ELLAA HOTEL

-Yash Vadalia (201001015) and Ayush Minocha (201001014)

Application Documentation:

FIRST TIER:

The database application of Ellaa hotel has a login page, in which a user can login as a database administrator or an employee and the database privileges are given accordingly .

Administrator privileges: The database administrator has complete access to all the records of all the tables in the database and complete permissions to insert or modify/delete the data records apart from viewing the data. Also the administrator can upgrade the privileges of any any normal user to administrator privileges.

The employee users on the other hand, have no rights to add or delete/modify the data, they can only view the data. An employee user can log in with his distinct user name and password.

Administrator interface :
Once the administrator logs in, he has access to four functionalities:
- Insert
- Delete
- Update
- View

which are common to all the five tables which are Employee, Dependant, Department, Guest and Reservation.

There will be separate forms for each functionality but only the administrator can access all of them. A normal user on the other hand has access to only the view form.

INSERT FORM:
For each table in the database the form will ask the user for values to those fields corresponding to the columns in that database table. The administrator will enter the required records through text boxes and hit submit to insert the values. There will be five insert forms in total, for five tables respectively. After an insert is made, an appropriate message is displayed along with the updated table.

UPDATE FORM:
As time passes the attributes of the tuple might need a change and to reflect the reality we need to update the data which in this case can be done via an update form. For updating any tuple in a table we identify it by its corresponding primary key and hence it is highly necessary to input the primary key. The rest of the form will be similar to insert form and the administrator can enter the updated values into the fields. After an update, an appropriate message is displayed with the modified table.

DELETE FORM:
To delete a record/records from a table, the administrator will have to enter the primary key of the corresponding table. After deletion, appropriate message along with the current state of the table is displayed.

VIEW FORM:
Administrator can either select all or specific set of id's(primary key) of a particular table to view the selected records with its entire details. The id's will be available in the drop down menu.

REGISTRATION FORM:
It will ask for basic details of the user and will register accordingly.

EMPLOYEE(Normal User) INTERFACE:

After an employee logs in,he can only view the data of all the tables.

MIDDLE TIER:

The links to some of the forms are :

web.iiit.ac.in/~ayush.minocha/insert_depend.php
web.iiit.ac.in/~ayush.minocha/insert_emp.php
web.iiit.ac.in/~ayush.minocha/insert_depend.php
web.iiit.ac.in/~ayush.minocha/delete.php
web.iiit.ac.in/~ayush.minocha/del_dep.php

THIRD TIER:

```
CREATE DATABASE EllaaHotel;

CREATE TABLE Guest (
PhNo            INT(10)         NOT NULL,
Address         VARCHAR(50)     NOT NULL,
FName           VARCHAR(30)     NOT NULL,
MName           VARCHAR(30)     NOT NULL,
LName           VARCHAR(30)     NOT NULL,
GuestId         INT(11)         NOT NULL,
PRIMARY KEY (GuestId)
);

CREATE TABLE Reservation (
CheckInDate     DATE            NOT NULL,
CheckOutDate    DATE            NOT NULL,
ResId           INT(11)         NOT NULL,
GuestId         INT(11)         NOT NULL,
PRIMARY KEY (ResId),
FOREIGN KEY (GuestId)           REFERENCES Guest(GuestId)
);

CREATE TABLE Employee (
PhNo            INT(10)         NOT NULL,
Address         VARCHAR(50)     NOT NULL,
FName           VARCHAR(30)     NOT NULL,
MName           VARCHAR(30)     NOT NULL,
LName           VARCHAR(30)     NOT NULL,
EmpId           INT(11)         NOT NULL,
Sex             VARCHAR(10)     NOT NULL,
DOB             DATE            NOT NULL,
Absents         INT(2)          NOT NULL,
DuesLeft        DOUBLE(7,2)     NOT NULL,
DepNo           INT(11)         NOT NULL,
PRIMARY KEY (EmpId),
FOREIGN KEY (DepNo)             REFERENCES Department(DepNo)
);

CREATE TABLE Department (
DepNo           INT(11)         NOT NULL,
DepName         VARCHAR(30)     NOT NULL,
MgrSSN          INT(11)         NOT NULL,
PRIMARY KEY (DepNo),
UNIQUE(DepName),
```

```
FOREIGN KEY (MgrSSN)                REFERENCES Employee(EmpId)
);

CREATE TABLE Dependant (
Name                VARCHAR(30)         NOT NULL,
Sex                 VARCHAR(10)         NOT NULL,
RelEmp              VARCHAR(10)         NOT NULL,
Age                 INT(2)              NOT NULL,
Id                  INT(11)             NOT NULL,
FOREIGN KEY (Id)                    REFERENCES Employee(EmpId)
);
```

INTEGRITY CONSTRAINTS (RELATIONAL MODEL)

Domain Constraints

1. Employee:
    - Phone Number – Integer
    - Address – String
    - First Name – String
    - Middle Name – String
    - Last Name – String
    - Gust Id – Integer
2. Reservation:
    - Check In Date – Date
    - Check Out Date - Date
    - Reservation Id – Integer
3. Employee:
    - First Name – String
    - Middle Name – String
    - Last Name – String
    - Sex – String
    - Date Of Birth – Date
    - Address – String
    - Absents Per Month – Integer
    - Employee Id – Integer
    - Phone Number – Integer
    - Dues Left – Double
4. Department:
    - Department Name – String
    - Department Number – Integer
5. Dependent:
    - Name - String
    - Sex – String
    - Relation To Employee – String
    - Age – Integer