BUSINESS PROPOSAL FOR MANZANEQUE LIMITED

IT HELPDESK SYSTEM

Student name : Hasitha Ekanayaka
Student id : 520919
Unit number and title: unit 4–Database Design & Development
Academic year : 22/23

INTRODUCTION

This proposal describes the system and user requirements and how a relational database should be designed to meet the demands of the business. The suggested method will guarantee efficient problem solving, offer perceptive analytics, and improve helpdesk efficiency.

User requirements

Helpdesk Operators

• Record and Monitor Inquiries: Note the identity of the caller, the help desk agent, the duration of the call, the computer's serial number, and any pertinent software and operating system details.

• Identification of the Caller:

Check the caller's ID number, job title, and department details against the personnel register.

• Software and Equipment Validation:

Verify that software and equipment have valid licenses by cross-referencing them with the corresponding registrations.

• Issue Tracking:

Give every question a special problem number so that any follow-up calls about the same issue can be easily referred to.

• Remarks and Explanations:

Permit operators to take thorough notes and provide thorough explanations of the issues they receive.

• Classification of Problems:

Select an issue kind from a pre-established list to facilitate efficient issue classification.

• History of Problem Resolution:

To enable speedier resolutions, make previous issues of the same kind, with the same tools, or from the same caller accessible.

Experts

- Expert Assignment:

Assign experts according to the category of the detected problem to ensure that they have the know-how to address particular problems.

- Management of Workload:

To assign assignments to the least busy specialist, show how many problems each specialist is currently addressing.

- Recording Resolutions:

Once an issue has been resolved, note the date, time, and specifics of the resolution.

System requirements-user:

- User authentication: Provides help desk operators and specialists with a secure login.
- Call Recording:

Record the name, helpdesk operator, time, computer specifications, and problem description of the caller.

- Identification of the Caller:

Take the caller's ID, department, and job title out of the personnel register.

- Software and Equipment Validation:

Check that the software and hardware match the equipment registry and have valid licenses.

- Issue Number Distribution:

Give every issue that is logged a unique problem number.

- Taking Notes:

Permit note-taking and problem-describing notes from operators.

- Issue Type Distribution:

Give help desk staff the option to select a problem category from a predetermined list.

- Improvement of Issue Types:

Permit problem categories to be refined in light of new knowledge.

- History of Problem Resolution:

Access prior issues with the same kind of device, from the same caller, or of a similar nature.

- Allocating Specialists: Determine the skill and workload of each specialist and assign them accordingly.

  Issue Resolve Logging:

- Note the date, time, specifics of the problem's resolution, and the amount of time it took.

.

System requirements-database:

- Database: Create a relational database to hold user information, hardware specifications, program details, troubleshooting logs, and past resolutions.
- Validation of Data:

Put data validation guidelines into practice to guarantee consistency and correctness.

- Establishing norms:

Use normalization strategies to reduce dependencies and redundancies.

- Interface with the user:

Provide a simple and easy-to-use interface for professionals and help desk operators.

- Documentation:

Provide reporting features to assess the overall effectiveness of the equipment, the workload of the specialists, and the need for training.

System requirements-Hardware and Software:

The hardware, software, and network infrastructure specifications required to deploy the Manzaneque Limited IT Helpdesk System are outlined in the system requirements. The general system requirements are as follows:

Hardware Requirements:

• Server: A specialized server that has the memory, computing power, and storage space needed to run the application and database.

In order to handle anticipated increases in data and user load, the server should be scalable.

• Server for Databases:

sufficient database storage, taking into account the anticipated volume of data.

enough RAM to effectively manage database queries.

For best database performance, use dependable and quick storage.

• Machines for clients:

Workstations or laptops for professionals and help desk staff.

ought to satisfy the minimal hardware needs of the selected operating system and application.

Software Requirements:

- Operating System:

Server: Windows,Linux,Android

Entity-Relationship Diagram (ERD)

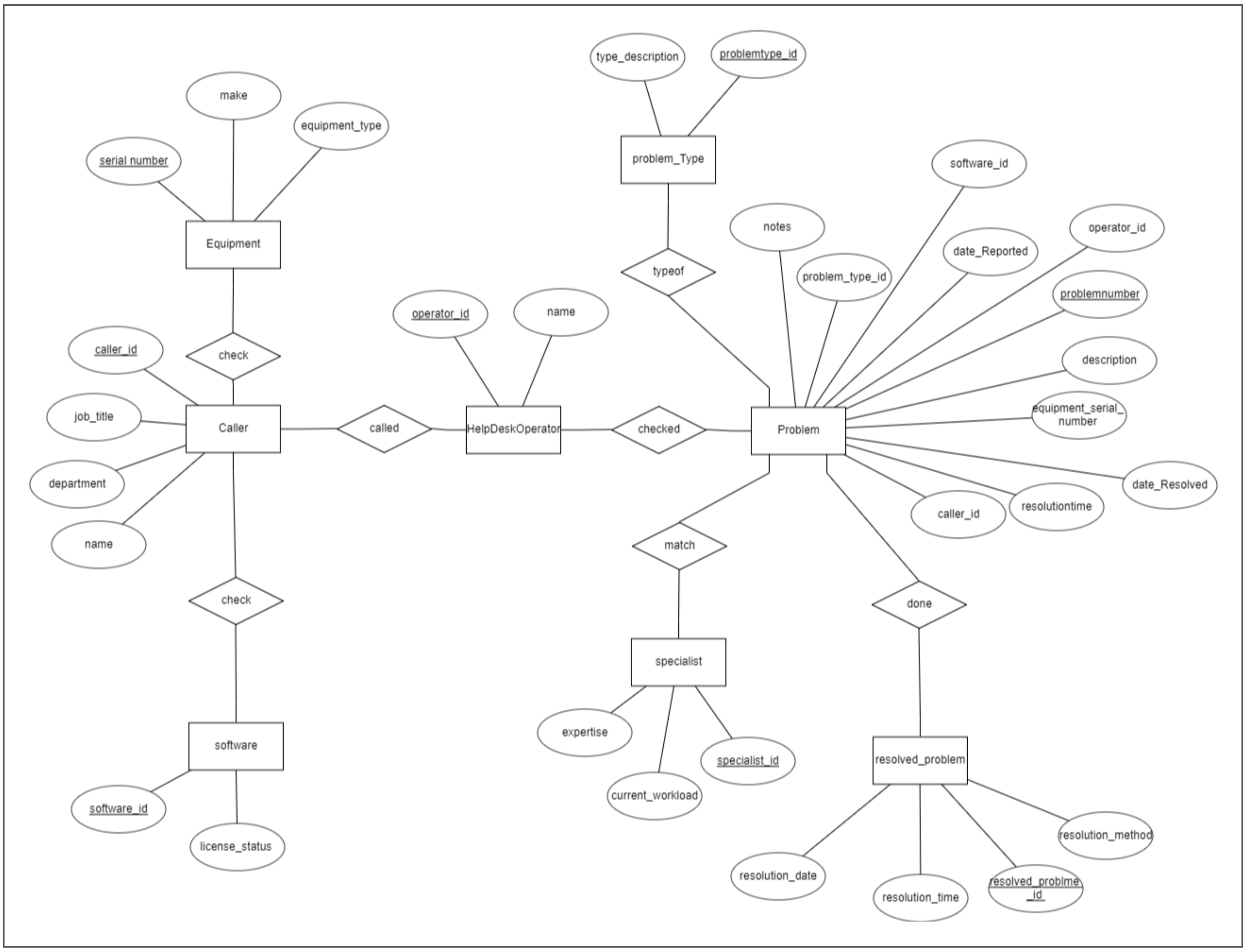

Figure 1:ER diagram

Entities:

- **Caller**

CallerID (Primary Key), Name, Department, Job Title, and Attributes

- **Helpdesk Operator**

Attributes: OperatorID (Primary Key), Name

- **Equipment**

Features: Manufacturer, Type of Equipment, Serial Number (Primary Key),

- **Software**

Attributes: SoftwareID (Primary Key), License Status

- **Problem**
  Features: Issue Number (Main Key), Synopsis, Comments, Date Submitted, Date Solved, Resolution Duration, The foreign keys that are called are CallerID, EquipmentSerialNumber, SoftwareID, OperatorID, and ProblemTypeID**.**
- **ResolvedProblem**

Attributes: ResolvedProblemID (Primary Key), Resolution Date, Resolution Time, Resolution Method

- **Problem Type**

Attributes: ProblemTypeID (Primary Key), Type Description

- **Specialist**

Attributes: SpecialistID (Primary Key), Expertise, Current Workload

Relationships:

- **Caller - Helpdesk Operator (Many-to-One)**

A caller logs multiple problems, but each problem is logged by one helpdesk operator.

Foreign Key: CallerID in Problem referencing Caller.

- **Caller - Equipment (One-to-Many)**

A caller can have multiple pieces of equipment.

Foreign Key: CallerID in Equipment referencing Caller.

- **Caller - Software (One-to-Many)**

A caller can use multiple software applications.

Foreign Key: CallerID in Software referencing Caller.

- **Caller - Problem (One-to-Many)**

A caller can report multiple problems.

Foreign Key: CallerID in Problem referencing Caller.

- **Equipment - Problem (One-to-Many)**

An equipment can have multiple reported problems.

Foreign Key: Serial Number in Problem referencing Equipment.

- **Software - Problem (One-to-Many)**

A software application can be associated with multiple reported problems.

Foreign Key: SoftwareID in Problem referencing Software.

- **Helpdesk Operator - Problem (One-to-Many)**

A helpdesk operator can log multiple problems.

Foreign Key: OperatorID in Problem referencing Helpdesk Operator.

- **Problem Type - Problem (One-to-Many)**

A problem type can be associated with multiple reported problems.

Foreign Key: ProblemTypeID in Problem referencing Problem Type.

- **Specialist - Problem (Many-to-Many)**

A specialist can work on multiple problems, and a problem can be worked on by multiple specialists.

Associative Table: Specialist_Problem with SpecialistID and ProblemNumber as Foreign Keys.

- **ResolvedProblem - Problem (One-to-One)**

A resolved problem is associated with exactly one problem, and a problem is associated with exactly one resolved problem.

Foreign Key: ProblemNumber in ResolvedProblem referencing Problem.

Normalization

The practice of structuring data in a database to lessen dependencies and redundancies is called normalization. Minimizing data abnormalities and achieving data integrity are the objectives. Large tables must be broken down into smaller, related tables and their relationships must be established as part of the normalization process. For the entities in the above scenario, the following is a basic normalization:

First Normal Form (1NF):

- Caller (1NF):

CallerID (PK), Name, Job Title, Department

- Helpdesk Operator (1NF):

OperatorID (PK), Name

- Equipment (1NF):

Serial Number (PK), Equipment Type, Make

- Software (1NF):

SoftwareID (PK), License Status

- Problem (1NF):

ProblemNumber (PK), Description, Notes, Date Reported, Date Resolved, Resolution Time, CallerID (FK), EquipmentSerialNumber (FK), SoftwareID (FK), OperatorID (FK), ProblemTypeID (FK)

- ResolvedProblem (1NF):

ResolvedProblemID (PK), Resolution Date, Resolution Time, Resolution Method, ProblemNumber (FK)

- Problem Type (1NF):

ProblemTypeID (PK), Type Description

- Specialist (1NF):

SpecialistID (PK), Expertise, Current Workload

Second Normal Form (2NF):

- Caller (2NF):

CallerID (PK), Name, Job Title, Department

- Helpdesk Operator (2NF):

OperatorID (PK), Name

- Equipment (2NF):

Serial Number (PK), Equipment Type, Make

- Software (2NF):

SoftwareID (PK), License Status

- Problem (2NF):

ProblemNumber (PK), Description, Notes, Date Reported, Date Resolved, Resolution Time

CallerID (FK), EquipmentSerialNumber (FK), SoftwareID (FK), OperatorID (FK), ProblemTypeID (FK)

- ResolvedProblem (2NF):

ResolvedProblemID (PK), Resolution Date, Resolution Time, Resolution Method

ProblemNumber (FK)

- Problem Type (2NF):

ProblemTypeID (PK), Type Description

- Specialist (2NF):

SpecialistID (PK), Expertise, Current Workload

Third Normal Form (3NF):

- Caller (3NF):

CallerID (PK), Name, Job Title, Department

- Helpdesk Operator (3NF):

OperatorID (PK), Name

- Equipment (3NF):

Serial Number (PK), Equipment Type, Make

- Software (3NF):

SoftwareID (PK), License Status

- Problem (3NF):

ProblemNumber (PK), Description, Notes, Date Reported, Date Resolved, Resolution Time

CallerID (FK), EquipmentSerialNumber (FK), SoftwareID (FK), OperatorID (FK), ProblemTypeID (FK)

- ResolvedProblem (3NF):

ResolvedProblemID (PK), Resolution Date, Resolution Time, Resolution Method

ProblemNumber (FK)

- Problem Type (3NF):

ProblemTypeID (PK), Type Description

- Specialist (3NF):

SpecialistID (PK), Expertise, Current Workload

This normalization assumes that each entity's attributes are functionally dependent on the primary key. Adjustments might be needed based on specific business rules and requirements.

Data validation

To guarantee the precision, consistency, and dependability of the data in a database, data validation is essential. The entities in the provided scenario can be subject to the following data validation rules:

Caller:

- Name:

Data Type: VARCHAR

Validation: Check for a reasonable length (e.g., not empty, not excessively long).

- Job Title:

Data Type: VARCHAR

Validation: Check for a reasonable length (e.g., not empty, not excessively long).

- Department:

Data Type: VARCHAR

Validation: Check for a reasonable length (e.g., not empty, not excessively long).

Helpdesk Operator:

- Name:

Data Type: VARCHAR

Validation: Check for a reasonable length (e.g., not empty, not excessively long).

Equipment:

- Serial Number:

Data Type: VARCHAR or INTEGER (depends on the format)

Validation: Check for uniqueness.

- Equipment Type:

Data Type: VARCHAR

Validation: Check for a reasonable length (e.g., not empty, not excessively long).

- Make:

Data Type: VARCHAR

Validation: Check for a reasonable length (e.g., not empty, not excessively long).

Software:

- SoftwareID:

Data Type: VARCHAR or INTEGER (depends on the format)

Validation: Check for uniqueness.

- License Status:

Data Type: VARCHAR

Validation: Check for a valid status (e.g., "Valid," "Expired").

Problem:

- Description:

Data Type: TEXT or VARCHAR

Validation: Check for a reasonable length (e.g., not excessively long).

- Notes:

Data Type: TEXT or VARCHAR

Validation: Check for a reasonable length (e.g., not excessively long).

- Date Reported:

Data Type: DATE or DATETIME

Validation: Ensure the date is in a valid format.

- Date Resolved:

Data Type: DATE or DATETIME

Validation: Ensure the date is in a valid format.

- Resolution Time:

Data Type: INTEGER or TIME

Validation: Ensure a positive value.

- ResolvedProblem:

Resolution Date:

Data Type: DATE or DATETIME

Validation: Ensure the date is in a valid format.

- Resolution Time:

Data Type: INTEGER or TIME

Validation: Ensure a positive value.

- Resolution Method:

Data Type: TEXT or VARCHAR

Validation: Check for a reasonable length (e.g., not excessively long).

Problem Type:

Type Description:

Data Type: VARCHAR

Validation: Check for a reasonable length (e.g., not empty, not excessively long).

Specialist:

- Expertise:

Data Type: VARCHAR

Validation: Check for a reasonable length (e.g., not empty, not excessively long).

- Current Workload:

Data Type: INTEGER

Validation: Ensure a non-negative value.

Output design

The way information is shown to users—through displays, reports, or other interfaces—is referred to as output design. In light of the Helpdesk system scenario for a real estate company, the following is a general summary of the possible outputs:

1. Call Logging Form:

form that help desk staff use to record new calls.

Caller name, operator name, call duration, serial number, operating system, software information, and problem description are among the fields.

2. Call Details Report:

Show precise call facts, such as caller ID, equipment and software details, problem type, and operator notes.

3. Problem List:

A list of all current problems, including problem number, caller name, problem type, and status (open or resolved).

4. Problem Type List:

Display a list of defined problem types for reference and selection.

5. Specialist Allocation Form:

Form for helpdesk operators to allocate a problem to a specialist.

Includes problem details, specialist selection, and workload information.

6. Specialist Workload Dashboard:

Display the workload of each specialist, showing the number of problems they are currently working on.

7. Resolved Problems Report:

List of problems that have been resolved, including problem number, resolution date, resolution time, and resolution method.

8. Caller Information Form:

Form to view and edit caller information.

Includes details such as name, job title, department, and a history of reported problems.

9. Equipment Information Form:

Form to view and edit equipment information.

Includes details such as serial number, equipment type, make, and a history of reported problems.

10. Software Information Form:

Form to view and edit software information.

Includes details such as software ID, license status, and a history of reported problems.

11. Problem Trends Report:

Analytical report showing trends in types of problems reported over time.

12. System Performance Dashboard:

Dashboard showing overall system performance, including the number of open problems, average resolution time, and operator efficiency.

13. Training Needs Analysis Report:

Report highlighting areas where employee training may be needed based on the types and frequency of reported problems.

14. Problem Resolution Time Analysis Report:

Report analyzing the time taken to resolve different types of problems.

15. User Feedback Form:

Form for users to provide feedback on problem resolution, helping to improve the helpdesk system.

Interface diagram

An interface diagram typically outlines how different components or modules in a system interact with each other. Since the given scenario involves a Helpdesk system for a real estate company, I'll provide a simplified representation of the main components and their interactions:

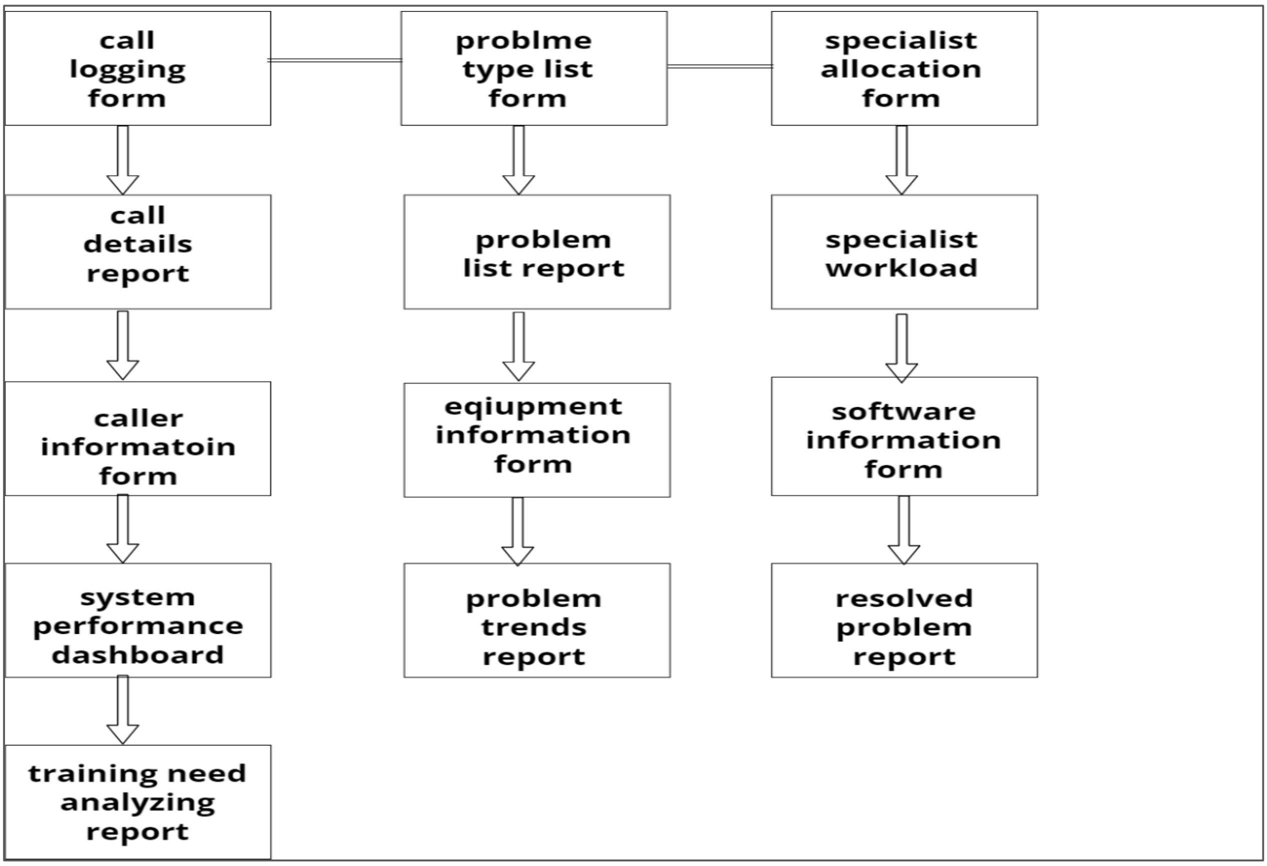

Figure 2:interface diagram

In this schematic:

New call information is entered using the Call Logging Form.

Predefined problem types can be selected using the Problem Type List Form.

A specialist is assigned an issue using the Specialist Allocation Form.

Numerous reports offer insights into many parts of the Helpdesk system, including the Call Details Report, Problem List Report, Specialist Workload, Resolved Problems Report, and others.

Details about callers, equipment, and software can be seen and edited using forms such as Caller Information, Equipment Information, and Software Information.

Dashboards similar to the System Performance Dashboards provide a summary of the general health of the system.

Reports using analytical data, such Training Needs Analysis and Problem Trends, offer insights on how to improve the system.

This diagram is a simplified representation, and the actual interactions may involve more details depending on the specific functionalities and technologies used in the implementation of the Helpdesk system.