Database Management System

Project Report

Complaint Management System

Made By :-
Rohan Acharya (330/CO/14)
Robin Ratan (329/CO/14)
Rishabh Singh (327/CO/14)

Acknowledgement

I have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. I would like to extend my sincere thanks to all of them.

I am highly indebted to Dr. Sushma Nagpal Ma'am for her guidance and constant supervision as well as for providing necessary information regarding the project & also for her support in completing the project.

I would like to express my gratitude towards my parents & my institution NSIT, for their kind co-operation and encouragement which helped me in completion of this project.
My thanks and appreciations also go to my colleagues in developing the project and people who have willingly helped me out with their abilities.

Table of Contents

1. Problem Statement
2. Abstract
3. Introduction
4. Entity - Relationship Model
5. Relational Model
6. Data Tables
7. Constraints
8. Functional Dependencies
9. Implementation
10. Advantages of Proposed System
11. Conclusion
12. Future Work

Problem Statement

1. The problem is to build a complaint system which records the information regarding the complaints registered by a customer against the company. It enables the user to effectively communicate with the authorities in the form of a feedback system aimed at improving company policies and features. It also encourages the customers to voice their inconvenience against a product or a company policy, thus increasing the opportunities for the company to improve its efficiency.
2. The complaint system will first display a log in screen to the user, indicating the user to enter his/her ID and the password assigned to the user. The user can either be a customer or an employee. Judging by the ID used for the log in, the system decides whether it's a customer or an employee who is trying to access the system.
3. If a customer has logged in to the system, he/she is greeted to a window that includes options that pertain to the customer's demands. The window contains three options:
    - Add a complaint: When this option is selected by the customer, a window opens which gives the customer a platform for writing and filing a complaint in the database. The complaint

number, which is unique to each complaint is provided automatically. The customer has the option to select a department to which the complaint is to be addressed. The text box prompt is for the customer to write the complaint. By clicking Submit button, the complaint of the customer will be stored in the database.

- Edit complaint: This option enables the customer to edit the complaint registered, meaning that the customer is able to change the complaint registered or delete the complaint registered. The window shows all the complaints filed by the customer against the company. A complaint is selected and edited by writing in the text box or it is deleted.
- Show status: When this option is clicked, the customer is given full list of the complaints filed by the customer with all the relevant information and also a status column is displayed which indicates whether a solution for the complaint has yet been provided or not.

4. If an employee has logged in to the system, he/she is greeted by a window that includes options that are of the employee's interest. This window has three options:

- Add Solution: This option enables the employee to enter a solution of the complaints registered by the customers. Since only a certain number of departments are associated with an employee, the employee will only be able to see the complaints that are regarding his/her departments. The employee can select the complaint out of the list and provide the solution by typing in the textbox.
- Update solution: An employee also has the option to update or rectify the already given solution regarding a registered complaint by clicking this option. Here the employee is provided with the list of solution provided by the employee along with their complaint number and complaint. The employee can change the solution but cannot delete the solution. That is, the employee must either replace the solution with another or let it remain the same.
- Show my solution: This option allows the employee to see all the solutions they have provided in the database till now.

5. The last option, which is common to both the customer and the employee the option to log out of the system. The user, when logged out, can access the system anytime and the data

pertaining to the user will not be removed from the database.

Abstract

This report includes a development presentation of a complaint management system for managing the data which is relevant to a complaint issued in a small company or organisation. The system as such as it has been developed is called Complaint Management System. It consists of functionally related GUI (application program) and Database.
This system consists of an application program, on one hand, and a database (repository of data) on the other. The program performs the basic operations upon the database as retrieving, inserting, updating and deleting data. The logical database model (tables, their content and the relationships between them) responds to the given task and cover the basic requirements. The Interface of the program is user-friendly, and the program is as easy for use as it is possible.
The connections with the database had been established, when a query is needed to be performed upon it. Exception-handling is also taken into account due to eventual exceptions that may occur.
The choice of the programming tools is individual and particular.

Introduction

At the very commencement, the given task at hand is broken down into the following steps:
1. Exploring the available development environments and techniques.
2. Database Analyzing.
3. Database design and Implementation.
4. Program's Structure Analyzing.
5. GUI (Graphical User Interface) constructing.
6. Bringing all the stuff together (controls data binding and functions implementation).
7. Tests.

Each one of these steps could be explained in some brief details as follows:

1. *Exploring the available development environments and techniques*
There is a lot of programming environments available to be used for such kind of elaborations. The point is to choose such an environment that we will be able to operate with in a convenient and easy way. This is more or less optional and individual process, that depends on the developer's experience as well.

2. *Database Analyzing*
It concerns all of the demands, put upon the database content and its functionality. The database should be designed and implemented in a way that the user would expect it to be.

3. *Database design and Implementation*
This step is tightly related with the previous one as it is completely determined by the requirements, analyzed and discussed in step2.

4. *Program's Structure Analyzing*
The application program as an interface between the users and the database should be an accurate "reflection" of the database on the screen; hence a well analyzed and defined structure is needed.

5. *GUI Constructing*
After analyzing the program's structure and defining what it should consist of, a graphical representation of this stuff is needed in order to enable the user to interact with the data.

6. *Bringing all the stuff together*
The next step that should be taken is connecting the program with the database and performing the necessary functionality upon all of the controls.

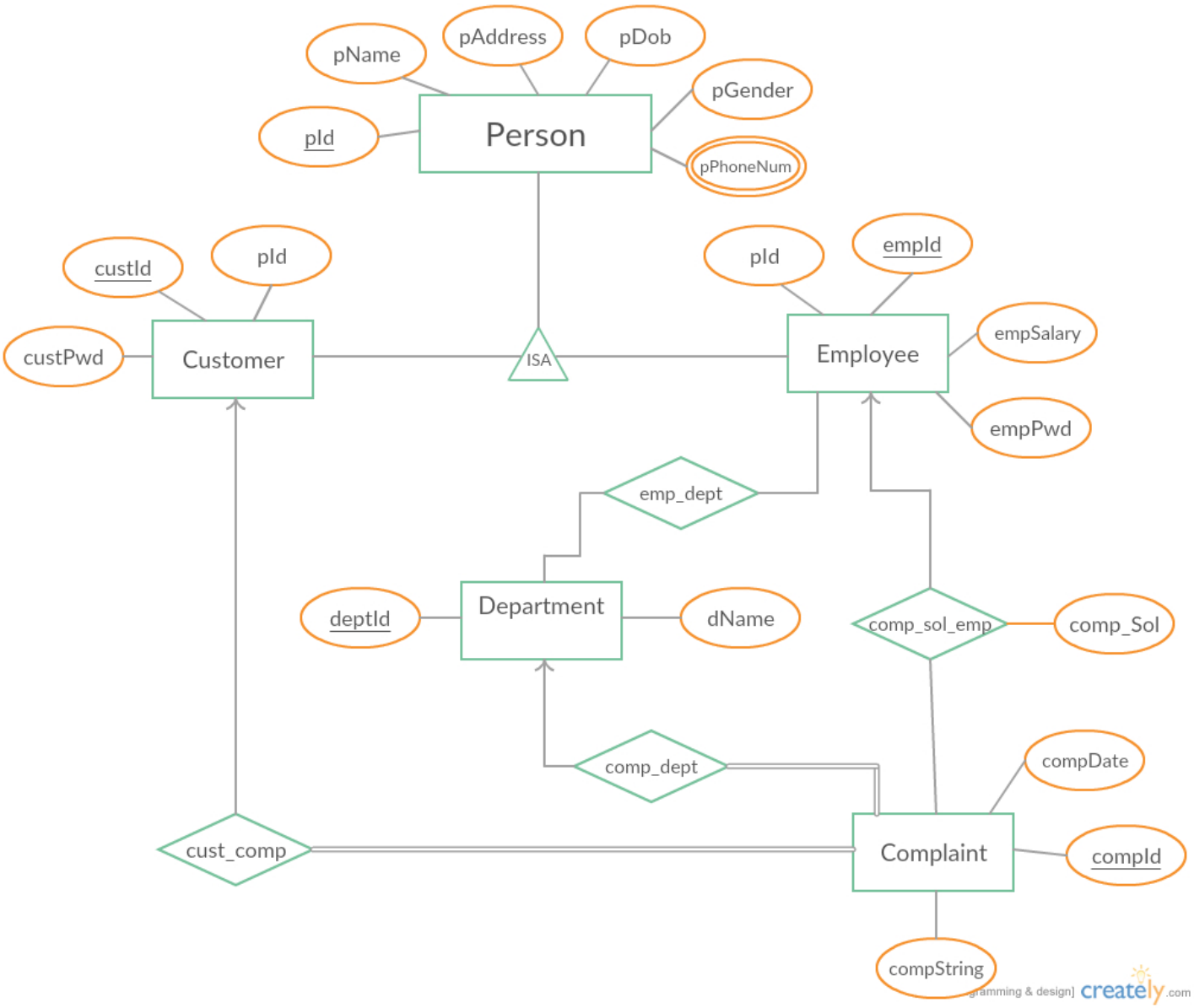
7. *Tests*
To ensure that everything works properly and as it has been expected, test performance has to be done upon the system's functionality.

In our implementation of the Complaint Management System the various complaints are stored in a database along with the complaint ID (compId), the complaint itself (compString), the date when the complaint is posted (compDate), the customer ID of the customer posting the complaint (custId) and the department ID of the department under which the complaint is being posted (deptId). The customer needs to sign into the system using their customer ID (custId) and password (custPwd) to be able to add a new complaint, or to modify/delete any previous complaint posted by them. On posting a new complaint, the system generates a complaint ID (compId) which can be used by the customer to access the complaint at a later time. The solution to any complaint is stored with the complaint ID, it's solution (compSol), and the employee ID of the concerned employee who has solved the complaint (if at all). The employee is required to sign in as well, using their respective ID (empId) and password (empPwd), to be able to view the status of and solve the pending complaints in the system. Employees can only solve those complaints which have been posted under the department that they belong to.
A separate relation (person_phone) is used to link a person (identified by their person ID) to multiple phone numbers. Both the employees and the customers have a unique person ID (pId) associated with them. Various common traits to both employee and customer are stored in the relation person, such as Name, DOB, Gender and Address. The employees

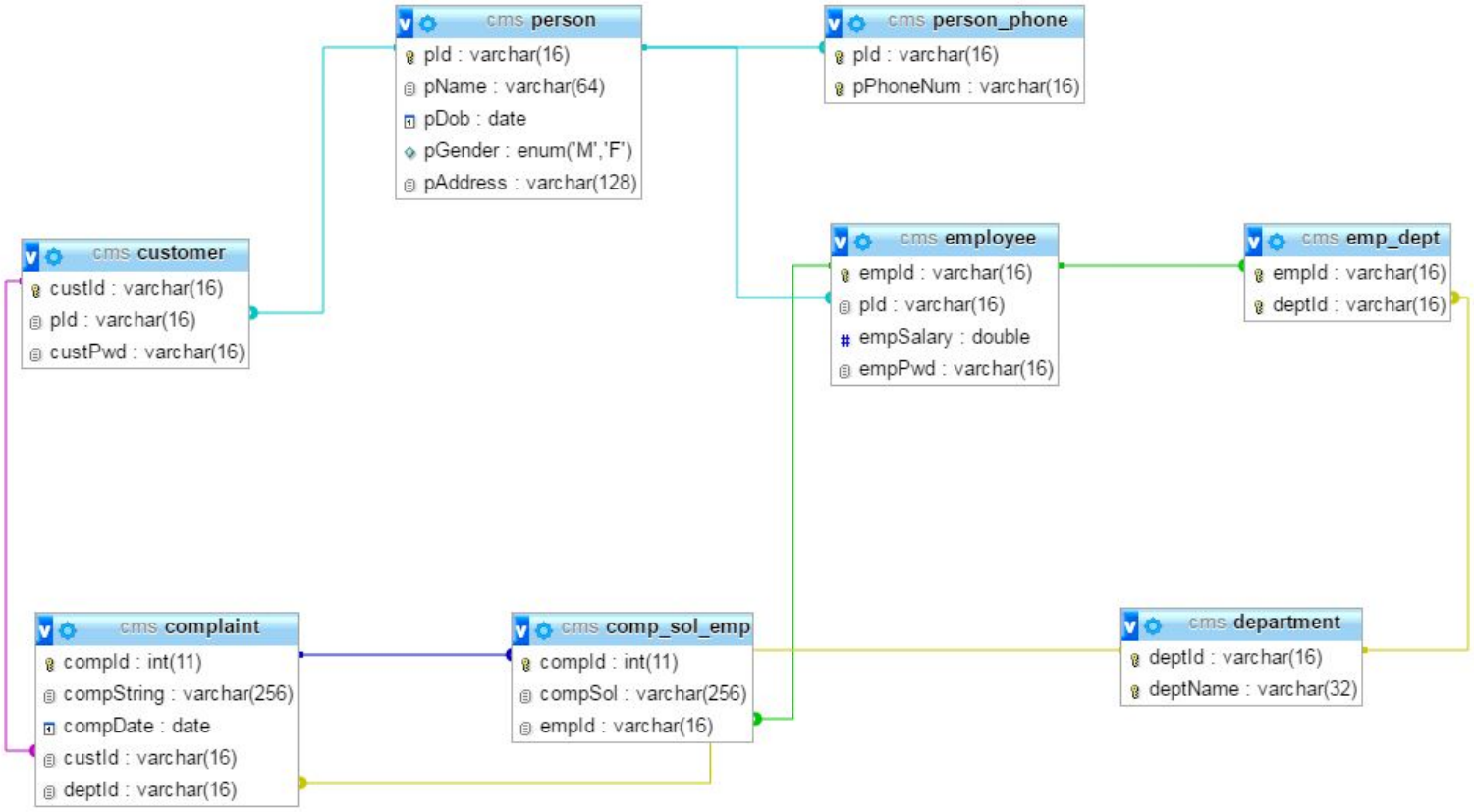
can be uniquely identified based on their employee ID, and the customer based on their customer ID. The relation emp_dept links an employee ID (empId) with a department ID (deptId) and the relation department links a department ID (deptId) with the name of the department (deptName).

Entity - Relationship Model

E-R Model



Relational Model



Data Tables

1. comp_sol_emp

Column	Type	Attributes	Null	Default	Extra	Links to
compId	int(11)		No			-> complaint.compId ON UPDATE RESTRICT ON DELETE RESTRICT
compSol	varchar(256)		No			
empId	varchar(16)		No			-> employee.empId ON UPDATE RESTRICT ON DELETE RESTRICT

2. complaint

Column	Type	Attributes	Null	Default	Extra	Links to
compId	int(11)		No		auto_increment	
compString	varchar(256)		No			
compDate	date		No			
custId	varchar(16)		No			-> customer.custId ON UPDATE CASCADE ON DELETE CASCADE
deptId	varchar(16)		No			-> department.deptId ON UPDATE CASCADE ON DELETE CASCADE

3. customer

Column	Type	Attributes	Null	Default	Extra	Links to
custId	varchar(16)		No			
pId	varchar(16)		No			-> person.pId ON UPDATE CASCADE ON DELETE CASCADE
custPwd	varchar(16)		No			

4. department

Column	Type	Attributes	Null	Default	Extra	Links to
deptId	varchar(16)		No			
deptName	varchar(32)		No			

5. emp_dept

Column	Type	Attributes	Null	Default	Extra	Links to
empId	varchar(16)		No			-> employee.empId ON UPDATE CASCADE ON DELETE CASCADE
deptId	varchar(16)		No			-> department.deptId ON UPDATE CASCADE ON DELETE CASCADE

6. employee

Column	Type	Attributes	Null	Default	Extra	Links to
empId	varchar(16)		No			
pId	varchar(16)		No			-> person.pId ON UPDATE CASCADE ON DELETE CASCADE
empSalary	double		No			
empPwd	varchar(16)		No			

7. person

Column	Type	Attributes	Null	Default	Extra	Links to
pId	varchar(16)		No			
pName	varchar(64)		No			
pDob	date		No			
pGender	enum('M', 'F')		No			
pAddress	varchar(128)		No			

8. person_phone

Column	Type	Attributes	Null	Default	Extra	Links to
pId	varchar(16)		No			-> person.pId ON UPDATE CASCADE ON DELETE CASCADE
pPhoneNum	varchar(16)		No			

Constraints

- Primary Key Constraint :

The PRIMARY KEY constraint uniquely identifies each record in a database table. Primary keys must contain UNIQUE values, and cannot contain NULL values. A table can have only one primary key, which may consist of single or multiple fields.

The following primary keys are used in our model :

I. custId (for customer)
II. compId (for complaint)
III. compId (for comp_sol_emp)
IV. pId (for person)
V. pId (for person_phone)
VI. empId (for employee)
VII. empId (for emp_dept)
VIII. deptID (for department)

- Foreign Key Constraint :

A FOREIGN KEY is a key used to link two tables together. A FOREIGN KEY in a table points to a PRIMARY KEY in another table.

The following foreign keys are used in our model :

I. pId references person (creates a FOREIGN KEY on the "pId" column when the "customer" table is created)
II. pId references person (creates a FOREIGN KEY on the "pId" column when the "employee" table is created)
III. pId references person (creates a FOREIGN KEY on the "pId" column when the “person_phone" table is created)
IV. custId references customer (creates a FOREIGN KEY on the "custId" column when the "complaint" table is created)
V. compId references complaint (creates a FOREIGN KEY on the "compId" column when the “comp_sol_emp” table is created)
VI. deptId references department (creates a FOREIGN KEY on the "deptId" column when the "complaint" table is created)
VII. empId references employee (creates a FOREIGN KEY on the "empId" column when the “comp_sol_emp” table is created)
VIII. empId references employee (creates a FOREIGN KEY on the "empId" column when the “emp_dept" table is created)

IX. deptId references department (creates a FOREIGN KEY on the "deptId" column when the “emp_dept" table is created)

- Not Null Constraint :

By default, a column can hold NULL values. The NOT NULL constraint enforces a column to NOT accept NULL values. This enforces a field to always contain a value, which means that you cannot insert a new record, or update a record without adding a value to this field.
All Columns used in our model do NOT accept NULL values.

USER CHARACTERISTICS

Every user:

- Should be comfortable with basic working of the computer
- Should know English
- Must carry a login ID and password used for authentication

DRAWBACKS

- The GUI restricted to English
- Login ID and password used for identification of user/administrator. There is no facility for a guest login.

Functional Dependencies

A functional dependency is a constraint between two sets of attributes in a relation from a database. In other words, functional dependency is a constraint that describes the relationship between attributes in a relation.

The following non-trivial functional dependencies are present in our model :

- person:

pId->pName, pAddress, pDob, pGender

- customer:

custId-> custPwd

- employee:

empId-> empSalary, empPwd

- complaint:

compId->compString, compDate

- department:

deptId->dName

- comp_sol_emp:

compId->compSol, empId

- person_phone:

pId->pPhoneNum

In all of the above schemas, the attribute on the LHS is also a super key (primary key, as well), therefore all of these schemas are in Boyce-Codd Normal Form.

For the remaining schema, there is only trivial functional dependency which is :

- emp_dept:

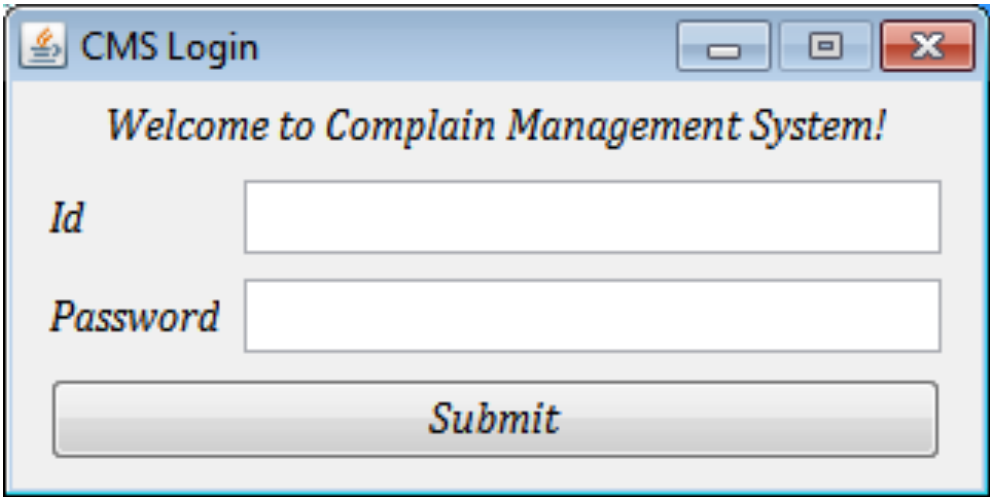
empId,deptId->empId,deptId

Since this is a trivial functional dependency, this schema is also in Boyce-Codd Normal Form.

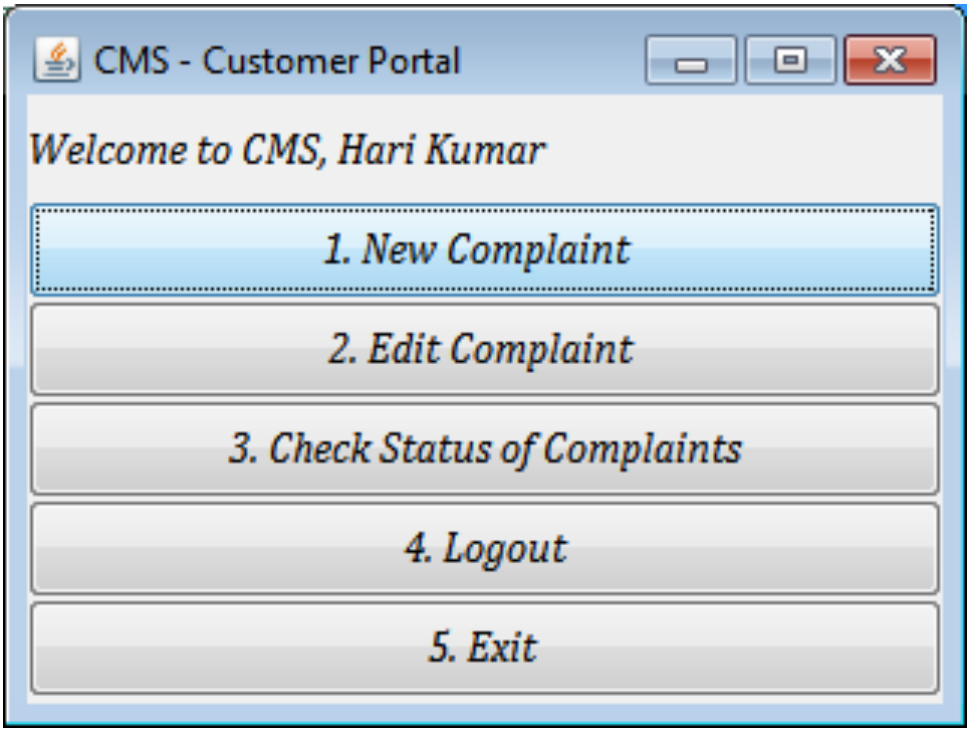
All the relations in the database have been normalized to BCNF form and thus contain no redundancy related to the functional dependencies.

Implementation

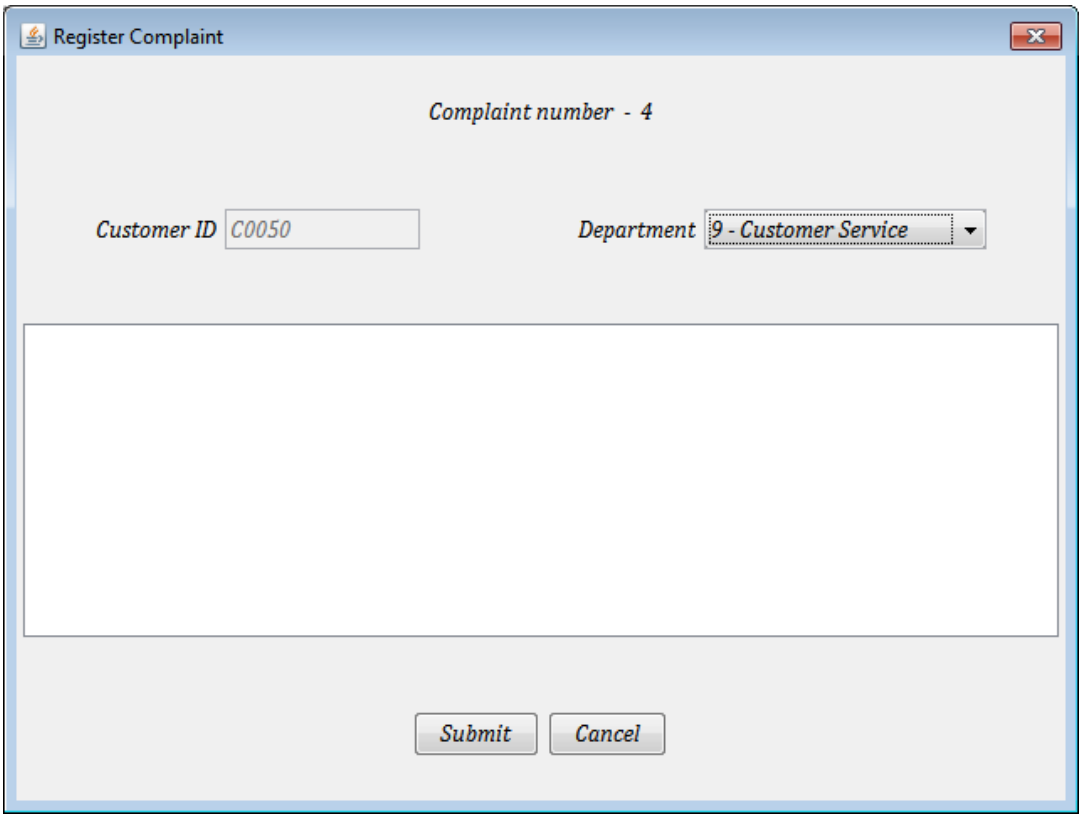
1. Login Screen



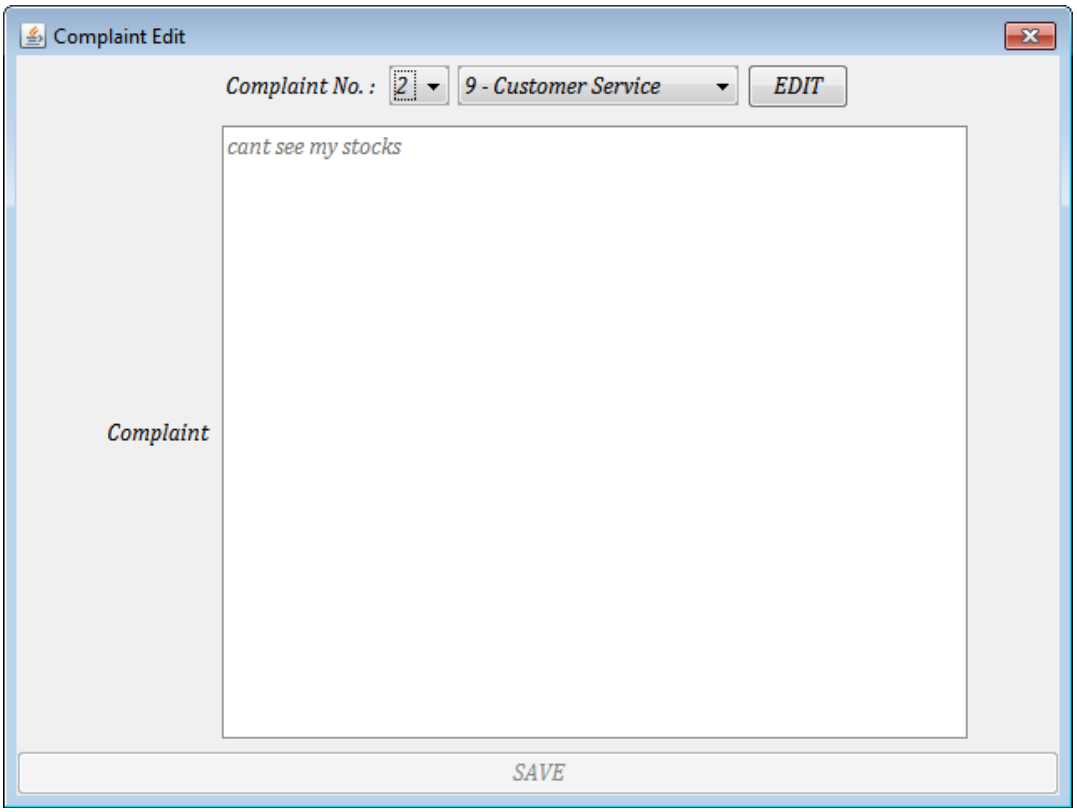
2. Customer Portal



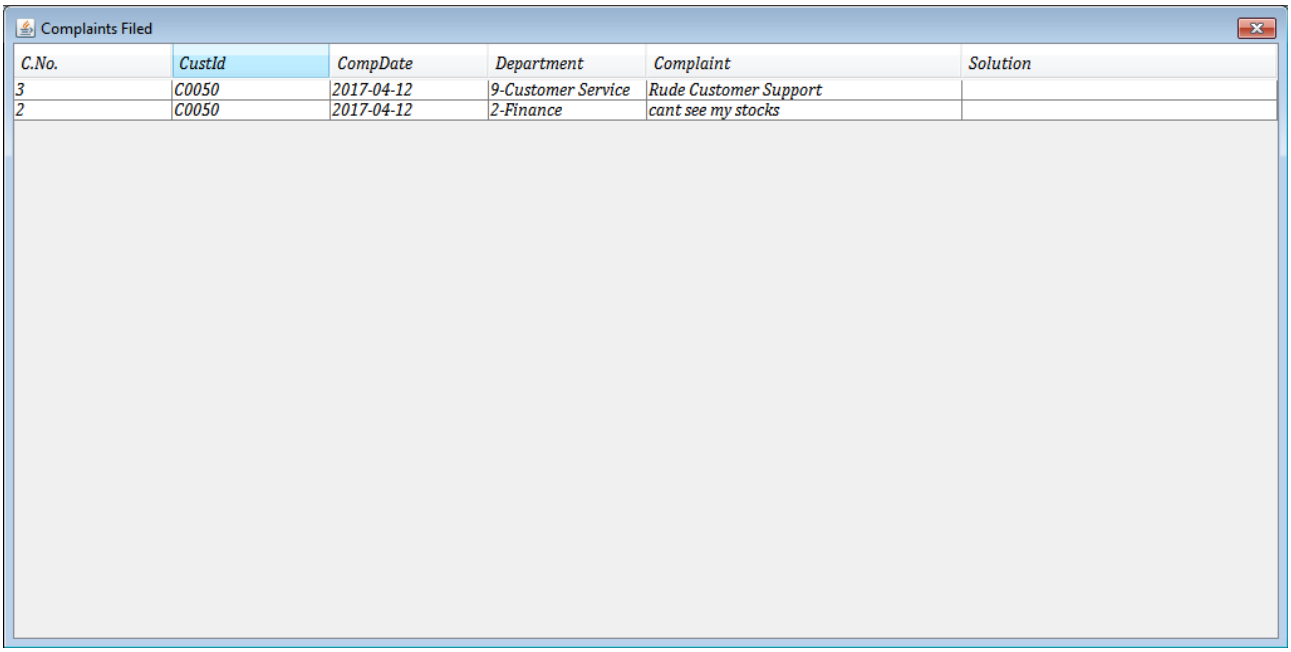
2.1 New Complaint



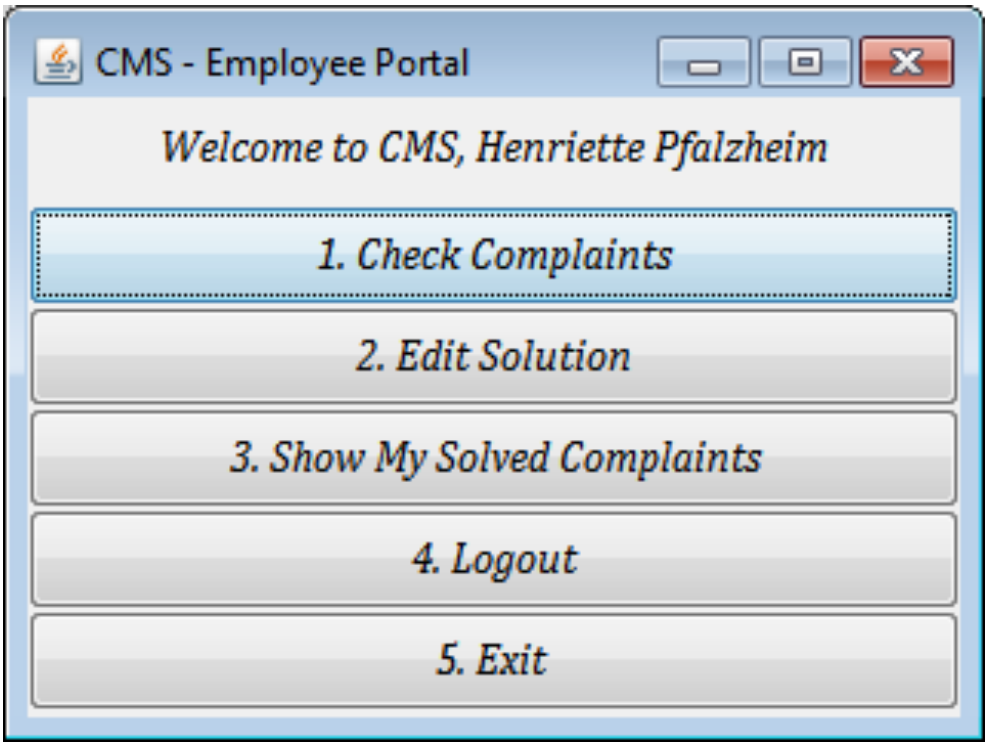
2.2. Edit Complaint



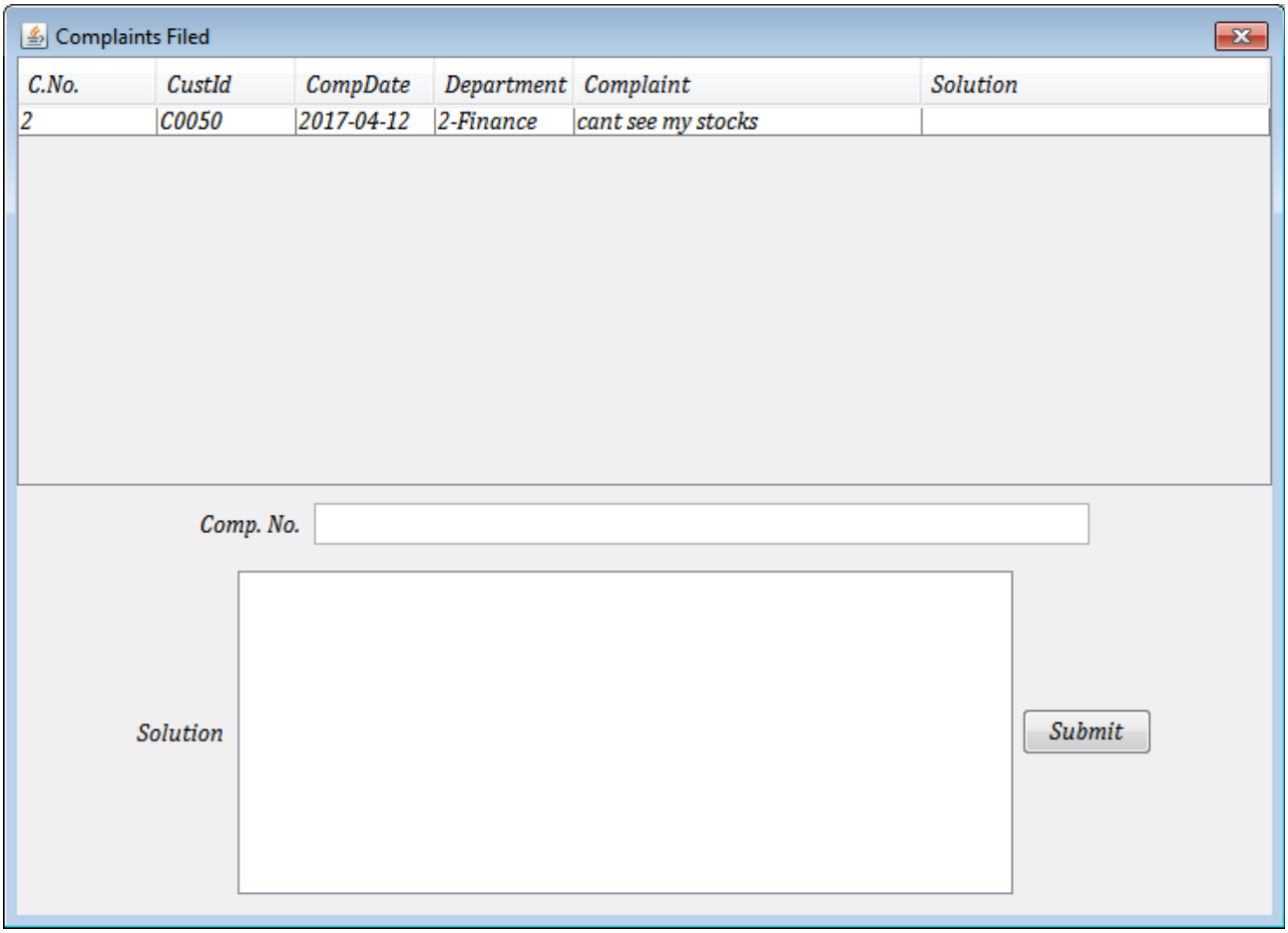
3. Complaints Filed



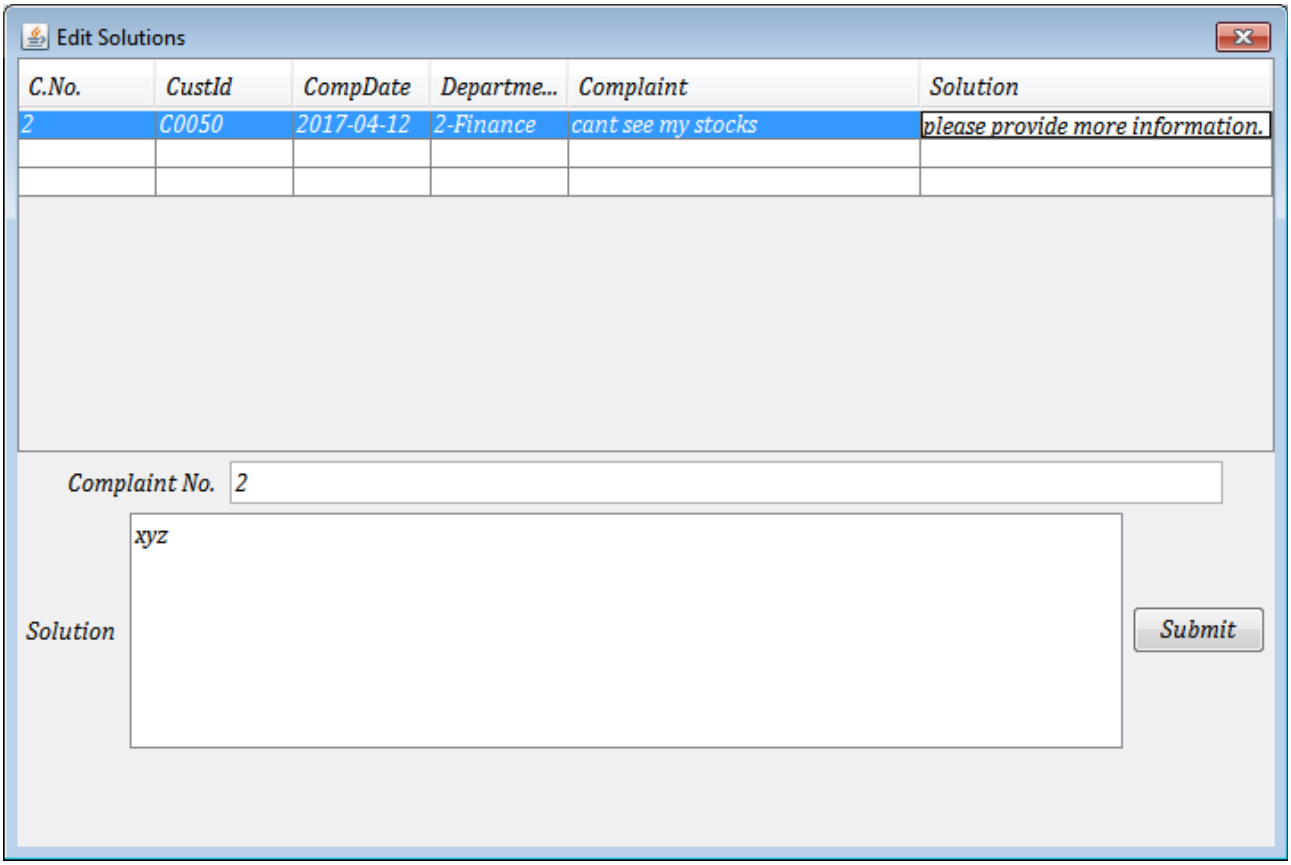
2. Employee Portal



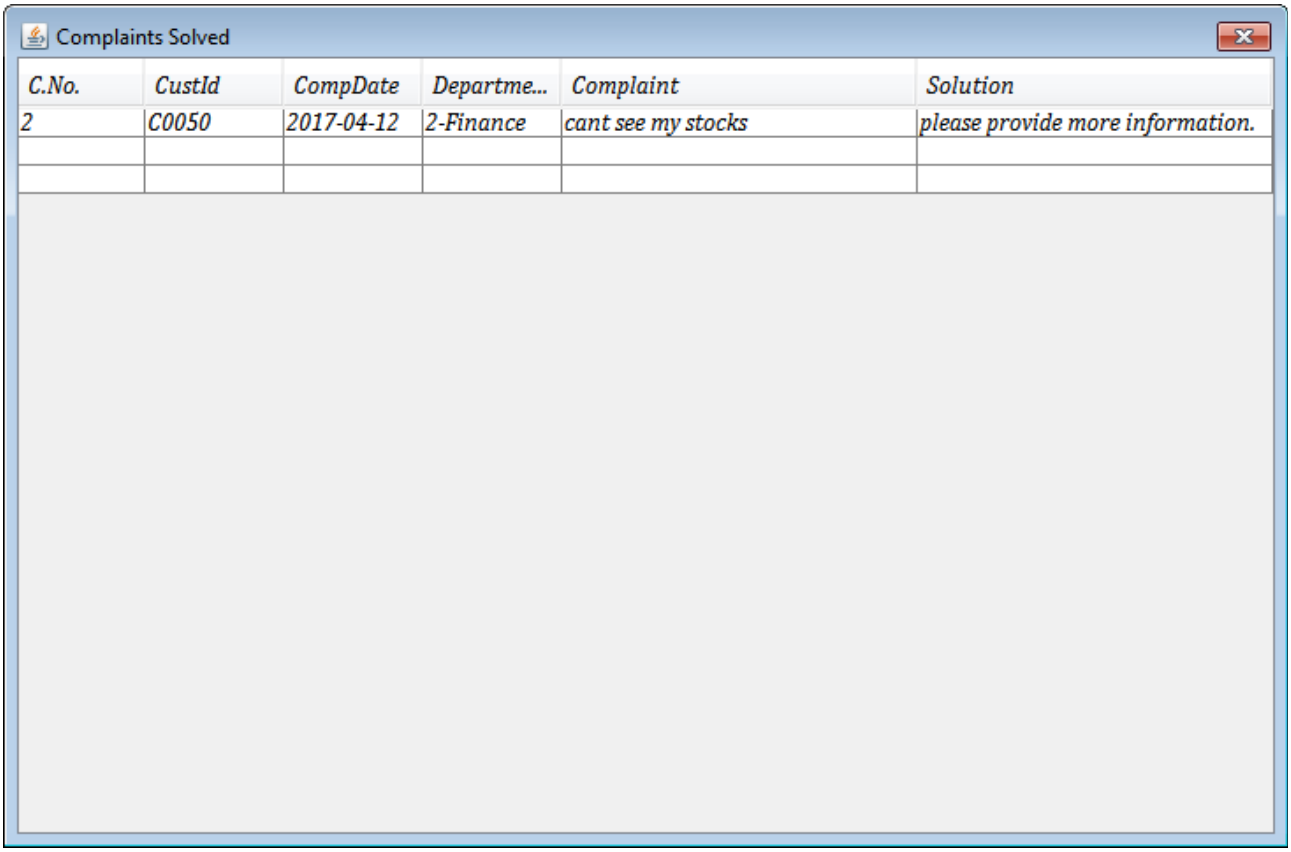
3.1 Solve Unsolved Complaints regarding department of currently logged in employee



3.2. Edit Solutions given previously by current logged in employee



3.3 Complaints Solved by current logged in employee



Advantages of Proposed System

- Very simple and easy to implement
- Security of data
- Ensure data accuracy
- Reduces the damages of machine
- Minimizes manual data entry
- Greater efficiency
- User friendly and interactive
- Less time consuming

Conclusion

In this report, a complaint management system's development has been presented. It was emphasized on the basic steps, consequently taken during the project's development course as a particular attention was turned to the basic operative functions performed upon the data into the database.
The report's content comprises the whole task solution, going through the database, the application's analyze and construction, and finishing with the implementation and the advantages of the system.
An application has been developed using MySQL development and java database programming connectivity via JDBC driver so as to meet the requirements of an organization, thereby ensuring quality performance. The data can be accessed, manipulated and retrieved very easily. To conclude this software has proved to be a user friendly interface.

Future Work

- The employees can have a point ranking system on the basis of how many complaints they have solved and can be allotted points for this. The earlier a complaint is solved, the greater is the number of points awarded to the respective employee. This system can be used to judge the performance of various employees in the organisation.
- The customers can give a feedback and a rating to the organisation depending on their level of satisfaction with the complaint management system, which can be used to improve the system based on the customer’s review.