.
- ii. **User Login User:** Sign in It must be possible for users to safely log in with their registered login credentials. To preserve user authentication during a session, employ session management.
- iii. **Password Recovery:** Make sure users have a way to retrieve their passwords, including email verification or security questions.

(Non-Functional Requirements)

- i. **Security:** Encrypt passwords and private user information. Use secure protocols (HTTPS) for communication
- ii. **Performance:** Make sure that the login and authentication processes respond quickly.
- iii. **Scalability:** Build the system to accommodate an increasing user base.

2. Client Managements

(Functional Requirements)

- i. **Registering as a Client:** Permit the entry of new customers with pertinent information.
- ii. **Updates to Client Information:** Give customers the option to amend their information.
- iii. **Client Lookup:** Incorporate a search feature to effectively locate and obtain customer data.

(Non-Functional Requirements)

- i. **Data Accuracy:** Make ensuring that the customer data is consistent and accurate.

- ii. **Control of Access:** Establish user roles and permissions to manage who can access client data.

- iii.

3. Location Management

(Functional Requirements)

- i. **Registration of Location:** Permit new places to be added with the required information.
- ii. **Change of Location:** Give users the option to update their location data.
- iii. **Location Lookup:** Put in place a location search feature.

(Non-Functional Requirements)

- i. **Data Correctness:** Make that the location data is accurate and consistent.
- ii. **Scalability:** Create the system to accommodate an increasing number of sites.

4. Staff Type

(Functional Requirements)

- i. **Staff Type Definition:** Permit the definition of several personnel categories together with the corresponding roles and authorizations.
- ii. **Staff Type Assignment:** Assignment Permit users to have different staff types assigned to them

(Non-Functional Requirements)

- i. **Access Control:** Based on staff kinds, specify access permissions.

5. Production Management

(Functional Requirements)

- ii. **Definition of a Product:** Permit the defining of various goods and services.
- iii. **Production Planning:** Put in place production schedule planning functionality.
- iv. **Inventory Management:** Incorporate inventory management and tracking features.

(Non-Functional Requirements)

- i. **Performance:** Make sure that massive amounts of production data are handled effectively.
- ii. **Scalability:** Build the system to accommodate an increasing quantity of items and production information.

Although two specification of system requirement. As,

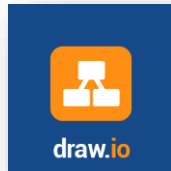
1. Software Specification

The software's that used to implement the Assignment in the below,

Microsoft Word



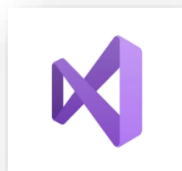
Draw.io



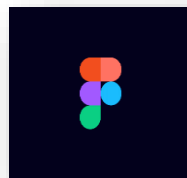
Microsoft SQL Server



Visual Studio



Figma



2. Hardware Specification

Hardware devices which use to develop the Assignment.

Acer Core i3 Laptop

Processor

Intel Core i5-10210U 4 x 1.6 - 4.2 GHz, Comet Lake-U

Graphics adapter

NVIDIA GeForce MX250 - 2048 MB VRAM, Core: 1519 MHz, Memory: 1502 MHz, GDDR5, Nvidia 430.90, Optimus Intel UHD Graphics

Memory

8 GB, 4 GB soldered, 1x 4 GB module, 1 slot, up to 20 GB

Display

15.60 inch 16:9, 1920 x 1080 pixel 141 PPI, LG Display LP156WFC SPD1, IPS, glossy: no, 60 Hz

1.2 Design a relational database system using conceptual design (ER Model) by including identifiers (primary Key) of entities and cardinalities, participations of relationships.

DBMS - Data Models

The logical structure of a database is modelled according to data models. In a DBMS, data models are essential components for introducing abstraction. Data models specify the relationships between data as well as how the data are handled and kept within the system. Flat data models, in which all the data are stored in a single plane, may be the first type of data model. Because the earlier data models were less scientific, they were more likely to introduce update anomalies and a great deal of duplication.

1. Entity-Relationship Model

The Entity-Relationship (ER) Model is predicated on the idea of actual entities and the connections between them. The Entity Relationship Model (ER Model) establishes entity sets, relationship sets, general attributes, and restrictions while incorporating real-world scenarios into the database model. The optimal application of the ER Model is in conceptual database design. ER Model's foundation is

- A. Entities and their attributes.
- B. Relationships among entities.

These concepts are explained below.

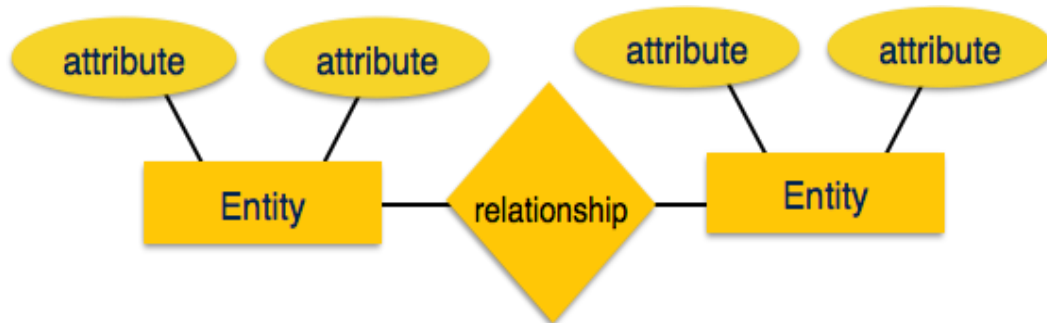


Figure 2 Entity RM

Entity – A real-world entity with properties referred to as attributes is called an entity in an ER model. Each attribute has a domain, or collection of values, that define it. For instance, a student is regarded as an entity in a school database. A student's name, age, class, and other characteristics are among them.

Relationship – Relationships are the logical associations that exist between entities. Different mappings are made between relationships and entities. The number of associations between two things is defined by mapping cardinalities.

Mapping cardinalities.

1. one to one
2. one to many
3. many to one
4. many to many

2.Relational Model

In DBMSs, the Relational Model is the most widely used data model. This approach is more grounded in science than others. This paradigm describes a table as an n-ry relation and is based on first-order predicate logic.

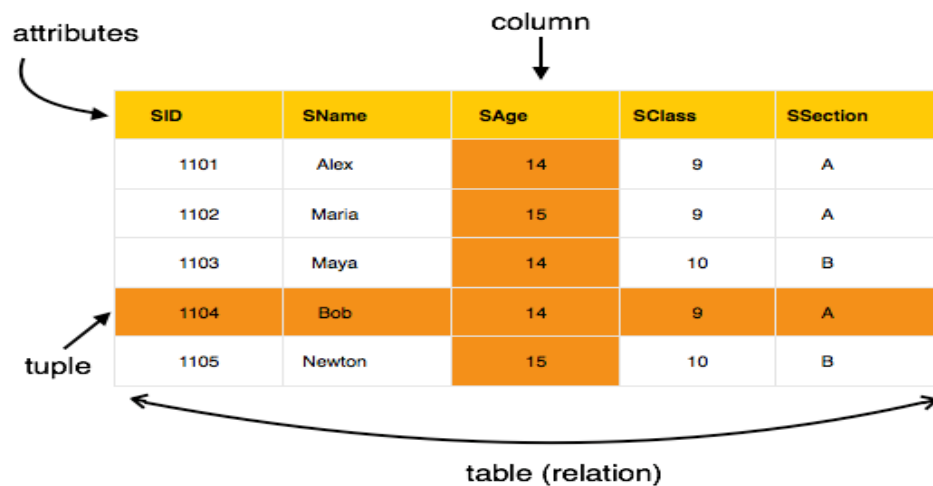


Figure 3 RM

The main highlights of this model are,

- i. Data is stored in tables called relations.
- ii. Relations can be normalized.
- iii. In normalized relations, values saved are atomic values.
- iv. Each row in a relation contains a unique value.
- v. Each column in a relation contains values from a same domain.

ER Diagram of Quiet Attic Films in London!

Entities

1. Production
2. Client
3. Properties
4. Location
5. Staff Type

Attributes

1. Production_ID (PK) / Production Name/ Production Type/ Production Days/
Client_ID(FK)
2. Client ID(PK)/ Client Name/ Contact {Multivalve Attribute}

3. Properties ID(PK)/ Property_Name/ Properties Type/ Client_ID(FK)/ Location_ID(FK)
4. Location_ID(PK)/ Location_Name/ Address
5. Staff_ID(PK)/ Staff_Name/ Production_ID(FK)

Cardinality of Relationship

1. Production & Client has One to One Relationship (1 – 1)
2. Production & Location has One to Many Relationship (1 – M)
3. Production & Staff Type has One to Many Relationship (1 – M)
4. Production & Properties has One to Many Relationship (1 – M)

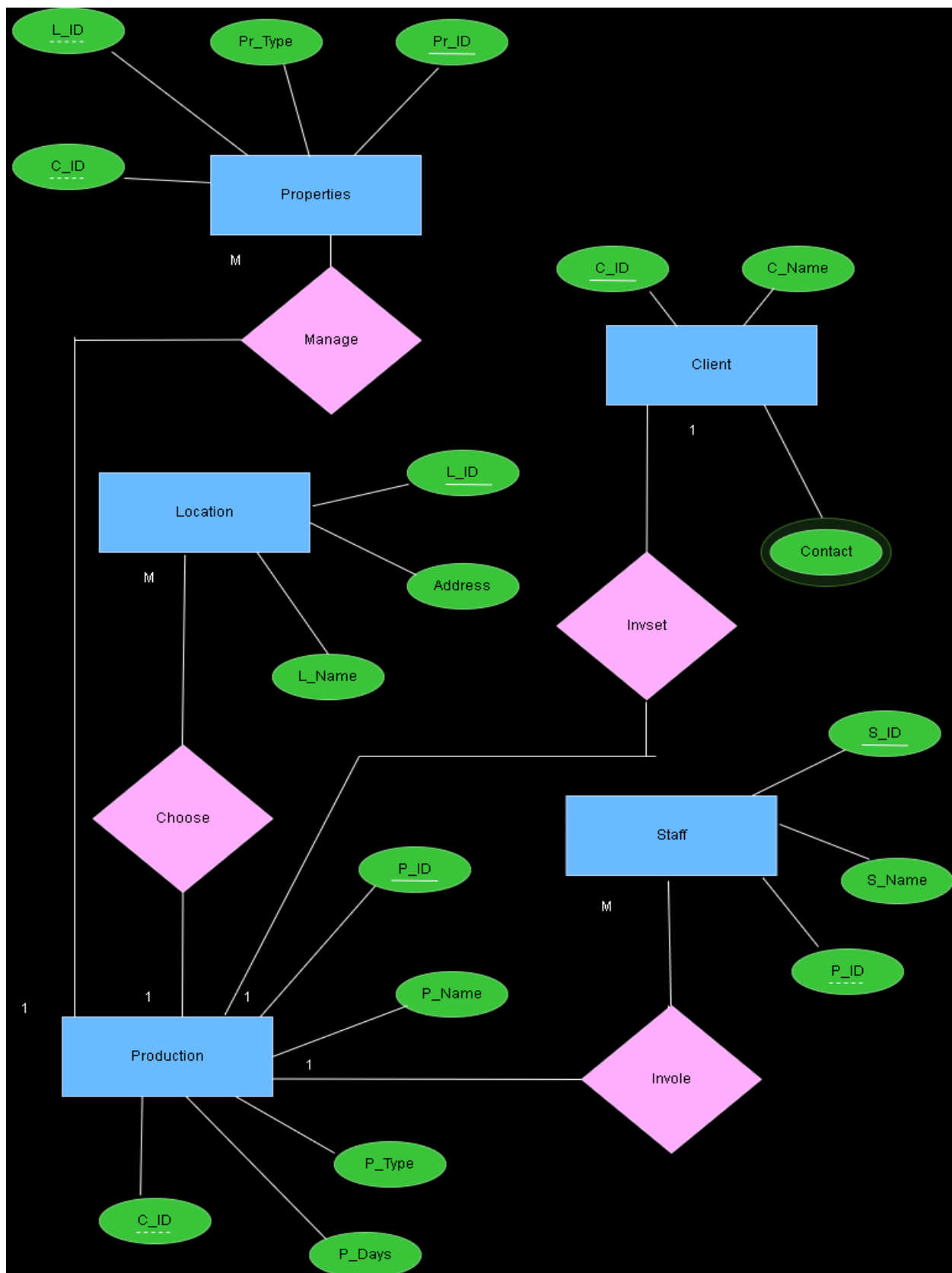


Figure 4 ER Diagram

1.3 Convert the ER Model into logical database design using relational database model including primary keys foreign keys and referential Integrities.

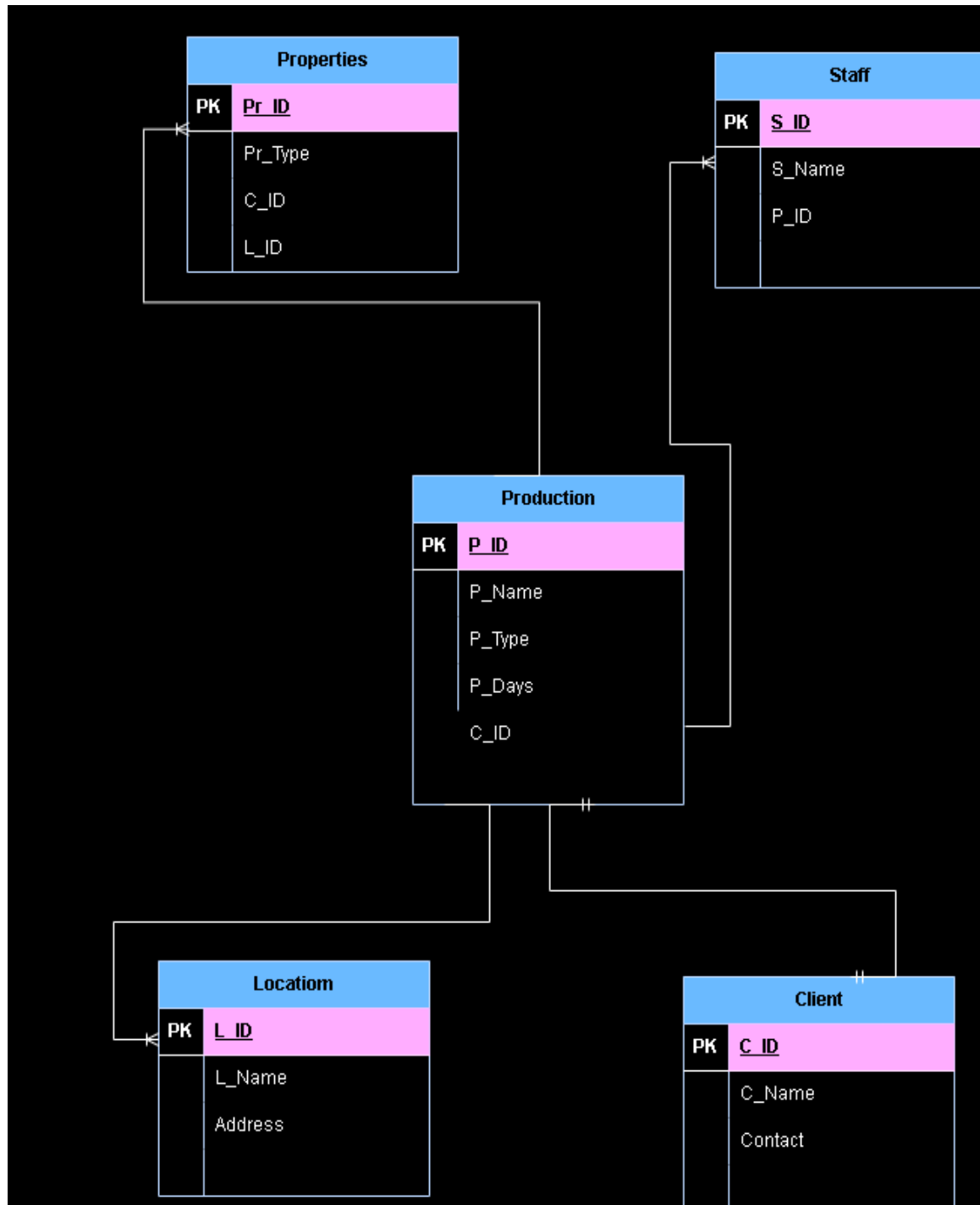


Figure 5 Logical DB

1.4 Check whether the provided logical design is normalised. If not, normalize the database by removing the anomalies.

What is Normalization?

The process of arranging data into tables so that queries executed on the database always return clear and intended results is known as database normalization. The foundation of relational database theory is this kind of normalization. It frequently leads to the formation of extra tables and may have the consequence of duplicating data within the database.

The relational database model was first described in a paper by IBM researcher E.F. Codd in 1970, and this is usually acknowledged as the source of the concept of database normalization. An integral part of the relational technique was what Codd referred to as "a normal form for database relations". In the 1970s and 1980s, when disc drives were highly expensive and highly efficient data storage was essential, such data normalization found a ready audience. Since then, other methods have also gained popularity, such as denormalization.

Data normalization rules

Data redundancy, or superfluous duplication, is not introduced by data normalization standards, despite the fact that they tend to increase data duplication. After the initial exercise of identifying the data objects that should be in the relational database, establishing the necessary tables and the columns inside each table, and determining their relationships, database normalization is usually a refinement process.

Data normalization example

Table 1 Normalization

Customer	Item purchased	Purchase price
Thomas	Shirt	\$40
Maria	Tennis shoes	\$35
Evelyn	Shirt	\$40
Pajaro	Trousers	\$25

In the event that the user wishes to remove a client from the table that is used to track item prices, the price will also be removed. In order to normalize the data, this table would need to be divided into two tables: one including details about each customer and the product they purchased, and the other containing information about each product and its price. Any changes made to one table would not have an impact on the other.

1. First normal form (1NF).

This is the "basic" level of database normalization, and it generally corresponds to the definition of any database, namely:

- 1) It contains two-dimensional tables with rows and columns.
- 2) Each column corresponds to a sub object or an attribute of the object represented by the entire table.
- 3) Each row represents a unique instance of that sub object or attribute and must be different in some way from any other row (that is, no duplicate rows are possible).
- 4) All entries in any column must be of the same kind. For example, in the column labeled "Customer," only customer names or numbers are permitted.

2.Second normal form (2NF).

Each column in a table at this level of normalization needs to be a function of the other columns in the table in order to avoid being a determiner of the contents of another column. For instance, the pricing would depend on the customer ID (which is eligible for a discount) and the particular product in a table with three columns that included the customer ID, the product sold, and the product's price at the time of sale. In this case, it is claimed that the information in the first and second columns influences the information in the third column. The 1NF scenario does not involve this dependency.

Because it uniquely identifies the rows in that table and satisfies the other acceptable conditions in the standard database management design, the column labelled customer ID is regarded as a primary key. Its values are not NULL and they won't fluctuate over time. The remaining column headers in the example above are regarded as potential keys. Prime attributes are those characteristics of those candidate keys that set them apart.

3.Third normal form (3NF).

Because an update to one row in a table may impact data that references this information from another table, modifications are still feasible at the second normal form. For instance, in the customer table previously mentioned, eliminating a row that details a client's purchase (perhaps due to a return) will also eliminate the information about the product's pricing. These tables would be split into two tables in the third normal form so that product price could be tracked independently.

The domain/key normalized form, which uses a key to uniquely identify each row in a table, and the Boyce-Codd normal form (BCNF), which improves and refines the methods employed in the 3NF to handle specific kinds of anomalies, are examples of extensions of basic normal forms.

(Rouse, 2019).

Staff Table

S_ID	S_Type (Name)	P_ID
------	---------------	------

Staff Type Table

S_ID	S_Type (Name)
------	---------------

Staff Production Table

S_ID	P_ID
------	------

Staff table normalize using 3NF because P_ID repeating in staff table.

1.5 Design set of simple interfaces to input and output for the above scenario using Wireframe or any interface-designing tool.

What is a Wireframe?

Basic blueprints called wireframes aid in requirement alignment, focusing and advancing productive UX design discussions within teams. Consider your wireframe to be the basic structure of your application, website, or other finished product. The design team and stakeholders may see in your wireframe the basic layouts of key pages, elements, and functionality, such as:

Used **Figma wireframe interface designing tool** for create these interfaces.

According to the scenario the following are the functionalities of the system!

1. Login Form
2. Main Menu
3. Production Details
4. Properties Details
5. Client Details
6. Location Details
7. Staff Details

1. Login Form



Figure 6 Login Form

2.Main Form

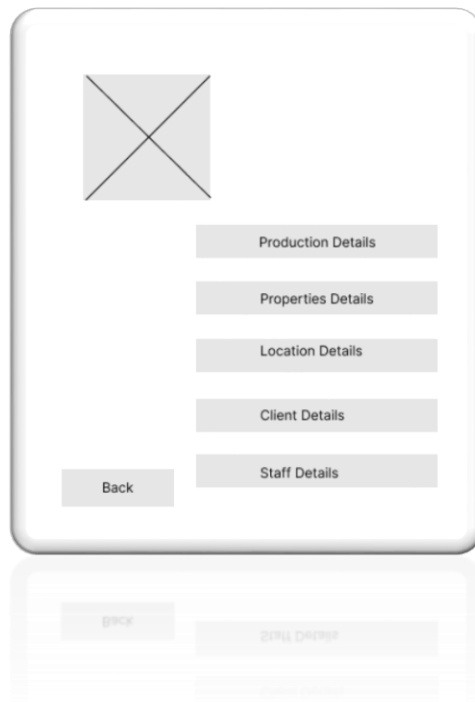


Figure 7 Main Menu

3.Production Details Form

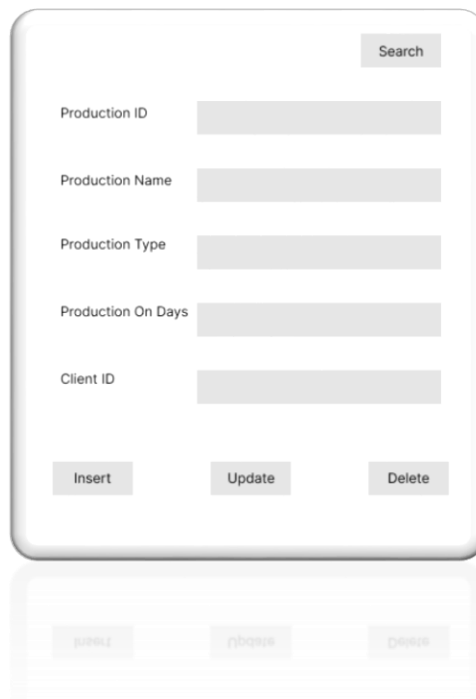
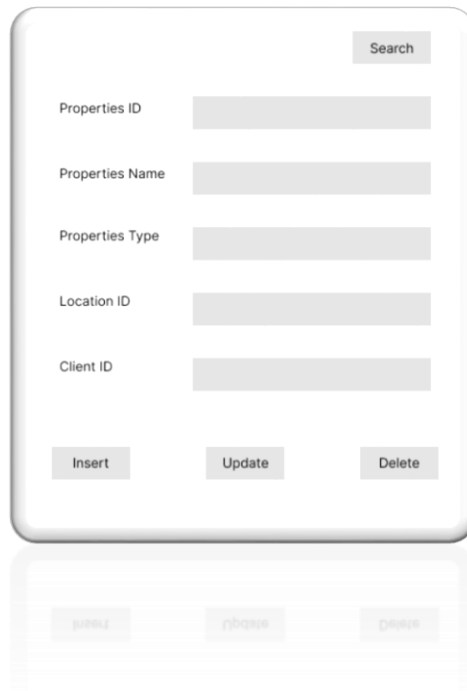


Figure 8 Production DF

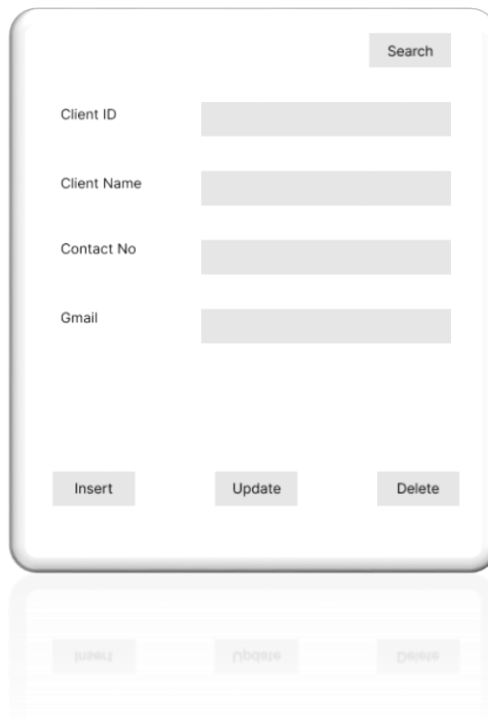
4.Properties Details Form



The Properties Details Form is a web-based interface for managing property records. It features a search bar at the top right. Below the search bar, there are five input fields for data entry: Properties ID, Properties Name, Properties Type, Location ID, and Client ID. At the bottom of the form, there are three buttons: Insert, Update, and Delete. The form is displayed on a light gray background with a subtle reflection effect below it.

Figure 9 Properties DF

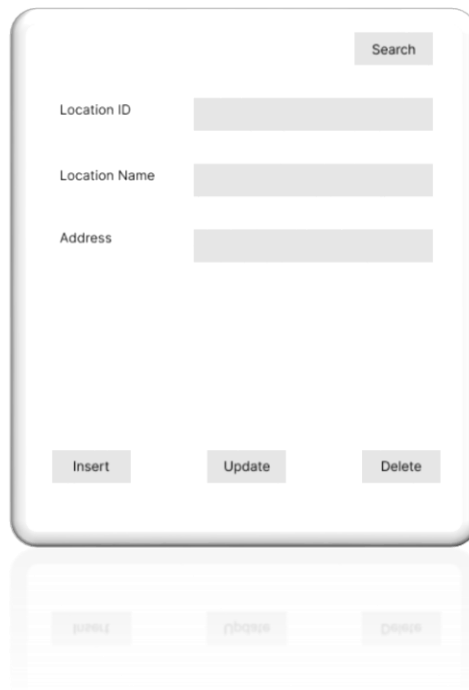
5.Client Details Form



The Client Details Form is a web-based interface for managing client records. It features a search bar at the top right. Below the search bar, there are four input fields for data entry: Client ID, Client Name, Contact No, and Gmail. At the bottom of the form, there are three buttons: Insert, Update, and Delete. The form is displayed on a light gray background with a subtle reflection effect below it.

Figure 10 Client DF

6.Location Details Form



The Location Details Form is a web-based interface for managing location data. It features a search bar at the top right with a 'Search' button. Below the search bar are three input fields: 'Location ID', 'Location Name', and 'Address'. At the bottom of the form are three buttons: 'Insert', 'Update', and 'Delete'. The form is displayed on a light gray background with a reflection effect below it.

Figure 11 Location DF

07.Staff Details Form



The Staff Details Form is a web-based interface for managing staff data. It features a search bar at the top right with a 'Search' button. Below the search bar are three input fields: 'Staff ID', 'Staff Type', and 'Production ID'. At the bottom of the form are three buttons: 'Insert', 'Update', and 'Delete'. The form is displayed on a light gray background with a reflection effect below it.

Figure 12 Staff DF

1.5 Evaluate the effectiveness of the given design (ERD and Logical design) in terms of the identified user and system requirements.

Analysing how well a design conforms to the stated needs and objectives is a crucial step in determining the efficacy of an Entity Relationship Diagram (ERD) and Logical Design with respect to recognised user and system requirements.

Verify that the ERD and Logical Design have addressed every specified user and system requirement. The requirements and the design aspects should be clearly traceable to one another.

1. Consistency: Verify that the design complies with the stated specifications.
Relationships that are in conflict with one another or characteristics that differ from what the system and users require should not exist.
2. Completeness: Determine if the ERD and logical design contain all pertinent information. Verify that no essential needs were missed or left out.
3. Normalization: Assess the design's normalization when it comes to relational databases. Verify the tables' proper normalization to cut down on redundancy and guarantee data integrity.
4. Adequacy of Relationships: Verify that the relationships between entities appropriately reflect the affiliations mentioned in the requirements by looking at their relationships. Relationships should appropriately reflect the restrictions and norms of the company.
5. Data Integrity: Verify whether the design incorporates key and foreign key methods, among other measures, to enforce data integrity. This guarantees that throughout the system, the data is correct and consistent.
6. Performance Considerations: Assess the design's functionality. Make that it satisfies the users' defined performance requirements, including response time and throughput.

7. **Flexibility and Scalability:** Evaluate if the design is adaptable enough to future modifications and has enough room to grow as users and data volumes do.
8. **User Feedback:** To learn about end users' opinions on the design, get their input. This can offer insightful information about whether the design satisfies their needs for usability and their expectations.
9. **Documentation:** Examine the ERD and Logical Design-related documents. It should be thorough and unambiguous, offering insights into the design choices and how they fit with the specifications.

Activity 02

2.1 Develop a relational database system according to the ER diagram you have created (Use SQL DDL statements).

What is SQL (Structured query language)?



Figure 13 SQL Server

A programming language called structured query language (SQL) is used to store and process data in relational databases. Information is stored in tabular form in relational databases, where distinct data properties and the numerous relationships between the data values are represented by rows and columns. Information can be stored, updated, removed, searched for, and retrieved from databases using SQL commands. SQL can also be used to optimize and maintain database performance.

(Amazon, 2023).

SQL categories in to 03 types,

1. DDL

Data Definition Language (DDL) statements are used to define the database structure or schema. Some examples:

CREATE - To create objects in the database

ALTER - Alters the structure of the database

DROP - Delete objects from the database

RENAME - Rename an object

2. DML

Data Manipulation Language (DML) statements are used for managing data within schema objects. Some examples:

SELECT - Retrieve data from a database

INSERT - Insert data into a table

UPDATE - Updates existing data within a table

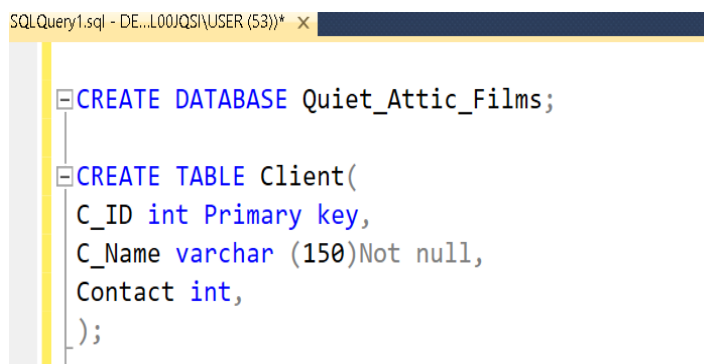
DELETE - Deletes all records from a table, the space for the records remains

3. DCL

Data Control Language (DCL) statements. Some examples:

GRANT - Gives user's access privileges to database

REVOKE - Withdraw access privileges given with the GRANT command



```
SQLQuery1.sql - DE...L00JQSI\USER (53))* x
CREATE DATABASE Quiet_Attic_Films;
CREATE TABLE Client(
  C_ID int Primary key,
  C_Name varchar (150)Not null,
  Contact int,
);
```

Figure 14 Client Table


```
CREATE TABLE Location(  
  L_ID int Primary key,  
  L_Name varchar (150)Not null,  
  Addresss varchar(100),  
);  
  
CREATE TABLE Staff(  
  S_ID int Primary key,  
  S_Name varchar (150)Not null,  
);
```

Figure 15 Location & Staff Table

```
CREATE TABLE Properties(  
  Pr_ID int Primary key,  
  Pr_Name varchar (150)Not null,  
  Pr_Type varchar (100),  
  L_ID int Foreign key references Location(L_ID),  
  C_ID int Foreign key references Client(C_ID)  
);
```

Figure 16 Properties Table

```
CREATE TABLE Production(  
  P_ID int Primary key,  
  P_Name varchar (150)Not null,  
  P_Type varchar (100),  
  P_OnDays int,  
  C_ID int Foreign key references Client(C_ID)  
);
```

Figure 17 Production Table

2.2 Provide evidence of the use of a suitable IDE to create a simple interface to insert, update and delete data in the database.

Microsoft created **Visual Studio**, an Integrated Development Environment (IDE) for creating desktop, console, online, mobile, cloud, and web services applications, among other types of applications. You may write both native and managed code with the aid of this IDE. It makes use of the several Microsoft software development platforms, including Windows API, Microsoft Silverlight, and Windows Store. You can write code in several different languages using this integrated development environment (IDE), including Python, JavaScript, C#, C++, and VB (Visual Basic). Thirty-six different programming languages are supported. It may be downloaded for both macOS and Windows.

Used visual Studio 2022 for create these interfaces. & these are the

Innovation Features of Visual Studio 2022!

1. C++
2. Diagnostics and debugging
3. Real-time collaboration
4. Insights and productivity
5. Asynchronous collaboration++
6. Improved code search

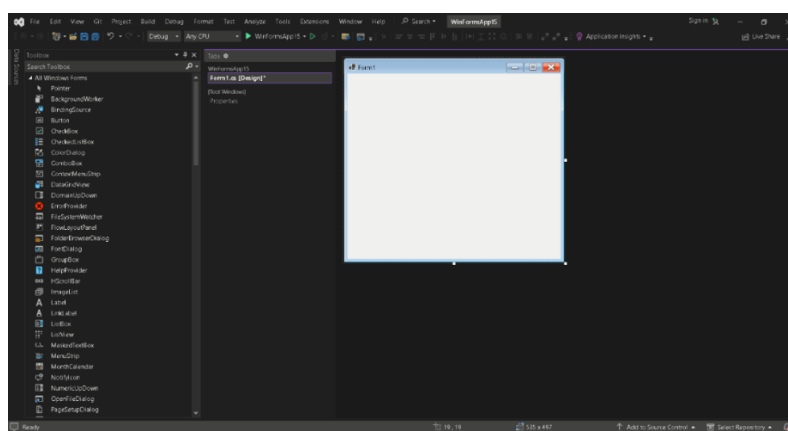


Figure 18 Visual Studio

1. Login Form

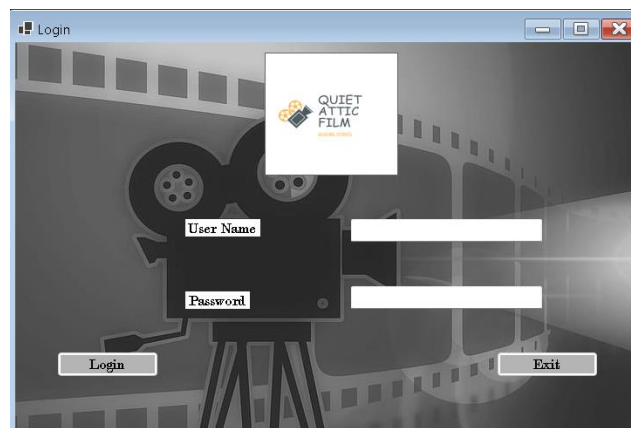
The screenshot shows a window titled "Login" with a film-themed background. In the top center is the "QUIET ATTIC FILM" logo. Below it are two input fields: "User Name" and "Password". At the bottom left is a "Login" button, and at the bottom right is an "Exit" button.

Figure 19 Login Form

2. Main Form

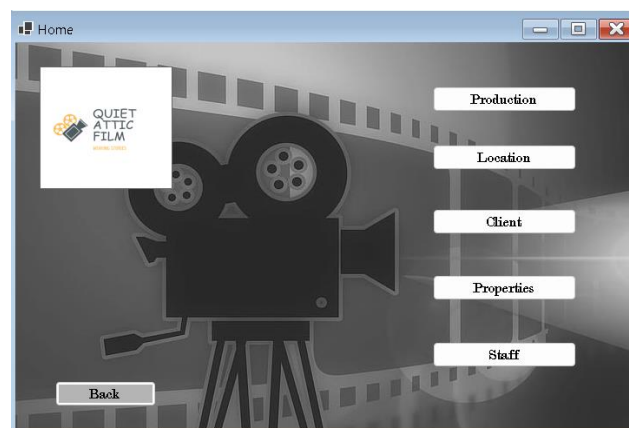
The screenshot shows a window titled "Home" with a film-themed background. On the left is the "QUIET ATTIC FILM" logo. On the right is a vertical list of menu items: "Production", "Location", "Client", "Properties", and "Staff". At the bottom left is a "Back" button.

Figure 20 Main Menu

3. Production Details Form

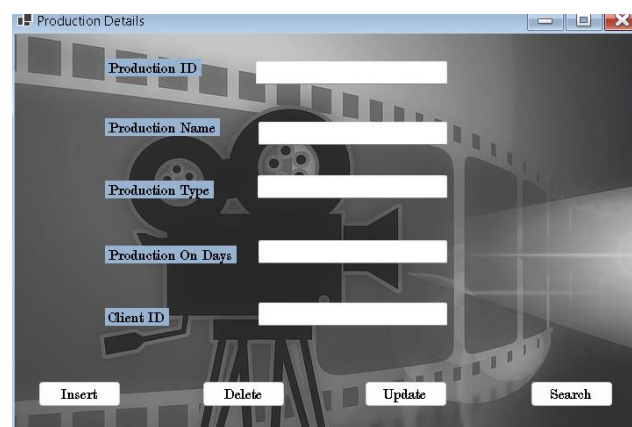
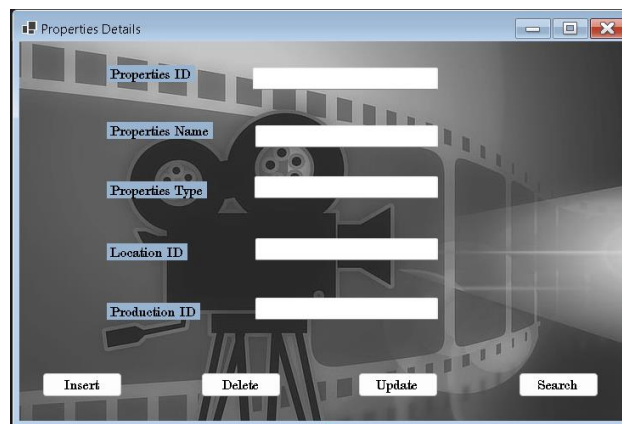
The screenshot shows a window titled "Production Details" with a film-themed background. It contains five input fields with labels: "Production ID", "Production Name", "Production Type", "Production On Days", and "Client ID". At the bottom are four buttons: "Insert", "Delete", "Update", and "Search".

Figure 21 Production DF

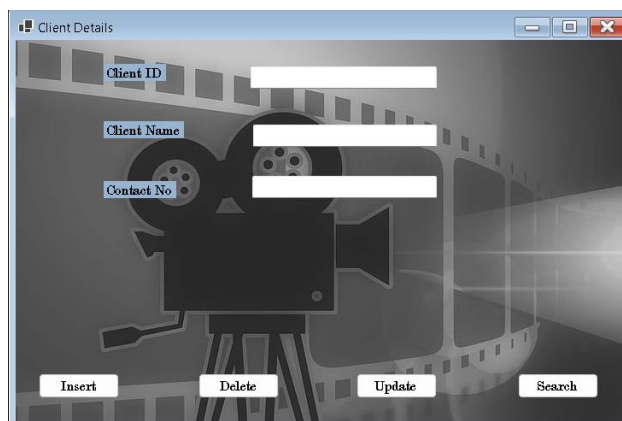
4.Properties Details Form



The Properties Details Form is a software window titled "Properties Details". It features a background image of a movie camera on a tripod. The form contains five text input fields with labels: "Properties ID", "Properties Name", "Properties Type", "Location ID", and "Production ID". At the bottom of the form, there are four buttons: "Insert", "Delete", "Update", and "Search".

Figure 22 Properties DF

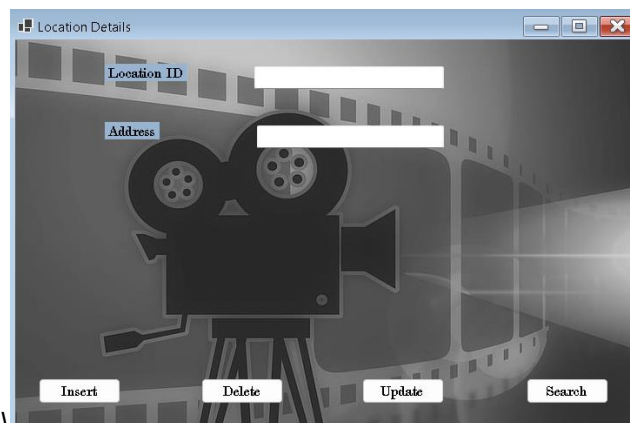
5.Client Details Form



The Client Details Form is a software window titled "Client Details". It features a background image of a movie camera on a tripod. The form contains three text input fields with labels: "Client ID", "Client Name", and "Contact No.". At the bottom of the form, there are four buttons: "Insert", "Delete", "Update", and "Search".

Figure 23 Client DF

6.Location Details Form



The Location Details Form is a software window titled "Location Details". It features a background image of a movie camera on a tripod. The form contains two text input fields with labels: "Location ID" and "Address". At the bottom of the form, there are four buttons: "Insert", "Delete", "Update", and "Search".

Figure 24 Location DF

07.Staff Details Form

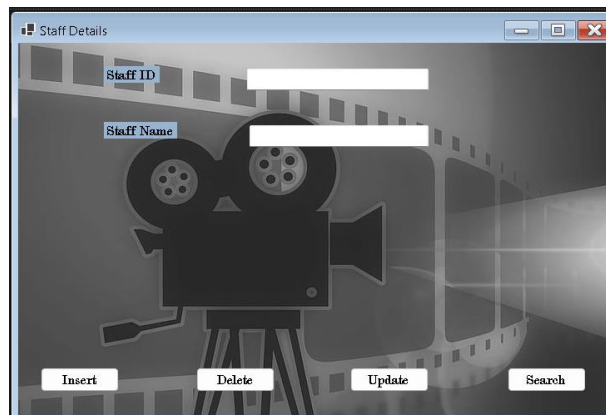


Figure 25 Staff DF

2.3 Implement proper security mechanisms in the developed database and evaluate the database solution developed in terms of its effectiveness with relevance to the user and system requirements identified, system security mechanisms (EX: -User groups, access permissions).

What is Database Security?

A range of techniques are used in database security to protect database management systems from unauthorized access and malevolent cyberattacks. Database security programmes are made to guard against misuse, damage, and intrusion not just for the data in the database but also for the data management system and all applications that utilize it.

Database security refers to the methods, procedures, and instruments used to create security inside a database environment.

How Can You Secure Your Database Server?

A real or virtual machine that hosts a database is called a database server.

Hardening, another name for database server security, is the process of securing a database server through safe operating system configuration, network security, and physical protection.

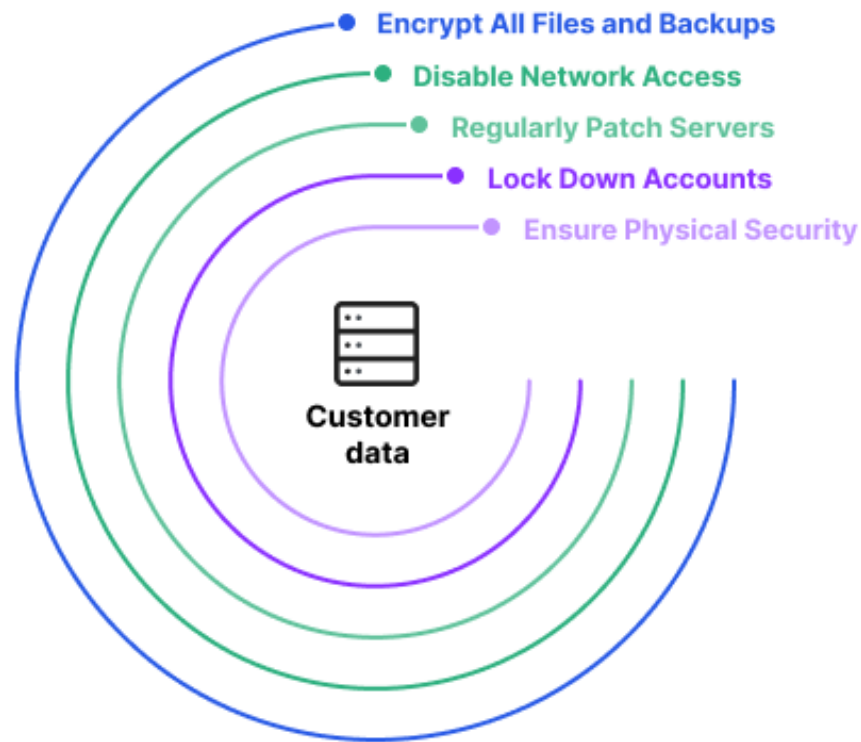


Figure 26 Database Security

1. Ensure Physical Database Security

If sensitive information is stored in your database, do not share a server with online apps. While putting your website and database on a hosting company may be easier and less expensive, you are entrusting someone else with the security of your data.

Should you want to handle your database through a web hosting service, make sure the provider has a solid security record. Given the potential lack of security, it is advisable to avoid using free hosting services.

Remember that managing your database in an on-premise data center exposes you to both internal and external attack dangers. Make sure your physical facility has locks, cameras, and security guards among other physical security measures. Physical server access needs to be tracked and restricted to those who are authorized. Furthermore, avoid storing database backups in publicly accessible places like web files, temporary partitions, or unprotected cloud storage buckets.

2. Lock Down Accounts and Privileges

Let's think about the database server, Oracle. The Oracle Database Configuration Assistant (DBCA) automatically expires and locks the majority of the default database user accounts after the database is installed.

This doesn't happen and default privileged accounts won't expire or lock if an Oracle database is installed manually. By default, their password and username remain the same. The first attempt to connect to the database will be made by an attacker using these credentials.

Ensuring that each privileged account on a database server has a strong and distinct password is of utmost importance. Accounts that are no longer needed should be closed and locked. Access to the remaining accounts must be restricted to the bare minimum. The tables and actions (like SELECT and INSERT) that each account needs should be limited to those that the user specifically requests. Refrain from setting up user accounts that have access to all database tables.

3. Regularly Patch Database servers

Make sure the patches are up to date. Because hackers are constantly looking for new security holes in databases and because malware and viruses are constantly being released, effective database patch management is an essential security procedure.

The stability and performance of databases will be enhanced by the timely distribution of key security hotfixes, cumulative updates, and database service packs.

4. Disable Public Network Access

Applications are kept in databases by organizations. In the majority of real-world situations, direct database access is not necessary for the end user. Therefore, unless you are a hosting provider, you should prevent any public network access to database servers. For remote administrators, an organization should ideally set up gateway servers (VPN or SSH tunnels).

5. Encrypt All Files and Backups

No matter how strong your defences are, hackers could still manage to get access to your system. However, there are other threats to your database's security besides attackers. Your company can potentially be at risk from your staff. An insider who is malevolent or negligent could potentially obtain access to a file to which they are not authorized.

Your data becomes illegible for both employees and attackers when it is encrypted. They are unable to access it without an encryption key, offering a last barrier of protection against unauthorized access. Protect your vital data by encrypting backups, data files, and essential application files to prevent unauthorized access.

(Imperva, 2023).

2.4 The maintenance of the database. Suggest improvements for any identified problems.

Optimize Queries

SQL queries that are not efficient are frequently the root cause of problems with database speed. One of the best ways to improve database speed is to optimize your SQL queries. Choosing the best way to increase query efficiency may present you with a number of challenges if you attempt to accomplish that manually.

These include knowing when to use EXISTS or IN, writing joins or subqueries, and more. You can build queries that increase efficiency and, consequently, overall database performance once you know which way to go. Less bottlenecks and disgruntled end users result from this.

Using a database performance analysis tool is the best way to optimize queries because it can help you identify the most wasteful queries and provide professional guidance on how to improve them.

Improve Indexes

The index is another crucial component of the database, in addition to queries. When indexing is done correctly, it can improve query execution time and boost database performance. By creating a data structure, indexing helps you keep all of your data organized and facilitates finding information. Indexing makes data easier to find, which

improves the effectiveness of data retrieval and speeds up the process overall, saving time and effort for both you and the system.

Defragment Data

One of the finest methods for improving database speed is data defragmentation. Your database may eventually become fragmented due to the volume of data that is continuously added to and removed from it. Because fragmentation makes it more difficult for a query to find the information it needs quickly, it might slow down the process of retrieving data. material defragmentation removes index page problems and enables pertinent material to be grouped together. This implies that your I/O-related processes will execute more quickly.

Increase Memory

When you don't have enough memory available for the database to function properly, its efficiency can suffer greatly. You may not be able to satisfy the demands of your database even though it appears that you have a lot of RAM overall. Examining the number of page faults in your system is a useful method to determine whether you require additional memory. A significant number of errors indicates that your hosts' memory is either running low or entirely exhausted. Increasing the amount of memory, you allocate will improve performance and efficiency.

Strengthen CPU

A more efficient database is directly correlated with a better CPU. That's why, if your database performance is a problem, you should think about upgrading to a higher-class CPU unit. Your CPU will be less stressed when handling numerous requests and programmes if it has greater power. All aspects of CPU performance should be monitored when evaluating your CPU, including CPU ready times, which inform you of the instances in which your system attempted to use the CPU but was unable to do so because other resources were being used.

Review Access

Examine your database access, including which apps are really accessing your database, as soon as you are certain that your database hardware is operating as intended. It's crucial to

avoid drawing hasty assumptions about which of your services or applications is to blame if one of them is experiencing poor database performance. While it's possible that just one client is having problems, it's also possible that the database is having problems as a whole. Examine who and what is gaining access to the database, and if there is just one service experiencing problems, examine its metrics in detail to try to identify the underlying cause. (Staff Contributor, 2023).

2.4 Assess the usage of the below SQL statements with the examples from the developed database to prove that the data extracted through them are meaningful and relevant to the given scenario.

Select/ Where / Update / Between / In / Group by / Order by / Having

1. Select/ Where Statements

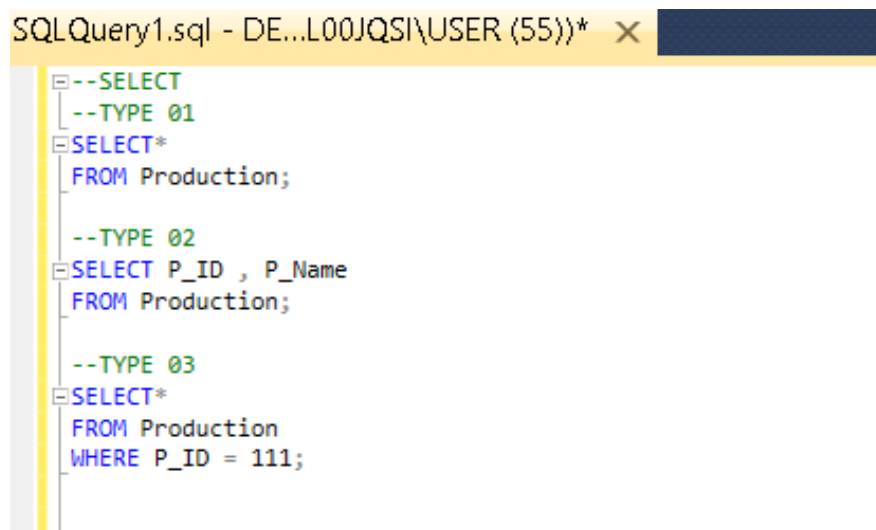
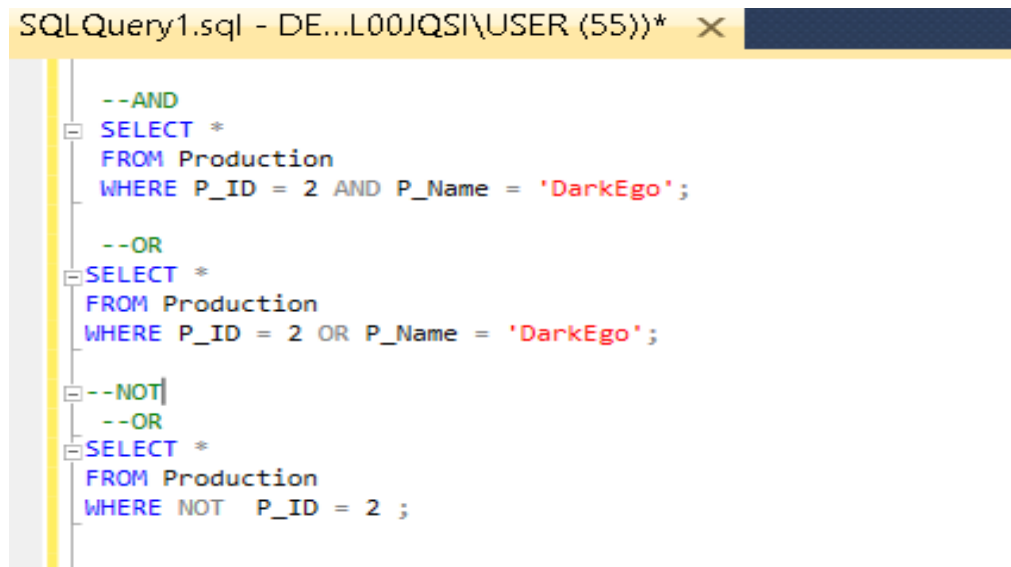


Figure 27 Select & Where Statement

You can write a SQL query in a variety of ways to get the precise data you want, and you can use the SQL **SELECT** statement to retrieve data from a specific database table. A table containing these data is returned when they are taken out of the database table.

1. **Type 01:** It could be necessary to choose every column in a database. Use of the asterisk (*) character is an alternative to listing each column.
2. **Type 02:** Alternatively, by listing the columns and dividing them with commas, you can obtain specific columns rather than all of the columns:

3. **Type 03:** It could be desirable to return just rows meeting a particular requirement. The optional **WHERE** clause can be used to define these criteria. Records that satisfy a specified condition(s) can be retrieved from a database table using the **WHERE** clause.



```
--AND
SELECT *
FROM Production
WHERE P_ID = 2 AND P_Name = 'DarkEgo';

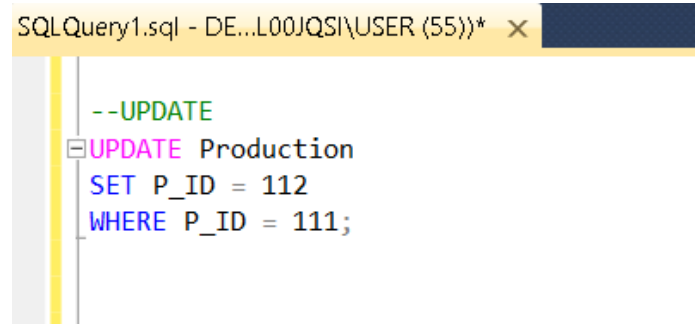
--OR
SELECT *
FROM Production
WHERE P_ID = 2 OR P_Name = 'DarkEgo';

--NOT
--OR
SELECT *
FROM Production
WHERE NOT P_ID = 2 ;
```

Figure 28 And Or & Not

1. It's likely that you'll wish to filter the table's contents using many conditions. This is possible with the **AND** operator. When displaying the output, the given conditions must be true.
2. To filter the tables, you may wish to use one or more conditions. This is possible with the **OR** operator. When the output is displayed, one of the conditions should be true.
3. You may wish to show the values that, under the specified conditions, are false. Most likely, **NOT** operator will work for you.

2. Update




```
SQLQuery1.sql - DE...L00JQSI\USER (55))* X
--UPDATE
UPDATE Production
SET P_ID = 112
WHERE P_ID = 111;
```

Figure 29 Update Statement

The current records in a table can be changed using the SQL **UPDATE** Statement. Given that it simply makes changes to the data in a table without altering the table's structure, this statement is a component of the Data Manipulation Language (DML)

3. Between

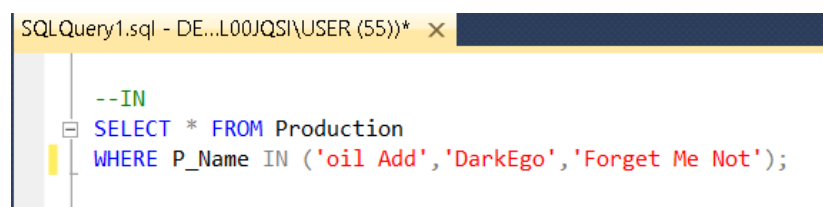


```
--BETWEEN
SELECT *
FROM Production
WHERE P_ID BETWEEN 2 AND 6;
```

Figure 30 Between Statement

To obtain the range of data you wish to filter, use the **BETWEEN** operator. Although using the equality and **AND** operators is an option, **BETWEEN** offers a more convenient syntax.

4. In

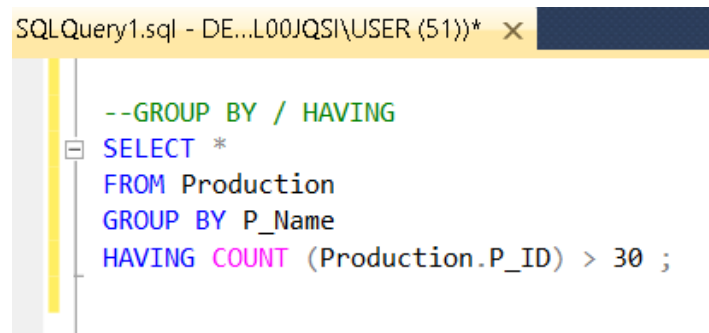


```
SQLQuery1.sql - DE...L00JQSI\USER (55))* X
--IN
SELECT * FROM Production
WHERE P_Name IN ('oil Add', 'DarkEgo', 'Forget Me Not');
```

Figure 31 IN statement

Additionally, you can specify multiple exact bases for filtering each row using the **IN** operator. For instance, only rows with values in the defined bracket can be retrieved:

5.Group by / Having



```
SQLQuery1.sql - DE...L00JQSI\USER (51))* X
--GROUP BY / HAVING
SELECT *
FROM Production
GROUP BY P_Name
HAVING COUNT (Production.P_ID) > 30 ;
```

Figure 32 Group by & Having Statement

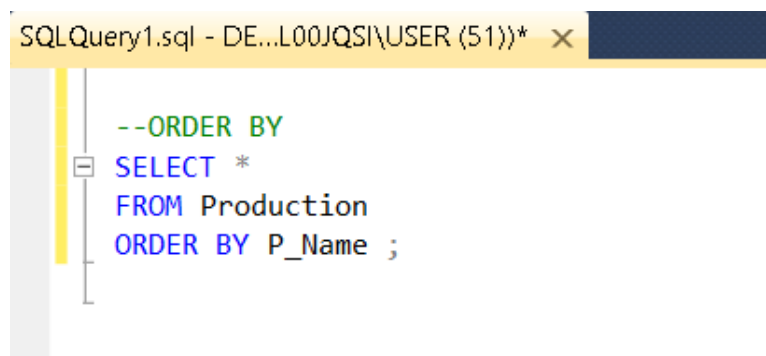
A popular SQL command for gathering data and drawing conclusions from it is **GROUP BY**.

When grouping data, there are three stages involved:

1. Split: the variables we have selected for the aggregate are used to divide the dataset into rows.
2. Submit an application: Calculate an aggregate function that yields a single number, such as the average, minimum, and maximum.
3. Combine: Every one of these outputs is combined to create a single, distinct table. We will then have a single value for the variable of interest in each mode.

Having filter rows in each of the groups that **GROUP BY** defines

6.Order By



```
SQLQuery1.sql - DE...L00JQSI\USER (51))* X
--ORDER BY
SELECT *
FROM Production
ORDER BY P_Name ;
```

Figure 33 Order By Statement

You've learned how to use SQL to retrieve data from your table thus far, but you'll notice that the data consistently appear in the same order. The `ORDER BY` clause allows you to change the order in which the data is retrieved.

Ascending (ASC) and descending (DESC) order are the two main choices.

Activity 03

3.1 Provide a suitable test plan to test the system against user and system requirements.

What is A Test Plan?

An official document called a test plan offers a comprehensive and well-structured explanation of the testing methods and techniques that will be applied to assess the software system that is being tested. This document provides in-depth information regarding the project's scope, goals, resources, schedule, approach, methodology, and risks.

The test planning process may differ depending on the strategy the team decides on. For example, with Agile development, the test plan can be built in smaller chunks during each iteration, in contrast to conventional testing methodologies where the test plan is established at the beginning of the project. This allows for better revision and adjustment of the test plan in the case that problems arise.

Levels of Testing



Figure 34 Level of Testing

1. Unit testing

Unit testing is a software testing procedure used to examine individual software parts, components, or units. The most fundamental kind of testing is this one, which aims to confirm that every code unit functions as intended and is free of glitches, errors, and faults. Software developers often complete this level during the development (or coding) process, isolating a part of code and confirming its accuracy. It's known as "Whitebox" testing, which allows code to be read but typically not changed. It should always be done early in the development process to ultimately save costs, time, and effort.

The earlier the better when it comes to unit testing! Developers frequently finish this testing before sending the code for additional testing.

2. Integration testing

The next step is integration testing, which involves combining and testing various software modules and components to ensure that everything is prepared for the next phase. The objective is to test to uncover any flaws or problems between the various modules, since a typical software project will probably consist of multiple modules, each created by a different programmer. This stage is also known as string testing, thread testing, or I & T (integration and testing).

Testers employ this strategy, which focuses on verifying the data flow from one module to the next. It confirms that everything functions as a single cohesive unit, making it another essential testing step.

3. System testing

System testing is the next step up from software testing. This verifies whether a system complies with the relevant specifications. Performance, load, reliability, security, and other components are inspected during system testing in order to assess the end-to-end system specifications.

This process is usually carried out on the finished software product by a qualified testing agent prior to its release onto the market with actual users. This is a crucial step since the project is almost finished, and it should be tested in a setting that will be identical to what the user would see when it is done.

4. Acceptance testing

Lastly, acceptance testing comes into play. Since the objective of this level is to determine whether the requirements have been satisfied upon delivery of the final product, it differs from the others in that it is carried out by the user or customer.

Acceptance testing, being the last stage, must validate the entire business process and verify that there are no problems in spelling, grammar, or appearance
(Calvello, 2022).

Testing Methods

Software testing can be conducted in a variety of ways. The available methods are briefly described in this chapter.

Black-Box Testing

Black-box testing is the process of testing an application without having any knowledge of its internal workings. The tester does not have access to the source code and is unaware of the system architecture. In a black-box test, the tester often interacts with the system's user interface by supplying inputs and analysing results while being unaware of the location and method of processing the inputs.

White-Box Testing

The in-depth analysis of the core logic and code structure is known as "white-box" testing. Glass testing and open-box testing are other terms for white-box testing. A tester must be familiar with the internal workings of the code in order to conduct white-box testing on an application.

To identify the unit or chunk of code that is acting strangely, the tester must examine the source code.

(Tutorials Point, 2023).

Test Plan for Quiet Attic Films Management System!

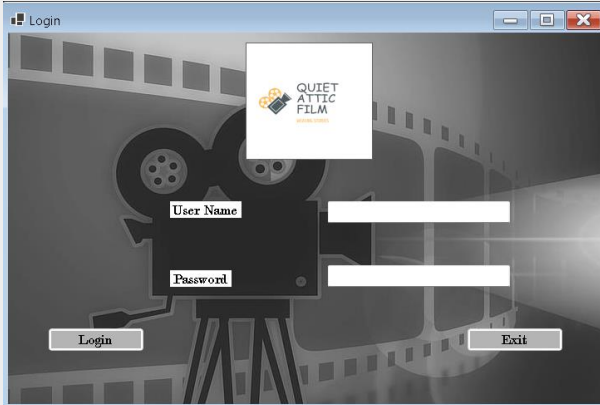
Table 2 Test Plan

Functions	<ol style="list-style-type: none"> 1. Login Form 2. Main Menu 3. Production Details 4. Properties Details 5. Client Details 6. Location Details 7. Staff Details
Functions which going to Test	<ol style="list-style-type: none"> 1. Login Form 2. Main Menu 3. Production Details 4. Properties Details 5. Client Details 6. Location Details 7. Staff Details
Test Level	Unit Testing
Test Method	White Box Testing

3.2 provide relevant test cases for the database you have implemented.

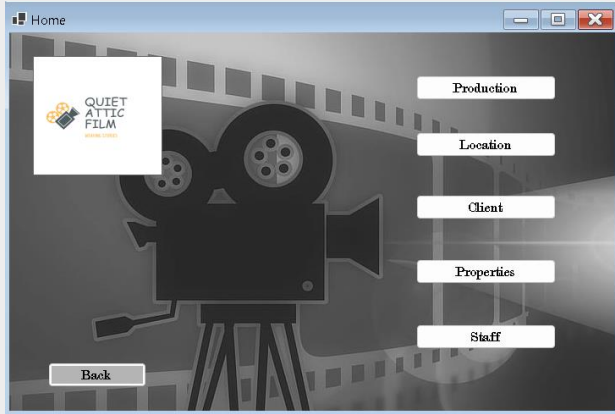
Test Case for Login

Table 3 Test Case for Login

Test ID	0001
Test Name	Login Form 
Steps to Test	Insert Username Insert Password Click on Login Button
Pre def Function	Register
Post def Function	Search Item
Date of Testing	11/15/2023
Expected Outcome	Successfully Login !
Actual Outcome	Login Success !
Tester Name	Thakshila Jayasinghe

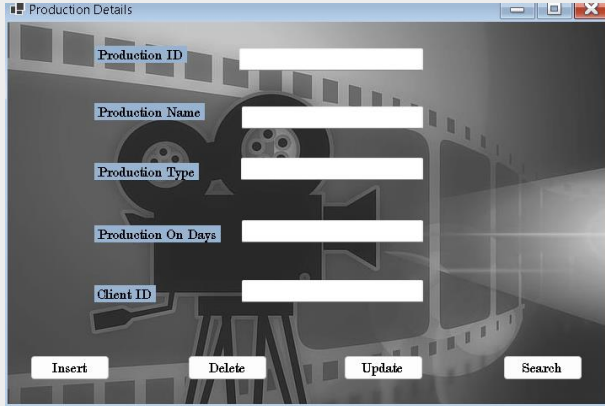
Test case for Main Menu

Table 4 Test case for Main Menu

Test ID	0002
Test Name	Main Menu 
Steps to Test	Click on Categories
Pre def Function	Login
Post def Function	Product Manage
Date of Testing	11/15/2023
Expected Outcome	Display Main menu
Actual Outcome	Successfully display main menu
Tester Name	Thakshila Jayasinghe

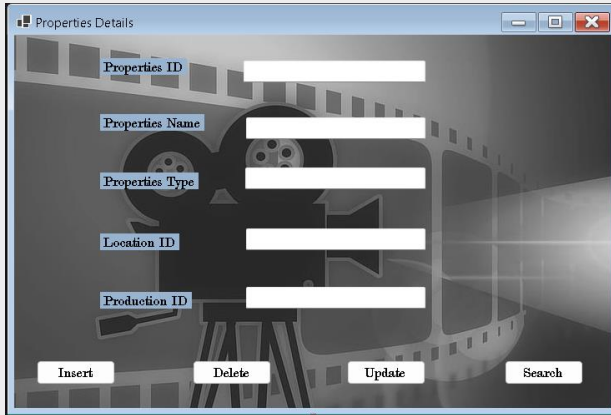
Test case for Production Details

Table 5 Test case for PD

Test ID	0003
Test Name	Production Details 
Steps to Test	Insert Details Delete Details Update Details Search Details
Pre def Function	Main Menu
Post def Function	Properties Manage
Date of Testing	11/15/2023
Expected Outcome	Insert, Update, Delete, Search Information
Actual Outcome	Successfully Insert, Update, Delete, Search Information
Tester Name	Thakshila Jayasinghe

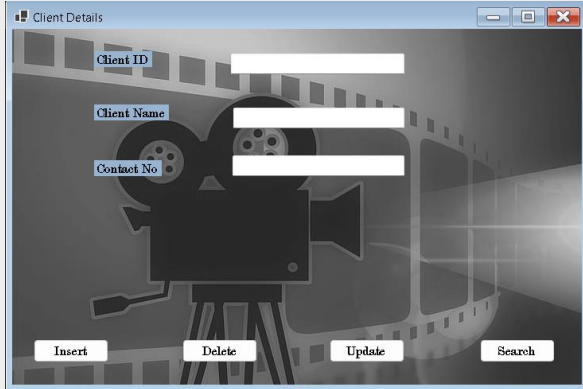
Test case for Properties Details

Table 6 Test case for PD

Test ID	0004
Test Name	Properties Details 
Steps to Test	Insert Details Delete Details Update Details Search Details
Pre def Function	Product Manage
Post def Function	Client Manage
Date of Testing	11/15/2023
Expected Outcome	Insert, Update, Delete, Search Information
Actual Outcome	Successfully Insert, Update, Delete, Search Information
Tester Name	Thakshila Jayasinghe

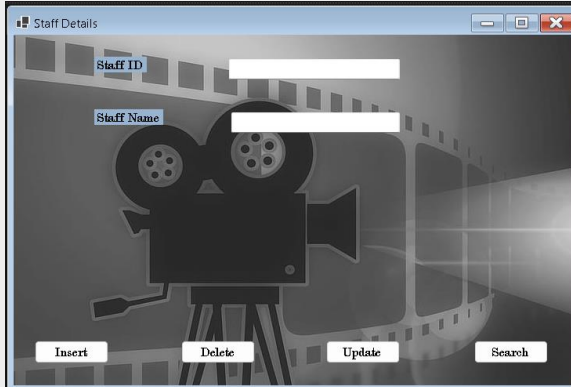
Test case for Client Details

Table 7 Test case for CD

Test ID	0005
Test Name	Client Details 
Steps to Test	Insert Details Delete Details Update Details Search Details
Pre def Function	Properties Manage
Post def Function	Staff Manage
Date of Testing	11/15/2023
Expected Outcome	Insert, Update, Delete, Search Information
Actual Outcome	Successfully Insert, Update, Delete, Search Information
Tester Name	Thakshila Jayasinghe

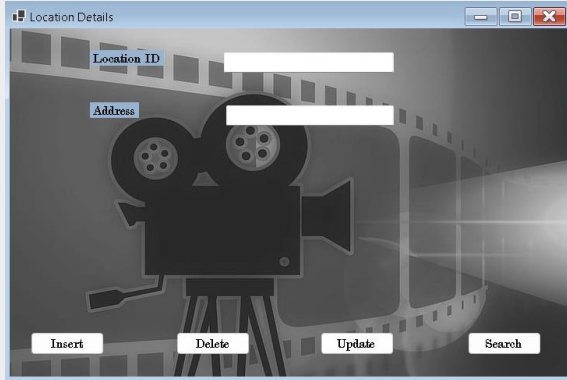
Test case for Staff Details

Table 8 Test case for SD

Test ID	0006
Test Name	Staff Details 
Steps to Test	Insert Details Delete Details Update Details Search Details
Pre def Function	Client Manage
Post def Function	Location Manage
Date of Testing	11/15/2023
Expected Outcome	Insert, Update, Delete, Search Information
Actual Outcome	Successfully Insert, Update, Delete, Search Information
Tester Name	Thakshila Jayasinghe

Test case for Location Details

Table 9 Test case for LD

Test ID	0007
Test Name	Location Details 
Steps to Test	Insert Details Delete Details Update Details Search Details
Pre def Function	Staff Manage
Post def Function	—
Date of Testing	11/15/2023
Expected Outcome	Insert, Update, Delete, Search Information
Actual Outcome	Successfully Insert, Update, Delete, Search Information
Tester Name	Thakshila Jayasinghe

3.3 Assess how the selected test data can be used to improve the effectiveness of testing.

Test cases can be found one by one by using the test ID and test name as testing data. The "Steps to test" section of each test case contains important test data since it describes how the system operates. Each test case's pre- and post-def function is a great aid in locating pertinent data. The most important information for system testing is the testing date. Despite being the primary result of testing, actual outcome data is still quite important. The real results speak to the aspects that need to be improved upon as well as the issues that exist. The most crucial thing to compare is the expected and actual results. Lastly, the tester's name was included. If any further issues are found, it is crucial to get in touch with the tester. These are the test data that were utilised for the testing. And it greatly aids in the efficacious and successful testing portion.

3.4 Get independent feedback on your database solution from the non-technical users and some developers (use surveys, questioners, interviews or any other feedback collecting method) and make a separate conclusion from the feedbacks.

What Does Feedback Mean?

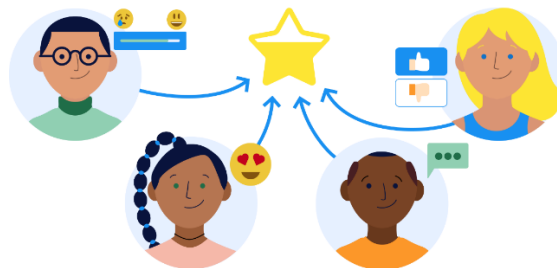



Figure 35 Feedback

Feedback is an event that occurs when the output of a system is used as input back into the system as part of a chain of cause and effect. This alters variables in the system, therefore resulting in different output and consequently different feedback as well, which can either be good or bad. In the case of a system which requires knowledge of the output in order to improve or deliver on a specific output, then feedback is essential and good. But for a system that does not require feedback, such as an audio system, then feedback is often bad. (Rouse, Technology Expert, 2016)

The google form created for get feedbacks.



Feedback Form

The Online Managment System for Quiet Attic Films in London !

QUIET ATTIC FILMS IN LONDON

Description (optional)

Your Name

Short-answer text

Your Gmail Address

Short-answer text

How would you rate the overall user interface ?

☐ Excellent

☐ Good

☐ Average

☐ Below Average

☐ Poor

How user experience of the film management system ?

☐ Excellent

☐ Good

☐ Average

☐ Below Average

☐ Poor

Suggest Improvement ?

Short-answer text

Figure 36 Feedback Form

Independent 07 feedbacks on database solution from the non-technical users and some developers.

7 responses
[Link to Sheets](#)

Accepting responses
☒

Summary
Question
Individual

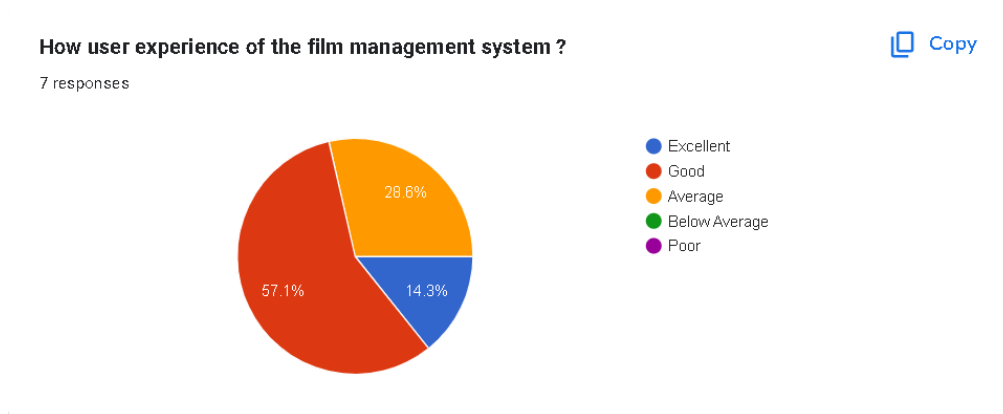
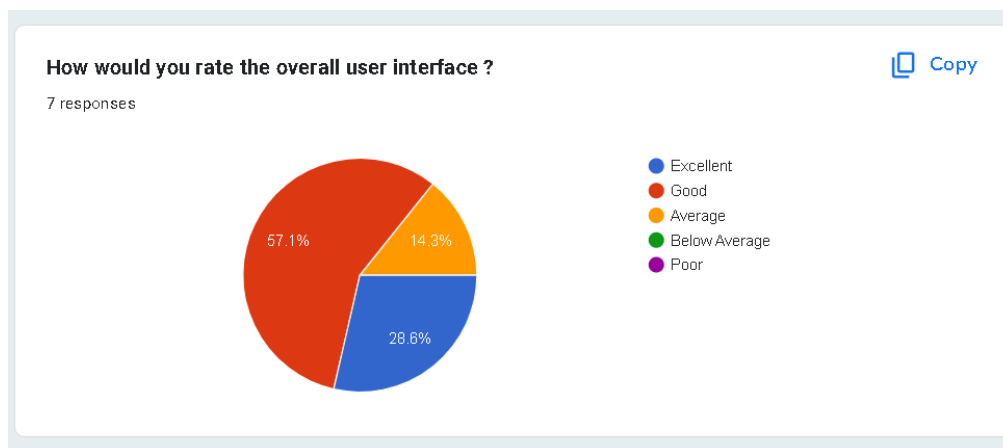
QUIET ATTIC FILMS IN LONDON

Your Name
7 responses

Shalani Perera
Savindi De Silva
Kavya Edirisinghe
Sachin Nawrathane
Manusha Nanayakkara
Kalana Feranando
Eshan Dananjaya

Your Gmail Address
7 responses

Sperera45@gmail.com
SavindiSilva5@Gmail.com
Kavyaedirisinghe69@gmail.com
SachinNawarathne45@gmail.com
ManushaNaya23@Gmail.com
Kalanafernando09@gmail.com
eshandananjaya99@gmail.com



Suggest Improvement ?

7 responses

System Good But sometime to its very slow .User want stay many time to use functions .
Syestem is so good & User freindly !
Some Button re not working , Update & Insert as well .Recommend to develop these .
Have some security issues. I used to wrong password to login but its working .Improve your security system well.
System is good . Recommended to added new feactures.
Intuitive Interface: Ensure the system has a user-friendly interface with clear navigation and easy-to-understand functionalities. Simplify complex tasks with clear instructions or tooltips. Search and Filtering: Improve the search and filtering options to make it easier for users to find movies based on different criteria such as genre, release date, actors, etc. Personalization: Implement a feature that allows users to create personalized watchlists, rate movies, or receive recommendations based on their preferences. Responsive Design: Make sure the system is responsive across various devices (desktop, mobile, tablet) for seamless access and usage

Figure 37 Feedback Responses

Conclusion from the feedbacks.

Received seven feedback from three non-technical users and four technical users. Receive positive feedback from is a pleasure. But everyone has responded to me with suggestions for system enhancements as well. Once you've grasped some system errors, go through them all. & what further features would you like to include. Suggested issues include the slow speed, poorly functioning buttons, and security flaws. Therefore, they made the decision to test each button once again. Try reducing the slow issue. & introduce fresh system security features. also add more features to the system as I have discovered that this is a recommendation from users.so glad the database system for the film production company because I've received positive comments from both technical and non-technical users. In the end, I believe the purpose is accomplished and successful.

Activity 04

4.1 Produce technical and user documentation for a fully functional system.

What is Software Documentation?

In software engineering, "documentation" refers to a broad category of written materials and documents that deal with the creation and application of software products. Whether produced by a small team or a large enterprise, all software development products need to have some accompanying documentation. Additionally, throughout the software development lifecycle (SDLC), other document kinds are produced. Documentation serves as a means of elucidating product functionality, consolidating project-related data, and facilitating discussions between developers and stakeholders regarding all pertinent questions.

Types of documentation

Ensuring that developers and stakeholders are working towards the same goals in order to complete the project is the primary aim of good documentation. Numerous sorts of documentation are available to accomplish these.

According to the following categories.

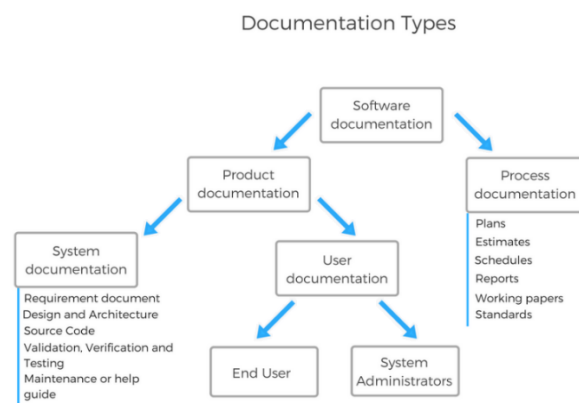


Figure 38 Documentation Types

All software documentation can be divided into two main categories:

1. Product documentation
2. Process documentation

1. Product documentation

Explains the product in development and offers guidance on how to use it for different purposes.

There are two categories of product documentation: system documentation and user documentation.

System documentation

Symbolizes documents outlining the components and the system as a whole. It consists of help manuals, design choices, architectural summaries, programme source code, and requirements documents.

User documentation

Includes instructions written mostly for system administrators and product end users. Tutorials, user manuals, installation, troubleshooting, and reference manuals are all examples of user documentation.

2. Process documentation

Consists of all documentation created throughout creation and upkeep that, well, explain the process. Project plans, test schedules, reports, standards, meeting minutes, and even corporate letters are typical types of process documentation

(Prototypr, 2018).

Technical Documentation

Any written document that explains the use, intent, development, or architecture of a good or service is considered technical documentation. Its objective is to elucidate a service that a company provides.

Technical documents come in a variety of forms, each aimed at a certain readership.

Technical writers, project managers, developers, and product or service professionals are typically in charge of writing technical papers.

Component of Technical Documentation

- 1. System Introduction**
- 2. User & System Requirement**
- 3. Diagrams**
- 4. Gantt chart**
- 5. Summary**

1. System Introduction

System of Management for Quiet Attic Films was created. Incorporated into seven interfaces, including the Main Menu, Login, Production, Properties, Client, Location, and Staff sections. In order to access the main menu and choose the appropriate category, one must first log in to the system with their login and password. To handle data, each category has an Insert, Update, Delete, and Search option.

2. User & System Requirement

User Requirement

1. User Authentication
2. Client Management
3. Location Management
4. Staff type
5. Production Management

System Requirement

1. User Authentication

(Functional Requirements)

- iv. User Registration: Users It should be possible for users to register for an account using a special username and password.
- v. User Login: Sign in It must be possible for users to safely log in with their registered login credentials. To preserve user authentication during a session, employ session management.
- vi. Password Recovery: Make sure users have a way to retrieve their passwords, including email verification or security questions.

(Non-Functional Requirements)

- iv. Security: Encrypt passwords and private user information. Use secure protocols (HTTPS) for communication
- v. Performance: Make sure that the login and authentication processes respond quickly.
- vi. Scalability: Build the system to accommodate an increasing user base.

2. Client Managements

(Functional Requirements)

- iv. Registering as a Client: Permit the entry of new customers with pertinent information.
- v. Updates to Client Information: Give customers the option to amend their information.
- vi. Client Lookup: Incorporate a search feature to effectively locate and obtain customer data.

(Non-Functional Requirements)

- iv. Data Accuracy: Make ensuring that the customer data is consistent and accurate.

- v. Control of Access: Establish user roles and permissions to manage who can access client data.

3. Location Management

(Functional Requirements)

- iv. Registration of Location: Permit new places to be added with the required information.
- v. Change of Location: Give users the option to update their location data.
- vi. Location Lookup: Put in place a location search feature.

(Non-Functional Requirements)

- iii. Data Correctness: Make that the location data is accurate and consistent.
- iv. Scalability: Create the system to accommodate an increasing number of sites.

4. Staff Type

(Functional Requirements)

- iii. Staff Type Definition: Permit the definition of several personnel categories together with the corresponding roles and authorizations.
- iv. Staff Type Assignment: Assignment Permit users to have different staff types assigned to them

(Non-Functional Requirements)

- v. Access Control: Based on staff kinds, specify access permissions.

5. Production Management

(Functional Requirements)

- vi. Definition of a Product: Permit the defining of various goods and services.
- vii. Production Planning: Put in place production schedule planning functionality.
- viii. Inventory Management: Incorporate inventory management and tracking features.

(Non-Functional Requirements)

- iii. Performance: Make sure that massive amounts of production data are handled effectively.
- iv. Scalability: Build the system to accommodate an increasing quantity of items and production information.

3.Diagrams

Use Case Diagram

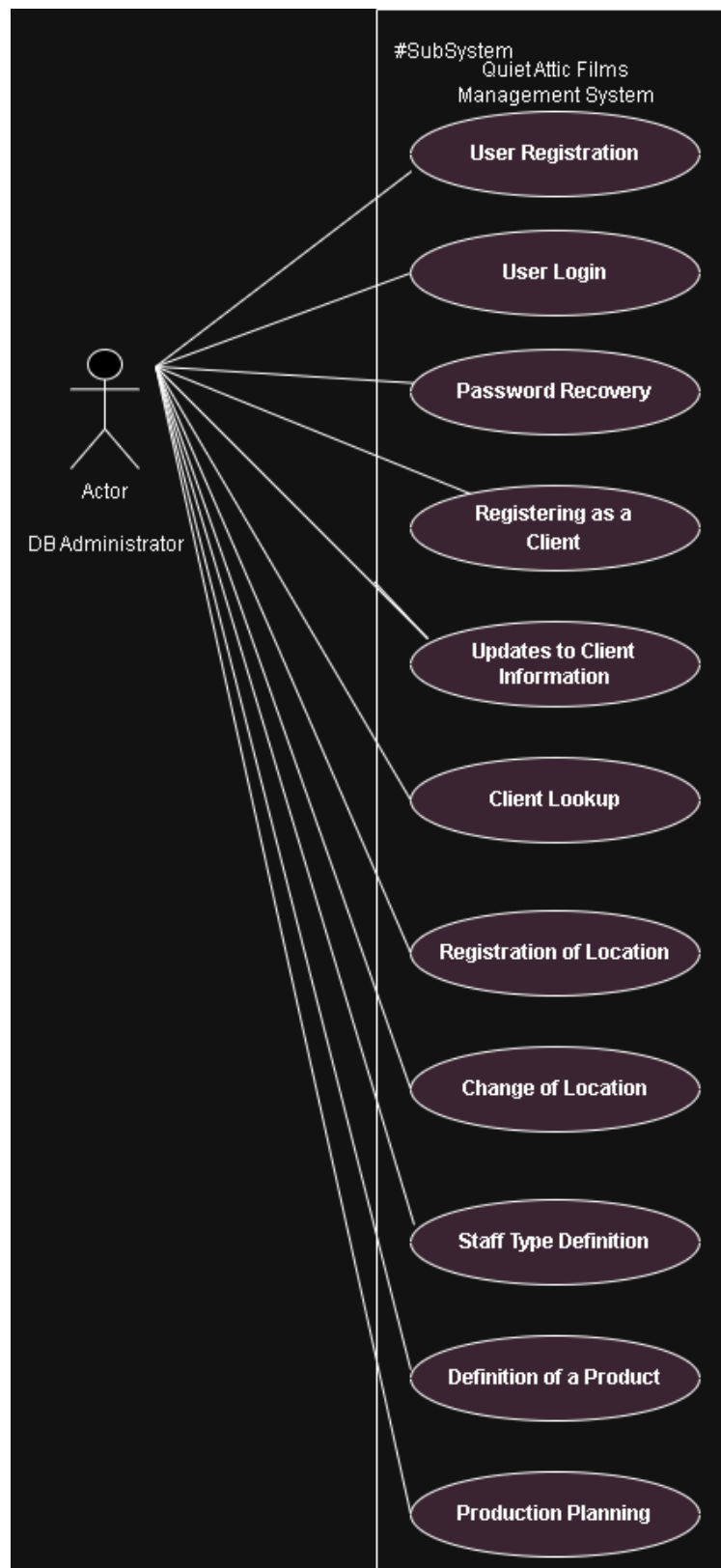


Figure 39 Use case Diagram

Class Diagram

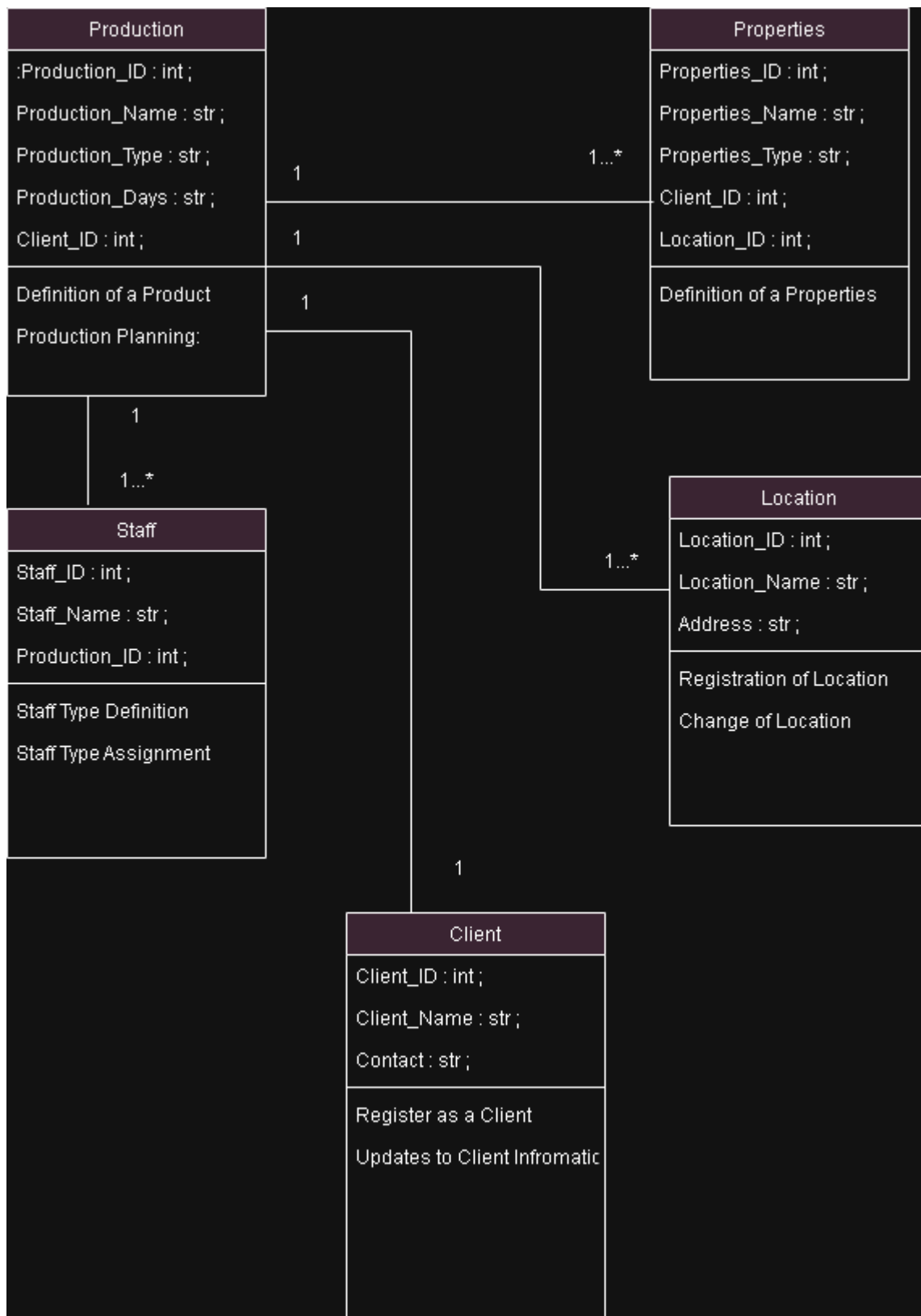


Figure 40 Class Diagram

Sequence Diagram

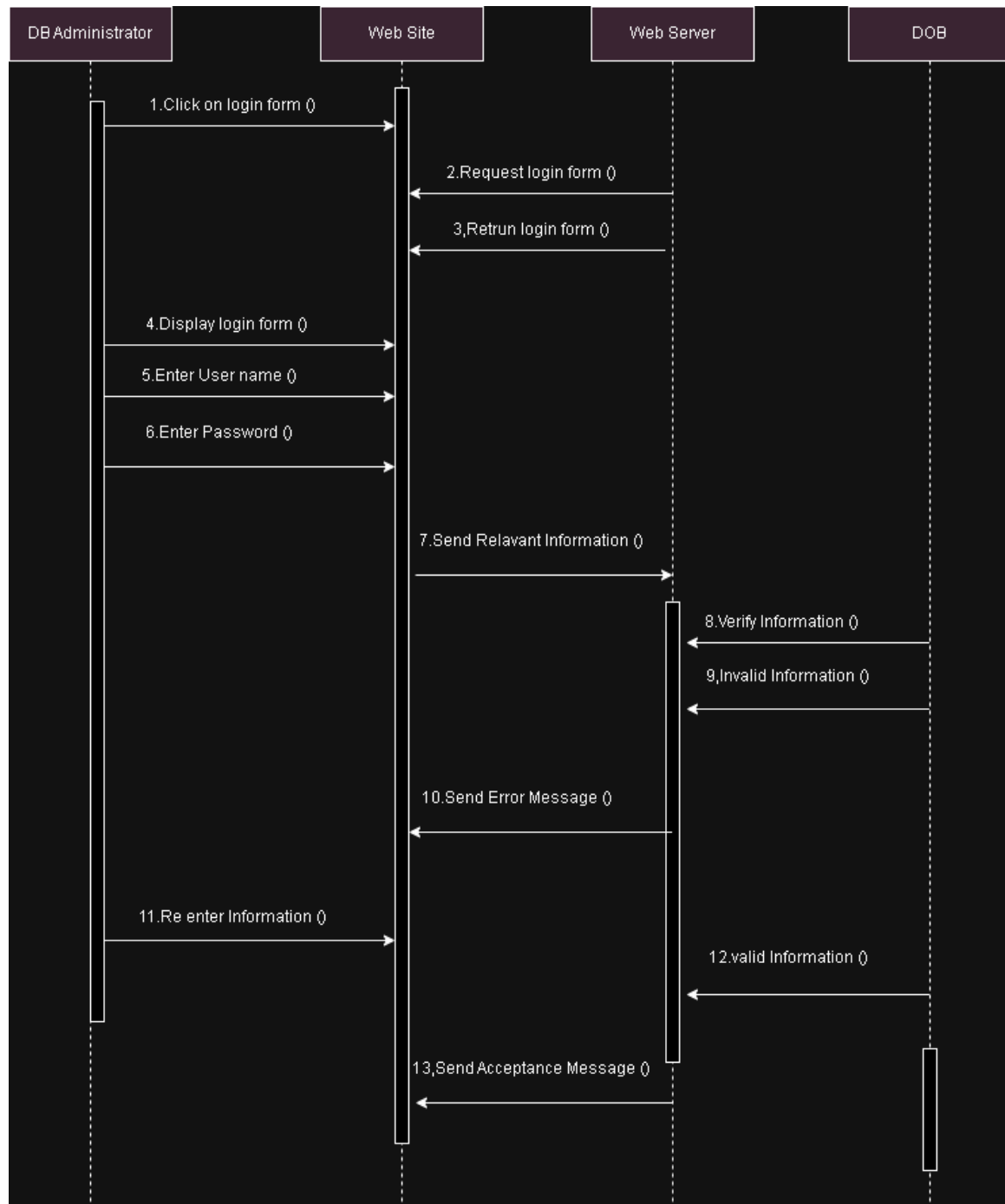


Figure 41 Sequence Diagram

4.Gantt Chart

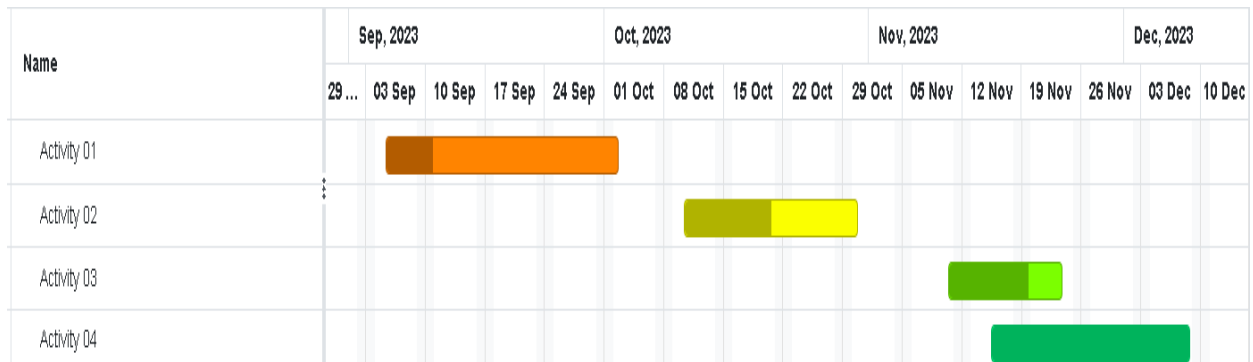


Figure 42 Gantt Chart

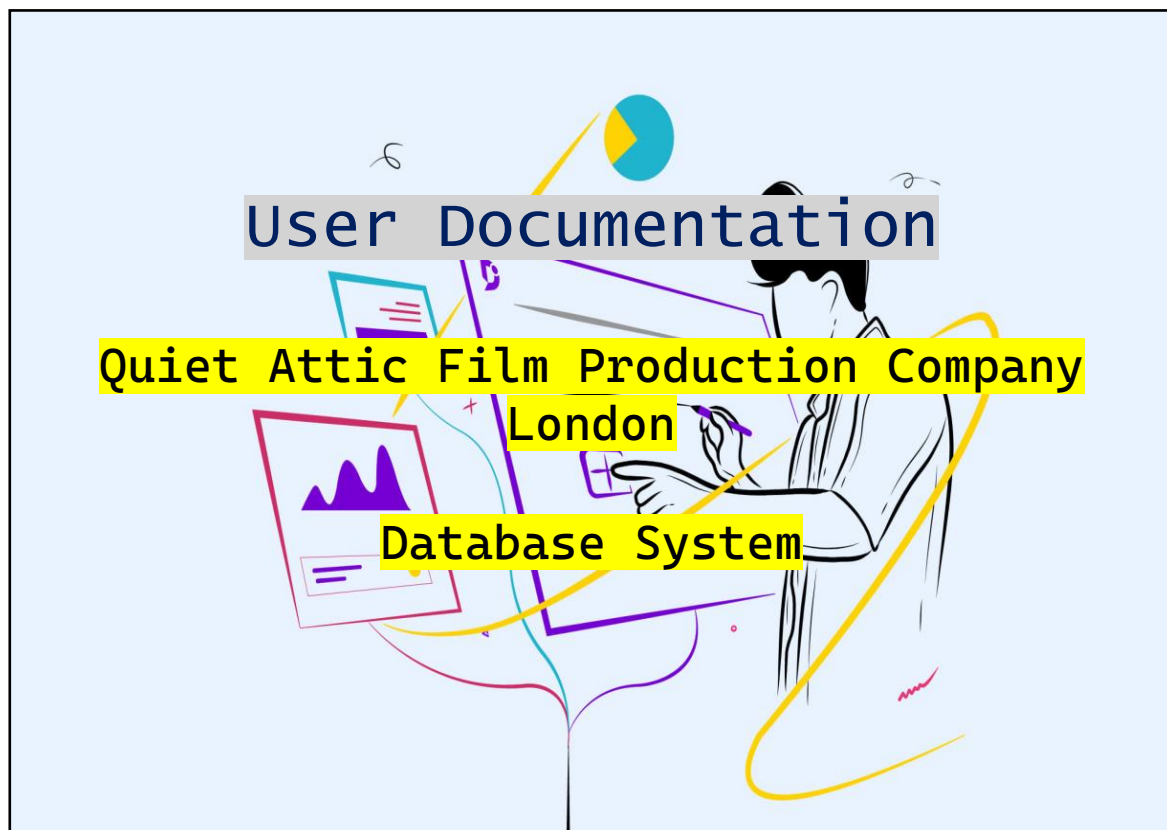
5.Summary

Created a database system for London's Quiet Attic Film Production Company in two months. Production, Client, Properties, Location, and Staff were identified as the entities. An ER diagram was first produced for the system and converted to a logical schematic. A wireframe for the system was then created, and a database system was developed using SQL for the film production company. After that, used Visual Studio to design interfaces for the system. verified their security mechanism, tested every scenario, and received input from both technical and non-technical people.

User documentation

Information produced to aid users in comprehending and utilizing a product is known as user documentation. Although this documentation can be in many different formats, it usually comes as written material with a product.

Various terms have been used to refer to modern user documentation, such as product information, user guides, manuals, and how-to instructions. To guarantee that a product is utilized accurately and consistently, user documentation is produced. Another name for it is end-user documentation.



How to Login Database System!

- 1. Enter your User Name**
- 2. Enter your Password**
- 3. Click Login**

If you don't get register yet, register via film production company & get your User Name & Password!

Only Quiet Attic Film Production Company DB administrator Can Change the Password.

Any Help Contact 071 2345698 Developer

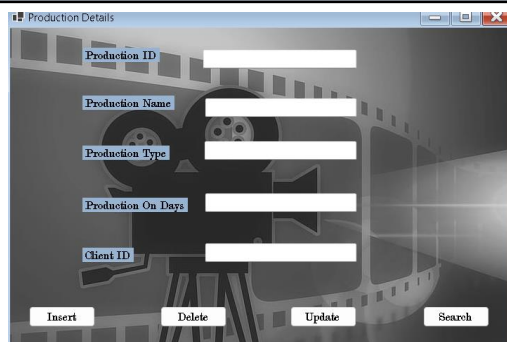


After Login Display Main Menu As this.

Choose the category what you want to login & click on it.

Then you can access relevant category details form.

If want get back in to Login Form , Click Back Button.



To add new records, fill the blanks & click Insert Button.

To delete any records, select the record want to delete & click Delete Button.

To update current records, fill the blanks & Click Update Button.

If want any search click Search Button.

4.2 Including data flow diagrams showing movement of data through the system, and flowcharts describing how the system works.

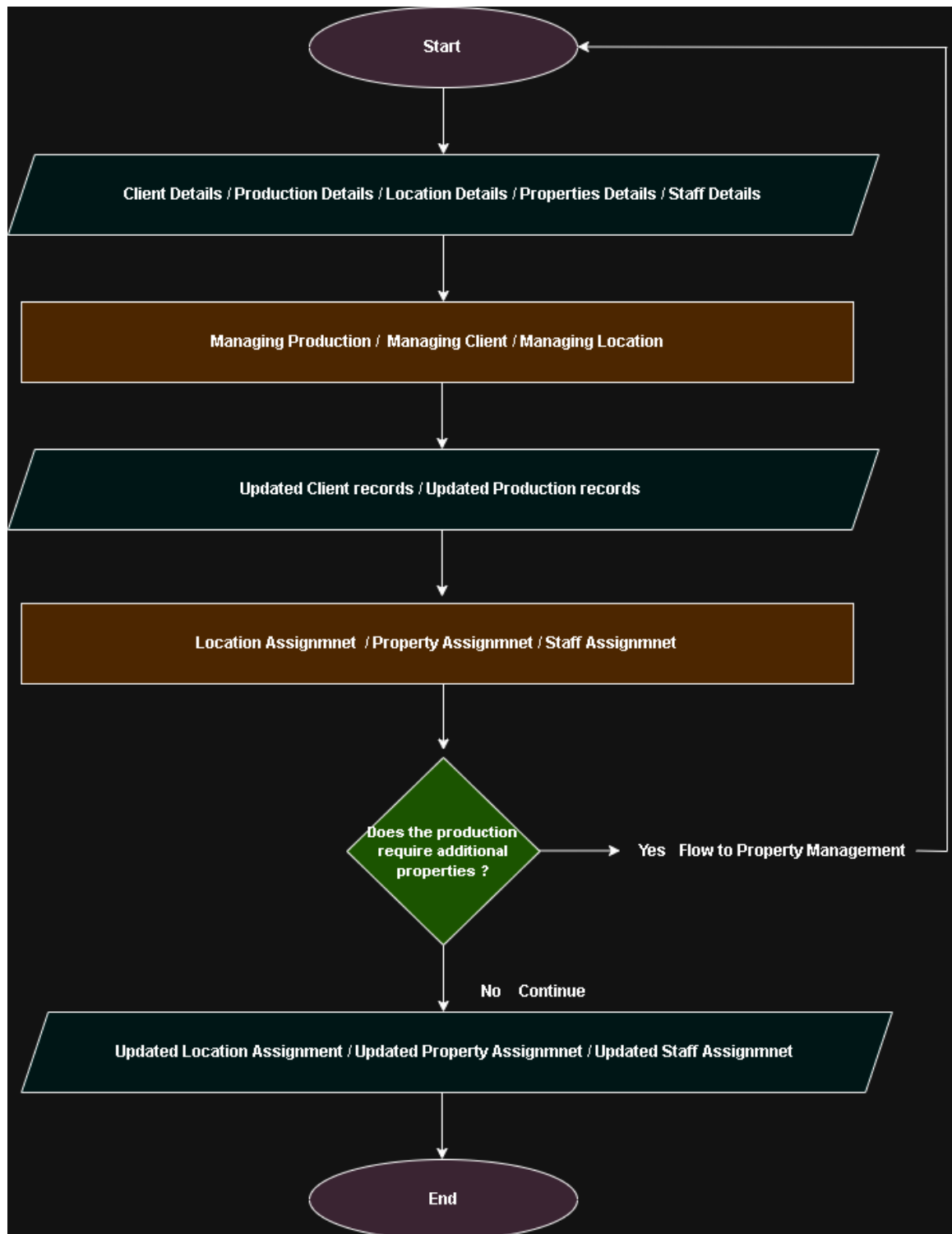


Figure 43 Flow Chart

This is a written explanation of the database system flowchart for Quiet Attic Films:

start:

The start terminal symbol, which denotes the start of the process, appears at the top of the flowchart.

Client Management:

data like client names, contact data, and project preferences are all part of the managed client information.

Production Management:

Clients are assigned to certain projects as part of the management of productions. Project names and timelines are among the production details that are documented.

Location Management:

Film production locations are overseen. This involves documenting specifics regarding the numerous locations where the productions are scheduled to occur.

Production Assignment:

Productions are reassigned to designated sites, and the kinds of personnel and properties needed are identified. A decision point determines if more attributes are required.

Property Assignment:

In the event that further properties are needed, the management of such properties is started. Information is kept on the extra properties.

Staff Assignment:

The production is given different categories of staff. This entails assigning responsibilities to the different characters in the movie.

Location Assignment:

The production is assigned locations. In this step, the locations for the film's various takes are finalized.

4.3 Evaluate the developed database by suggesting future enhancements to ensure the effectiveness of the system.

To ensure the effectiveness of the system for Quiet Attic Films, consider implementing the following future enhancements:

1. **User Authentication and Authorization:**

Establish a safe user authentication mechanism to manage database access. To protect the privacy and integrity of your data, assign varying degrees of authorization to different positions inside the company.

2. **Mobile Compatibility:**

Provide a mobile-friendly user interface or a specific mobile application so that users may access and update information while they are on the go. Employees that might need to oversee productions or verify details while on location will find this to be extremely helpful.

3. **Integration with Project Management Tools:**

Connect the system to well-known project management software to improve teamwork and expedite processes. This can involve communication, scheduling, and task management systems.

4. **Automated Notifications:**

Set up automated notification systems to inform users of impending deadlines, significant updates, or production modifications. This guarantees prompt action and aids in informing the team.

5. **Data Analytics and Reporting:**

Incorporate data analytics tools to offer insights into customer preferences, resource usage, and production patterns. Planning resources and making strategic decisions can both benefit from this.

Quiet Attic Films may make sure that their database system is reliable, easy to use, and in line with the changing requirements of the company by implementing these upcoming improvements.

References

- Staff Contributor. (2023, July 28). Retrieved from <https://www.dnsstuff.com/how-to-increase-database-performance>
- Amazon. (2023). *Amazon*. Retrieved from <https://aws.amazon.com/what-is/sql/>
- Anon. (2019). *TutorialsPoint*. Retrieved from https://www.tutorialspoint.com/dbms/dbms_data_models.htm
- Ben Lutkevich. (2023, February). *Tech Target*. Retrieved from Tech Target Network: <https://www.techtarget.com/searchdatamanagement/definition/database>
- Calvello, M. (2022, October 08). Retrieved from <https://fellow.app/blog/engineering/the-levels-of-testing-in-software-engineering-explained/>
- Collegenote. (2023). *Collegenote*. Retrieved from Collegenote: <https://www.collegenote.net/curriculum/software-engineering-csit/54/310/>
- Hughes, A. (2023). *TechTarget*. Retrieved from <https://www.techtarget.com/searchdatamanagement/definition/SQL-Server>
- Imperva. (2023). Retrieved from <https://www.imperva.com/learn/data-security/database-security/>
- Olawanle, J. (2022, October 24). Retrieved from <https://www.freecodecamp.org/news/sql-select-statement-and-query-examples/>
- Prototypr. (2018, January 18). Retrieved from <https://blog.prototypr.io/software-documentation-types-and-best-practices-1726ca595c7f>
- Rouse, M. (2016, January 07). *Technology Expert*. Retrieved from <https://www.techopedia.com/definition/7159/feedback>
- Rouse, M. (2019, September). *TechTarget*.
- Tutorials Point. (2023). Retrieved from https://www.tutorialspoint.com/software_testing/software_testing_methods.htm
- Unknown. (2023). *Erwin*.

Grading Criteria	Achieved	Feedback
LO1 Use an appropriate design tool to design a relational database system for a substantial problem.		
P1 Design a relational database system using appropriate design tools and techniques, containing at least four interrelated tables, with clear statements of user and system requirements.		
M1 Produce a comprehensive design for a fully-functional system, which includes interface and output designs, data validations and data normalization.		
D1 Evaluate the effectiveness of the design in relation to user and system requirements.		

LO2 Develop a fully-functional relational database system, based on an existing system design		
P2 Develop the database system with evidence of user interface, output and data validations, and querying across multiple tables.		
P3 Implement a query language into the relational database system		
. M2 Implement a fully functional database system, which includes system security and database maintenance.		
M3 Assess whether meaningful data has been extracted through the use of query tools to produce appropriate management information		
LO3 Test the systems against user and system requirements		
P4 Test the system against user and system requirements.		

M4 Assess the effectiveness of the testing, including an explanation of the choice of test data used.		
D2 Evaluate the effectiveness of the database solution in relation to user and system requirements and suggest improvements.		
LO4 Produce technical and user documentation		
P5 Produce technical and user documentation.		
M5 Produce technical and user documentation for a fully-functional system, including data flow diagrams and flowcharts, describing how the system works.		
D3 Evaluate the database in terms of improvements needed to ensure the continued effectiveness of the system.		