**Payroll/Employee Management System for Beimnet Café & Restaurant**

**Chapter 1**

* 1. **Project Overview**

This project is going to solve the problem that Beimnet café and restaurant has regarding to the payroll system of the employees.

The system contains employee’s details such as employee’s names, employees address, employee’s salary, etc. in computerized form in order to update when necessary, retrieve and the like. Currently the Hotel is using manual system of storing his employee’s information and paying their salaries. This storing of employees data and paying their salaries in the manual form has many problems:-

* It may result to easily losing of data’s.
* The data’s cannot be available for long periods because they are stored on paper. Papers do not have such longer life.
* Finding of employee’s information were time consuming.

Due to the above listed problems, we are going to design a system that keeps employee’s details in computerized way to reduce the problems occurred.

The proposed system perform the following tasks:-

* Recording employee’s information
* Payroll system. Its function is to store employee’s salary and when paying salary of employee there some other things like tax, health…. That can be reduced from their salary.
* Database system to store employee’s information details.
  1. **Objective of the Project**

General Objective

General objective of our project is designing Data Base system to handle data of employee’ to increase the efficiency of the hotel on data handling and payroll system of his employee’s. Design the solution for the problem we had seen in the past phase.

/////To design the database model and relationship of the entities

Specific Objective

* Make document organization (i.e. employee’s basic information) computerized with the futures like update, delete, and register if a new employee has been hired.
* Calculate the total amount of salary that an employee could get including all amounts due to each such as tax. // the is not specific objective..

What steps do you go to achieve the general objective (designing the database)???????

* 1. **Scope**

Make the payroll system of the Beimnet café & restaurant computerized.

* Make a system to record the basic information of the employees such as name, address, identification number, phone number, salary.
* Make a system to delete, update or upgrade the employee information if there are any changes (i.e. if salary of employee increased, or decreased or the employee may be leaving his work).
* Make a system calculates the total amount of salary that an employee could get including all amounts due to each such as tax.
* Making a system password protected for a security of data.

**//This is phase three part**

* 1. **Functional and Non-functional requirements of the proposed system**

**1.4.1 Functional requirements**

Functional requirements describe what the system should do and functions that can be captured in use cases. They are behaviors that can be analyzed by drawing sequence diagrams including ER-diagrams, charts, tables etc. These functional requirements are those things and inquiries that are required to the system as an input so as the system will work based on it.

This proposed system registers new employee’s information and calculates the total amount of salary that an employee gets including all their dues. It also update basic information every time when new employee been hired and an employee leaves his work.

The basic information about employee includes Full-Name, Identification number, Address, Salary,

**1.4.2 Non-functional requirements**

Besides its main goal or function, the system also requires other additional requirement for its functions to work well. This are called non-functional requirements. Non- functional constraints are global constraints on a software system. For example requirements such as development cost, operational cost, performance, reliability (accuracy, completeness and integrity), Maintainability, portability, robustness, security etc. They often known as software qualities. Among these, keeping security of data is the first and the most.

**Chapter 2**

**2.1 Designing The Proposed System**

The proposed system design describes the relevant architecture of the system what it should be, what it looks like and how it works. The system was constructed based on different entities, which includes an attributes inside it. These entities are employee, manager, salary. Each of the entities contains their own attributes. Those entities have a relationship with each other as the case with the real life. This relationship can be one-to-one, one-to-many, many-to-one, many-to-many, ternary relationship or binary relationship.

**2.1.1 Defining Entities**

An **entity** is a group of attributes with rules or data in common. It often represents a thingin the real world with an independent existence. An entity may be an object with a physical existence (for example, a particular person, car, house, or employee) or it may be an object with a conceptual existence (for instance, a company, a job, or a university course).

An **attribute** the particular properties that describe entity.

In our case,

* An EMPLOYEE entity is described by having an attributes employee’s name, age, address, salary, etc.
* A SALARY entity is described by having an attributes medical allowance, basic…
* An OWNER entity is described by having an attributes owner id and owner name.

**2.1.2 Entity Relationship Diagram**

An Entity Relationship Diagram (ERD) is a visual representation of different data using conventions that describe how these data are related to each other. Here we have figure out the relationship among each entity based on the function of each entity.

1. **EMPLOYEE entity**

We have many attributes under this entity type.

.

Employee

// **show the primary key.. Apply this comment for every entity**

**// phone-No is multi-valued attribute, mobile phone,** home **phone… show this..**

**// Age will not be attribute// make it birth date , then you will calculate the age(implementation).**

**// first name last name**

**Which attribute going to hold the role of the employee?**

**Eg the employee may be casher or manager or waiter and so on..**

Figure 1:- ER diagram for EMPLOYEE entity.

1. **SALARY Entity**

SALARY

How do you know a specific salary is for a specific employee?

Here you have to use emp\_id as a foreign key .

Figure 2:- ER Diagram of SALARY entity.

1. **Owner Entity**

OWNER

Figure 3:- ER Diagram of OWNER Entity.

**// owner will not be an entity for your system, remove it**

**???? I think your system has a login or authentication feature, if that is the case which entity will hold username and password information for each employee?**

**???? where is the diagram that shows the relationship between entities.**

2.2 **Schema or Table description of each entity**

Here is the schema or table description of each entity along with their corresponding attributes.

EMPLOYEE entity

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Emp\_Id | Name | Address | Sex | City | Kebele | BDate | Age | HNo | Phone no |

SALARY entity

|  |  |  |  |
| --- | --- | --- | --- |
| Medical allowance | Basic salary | Tax | Retirement |

OWNER entity

|  |
| --- |
| OName |

**2.3 Normalization**

Normalization is the process of organizing the fields and tables of a relational database to minimize redundancy (repetition of data). Normalization usually involves dividing large tables into smaller (and less redundant) tables and defining relationships between them.

**First Normal Form (1NF)**: The domain of each attribute contains only atomic values, and the value of each attribute contains only a single value from that domain. Relation should have no non-atomic attributes or nested relations.

**Second Normal Form (2NF)**:- Here we create separate tables for sets of values that apply to multiple records and relate these tables with a “foreign Key’’. Since by first normalization scheme (1NF) many potential problems associated with the data redundancy occur more frequently and not all attributes are fully dependent on the primary key, these anomalies are required. This type of problem only arises when the key is a composite key.

**Third Normal Form (3NF):** Every non-prime attribute is non-transitively dependent on every candidate key in the table. The requirements to satisfy the 3NF:

* All requirements for 2NF must met.
* Eliminate fields that do not depend on the primary key;
* That is, any field that is dependent not only on the primary key but also on another field must move to another table.

**2.3.1 Normalization for EMPLOYEE entity**

### Un-Normalized Form (UNF)

If a table contains non-atomic values on each row, it is said to be in UNF. For instance,

Employee(EID, Name, Address, City, Sex, Kebele, PhoneNumber, Age, BDate, HNo)

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Attribute** | EID | Name | Address | City | Sex | Kebele | PhoneNumber | Age | BDate | HNo |
| **Data type** | int | Vchar | Vchar | Vchar | Vchar | Vchar | Int | Int | Date | integer |

### First Normal Form (1NF)

The above table in UNF can be used to create the following table in 1NF. The only attribute values permitted by 1NF are single atomic (or indivisible) values. The table contains atomic values.so we should have to create table for Address and Employee.

**Address (EID, PhoneNumber, City, Kebele, HNo)**

Address

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| EID | PhoneNumber | City | Kebele | HNo |

**Employee (EID, Name, Sex, Age)**

Employee

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| EID | Name | Sex | Age | BDate |

Notice that EID is the primary key of the Employee while PhoneNumber is the partial key of the nested relation (Address); that is, within each tuple, the nested relation must have unique values of PhoneNumber. To normalize this into INF, we remove the nested relation attributes into a new relation and propagate the primary keyinto it; the primary key of the new relation will combine the partial key with the primary key of the original relation. Decomposition and primary key propagation yield the schemas Address and Employee as shown.in the above.

**Second Normal Form (2NF)**

To be in 2NF it must be-

* In 1NF
* Eliminate partial dependencies.

There is a partial dependency using (EID, BDate) as candidate key and Age, which is partially dependent on candidate key. i.e.

* Age is dependent on BDate rather than the primary key (EID).
* Therefore we have to create a separate table for Age attribute as follows.

**Age (EID, BDate)**

|  |  |
| --- | --- |
| EID | BDate |

**Address (EID, PhoneNumber, City, Kebele, HNo)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| EID | PhoneNumber | City | Kebele | HNo |

**Employee (EID, Name, Sex,)**

|  |  |  |
| --- | --- | --- |
| EID | Name | Sex |

**Third Normal Form (3NF)**

To be in 3NF it must be-

* should be in 2NF
* Eliminate transitive dependencies (indirect dependency on PK because of some other non-key attribute).

**2.3.2 Normalization for SALARY entity**

**(We are having difficulties on identifying the primary key for SALARY & OWNER ENTITY, that’s why we are not doing the normalization for them, PLEASE HELP US)**

Salary (EID, Medical Allowance, Basic, Tax, Retirement)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Attribute** | EID | Medical Allowance | Basic | Tax | Retirement |
| **Data Type** | integer | Integer | Integer | Integer | Integer |

**2.3.3 Normalization for OWNER entity**

Owner (OwnerId, OwnerName)

|  |  |
| --- | --- |
| **Attribute** | OwnerName |
| **Data Type** | Varchar |

**Chapter 3**

**Database Security of the System**

One of the main targets that should not be forgotten is security of the system. We mean that data should be kept in a way that it cannot be accessed by unauthorized person. Thus, it should include Confidentiality, integrity, and availability of the data. The Owner or the person who is given the authority should store data of each employee. Nobody have the power to see the data except those who stored it and that should be secured. In case of data record no body have to update and inserts data. Since all data will available in the system, it should be done by the system. Over all permission is given by manager who will be the owner of the database.

To keep security of the system we preferred to make username and password. Every have its own permission. As a result, he/she can see functionalities that allowed for him/her respectively

// I did not see the normalization part; I mean I saw it partially.

// correct the document based on the comments. And think your system deeply, it may have more entities. And apply the changes to the normalization part.

**Chapter 4**

**Future Phase**

By the help of this project, in the future phase we aimed to solve problems of Beimnet café & restaurant has regarding to payroll system using an appropriate language and system as possible as we can depending on the conditions.