

**DEPARTMENT OF COMPUTER SCIENCE AND
ENGINEERING**

**RELATIONAL DATABASE
MANAGEMENT SYSTEM**

A MINI PROJECT REPORT

On

THE YELLOW PIXELS

Submitted by

1) Nikhil Kamath	4NM16CS082
2) Pawan Bhandarkar	4NM16CS090



NITTE
EDUCATION TRUST

N.M.A.M. INSTITUTE OF TECHNOLOGY
(An Autonomous Institution affiliated to Visvesvaraya Technological University, Belagavi)
Nitte – 574 110, Karnataka, India



NITTE
EDUCATION TRUST

N.M.A.M. INSTITUTE OF TECHNOLOGY

(An Autonomous Institution affiliated to Visvesvaraya Technological University, Belagavi)

Nitte – 574 110, Karnataka, India

(ISO 9001:2015 Certified), Accredited with 'A' Grade by NAAC

☎: 08258 - 281039 - 281263, Fax: 08258 - 281265

Department of Computer Science and Engineering

B.E. CSE Program Accredited by NBA, New Delhi from 1-7-2018 to 30-6-2021

CERTIFICATE

Certified that the project work entitled “Violin Synthesizer” is a bonafide work carried out by P PAWAN BHANDARKAR, 4NM16CS090 and NIKHIL KAMATH, 4NM16CS082 in partial fulfilment of the requirements for the award of Bachelor of Engineering Degree in Computer Science and Engineering prescribed by Visvesvaraya Technological University, Belgaum during the year 2017-2018.

Signature of Guide

Sudeepa K B

Associate Professor

Dept of CSE

NMAMIT Nitte

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to our supervisor Mr Sudeepa K B for his help throughout the course of the project. Also, we would like to thank the Head of our Department, Dr K R Udaya Kumar Reddy for providing us with this great opportunity. Working on this project has helped us in improving our knowledge on concepts like database design, establishing server connections in MySQL, querying and updating on real-world databases and other related concepts.

Table of Contents

<u>Contents</u>	<u>Page No.</u>
1. <i>Abstract</i>	5
2. <i>Introduction</i>	6
3. <i>ER Diagram</i>	7
4. <i>Implementation</i>	9
5. <i>Screenshots</i>	12
6. <i>Conclusion</i>	
7. <i>References</i>	

Abstract

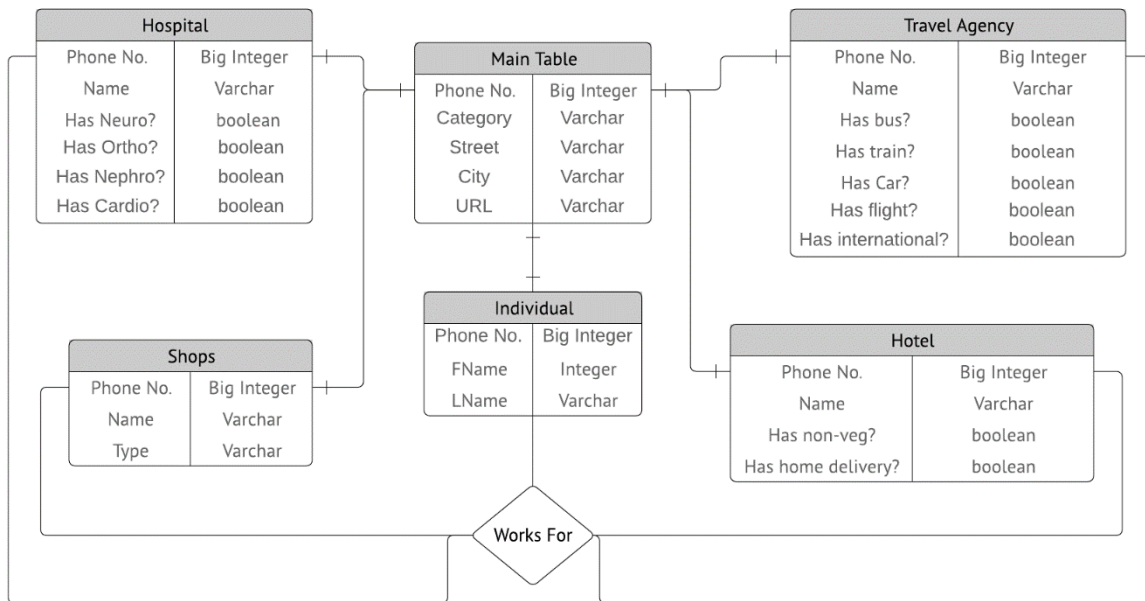
This project has been inspired by the popular telephone directory book, The Yellow Pages that have been in use all over the country for so many years. With the digital revolution upon us, we felt a strong need to digitize the Yellow Pages and hence we've implemented our take on the same which we'd like to call, The Yellow Pixels. Due to a shortage of real-world resources for the complete implementation of this project, our model serves as a proof of concept and hence, is limited to contain data regarding five different categories: Hotels, Hospitals, Travel Agencies, Shops and Individuals. By navigating the UI, users are able to select the category of their interest and by specifying the attributes they wish to search by, quickly fetch the contact and address information from the database and display it on the application.

Introduction

The main goal of this project is to provide users with an easy-to-use interface for navigating a vast database of contact information, regarding various individuals and organizations. The database consists primarily of 5 tables, each one corresponding to details specific to each category. A sixth table, called the “Main Table” is also present and it stores information that is common to all the other tables such as the Addresses, names, etc. Since all phone numbers are unique, they are used as the primary keys and will be the main link between all the tables in the database.

In its current state, the database is populated with dummy data that we came up with, for demonstration purposes. The highlight of the application, alongside the attractive user interface and ease of use, is the added functionality that, users are able to add new entries to the database as they wish, effectively provided more contact information available to the subsequent users. If an address URL is specified, the entry can also be located on the map for easily understanding its address. With a sufficient amount of time and resources, the scope of this application can be broadened to contain different types of organizations, provide user authentication facilities and also automated real-time updates to the database.

The Entity Relationship Diagram



The database contains the following 6 tables, as illustrated above

1) The Main Table:

This table contains information that is common to all the other tables in the database and serves as a “hub” through which all other data can be linked and accessed. It contains information about the phone number, category, address and GPS URL for each entry in the database.

2) Hospital Table:

This table is linked to the main table and also contains information that is specific to hospitals in general, such the name of the hospital and specialization information which tells us whether the particular hospital has various domains of expertise in the fields of Cardiology, Nephrology, Neuroscience and Orthopedics.

3) Hotel Table:

Similar to the hospital table, the Hotel Table is linked to the main table and contains information specific to hotels, such as the name of the hotel, whether the hotel serves non-veg or not and if the hotel provides home delivery services.

4) Shops Table:

The shops table is also linked to the Main table and contains information specific to Shops such as the category of the shop (electronics, jewellery, grocery etc.) and the name of the shop.

5) Travel Agency:

This table contains information regarding various travel agencies, such as whether the travel agency provides for communication via cars, buses, trains, planes or combinations of the aforementioned or if they offer international packages. This table is linked to the Main table and also contains an attribute that describes the Travel Agency Name

6) Individual

The last table in the database is the Individual database, which contains information regarding various independent users who have signed up to be represented in the Yellow Pixels. Individual data includes their First name, Last name and phone numbers.

The **Works For** relation exists to link the various individuals to the type of occupation or in other words, their “Field of Work”. This is necessary because, with this, a person using the application will be able to quickly list out the contact details of all people working under a certain category. Hence it relates the individual table with the other tables in the database.

Implementation

This project, in its entirety, was built using the following software:

- 1) **MySQL Workbench:** used to design the database and create and populate the tables with some initial data
- 2) **IntelliJ IDEA:** used to design the front end GUI in java, using java Swings.
- 3) **Lucid Chart :** Used to design the ER diagram (previous section)

There are multiple different GUI forms present within the project directory and each one serves a different purpose within the application. There are 5 forms that represent the part of the application that makes it possible to query the database in order to find out contact information regarding the entries. Five more forms exist that allow entering new data into the database. The entire UI for the app has been coloured yellow as a tribute to the Yellow Pages

When you open the application, you are welcomed by a friendly yellow widget where you choose your initial requirement from the available category options after the selection of which, on hitting the “GO” button, a new window loads depending on the category chosen. In each of these windows, you are given multiple options to filter the database and fetch those tuples that satisfy your search criteria. Each of these windows also has a button that enables you to locate the addresses on a google map, making navigation easier.

Within each of the category-specific windows, is a button that allows you to easily insert new data into the corresponding table of the database. There is one unique insert window for each category and within the Java class of each of these insertion windows, is the code meat for handling updates specific to that table. Hence, errors in one particular window do not affect the overall execution of the app.

All the heavy lifting is done using JDBC, using the java.sql module which allows easy manipulation of databases in Java. The user interface is designed using the Swings module of Java. We would also like to mention the extensive use of GitHub that was involved in the making of the project, without the version control capabilities of which we wouldn't be able to pull off this application.

The following MySQL statements were used to create the database

1) Main Table

```
CREATE TABLE Main_table(  
  phone_num bigint,  
  category varchar(20),  
  street varchar(20),  
  city varchar(20),  
  url varchar(350),  
  PRIMARY KEY(phone_num)  
);
```

2) Travel Agency

```
CREATE TABLE Travel_agency(  
  phone_num bigint PRIMARY KEY,  
  name varchar(20),  
  has_bus boolean,  
  has_train boolean,  
  has_flight boolean,  
  has_car boolean,  
  has_international boolean,  
  FOREIGN KEY(phone_num) REFERENCES Main_table(phone_num)  
  ON DELETE CASCADE ON UPDATE CASCADE  
);
```

3) Hospital

```
CREATE TABLE Hospital(  
  phone_num bigint PRIMARY KEY,  
  name varchar(20),  
  has_neuro boolean,  
  has_ortho boolean,  
  has_nephro boolean,  
  has_cardio boolean,  
  FOREIGN KEY(phone_num) REFERENCES Main_table(phone_num)  
  ON DELETE CASCADE ON UPDATE CASCADE);
```

4) Hotel

```
CREATE TABLE Hotel(  
phone_num bigint PRIMARY KEY,  
name varchar(20),  
has_nonveg boolean,  
has_homedelivery boolean,  
FOREIGN KEY(phone_num) REFERENCES Main_table(phone_num)  
ON DELETE CASCADE ON UPDATE CASCADE  
);
```

5) Shops

```
CREATE TABLE Shops(  
phone_num bigint PRIMARY KEY,  
name varchar(20),  
shop_type varchar(20),  
FOREIGN KEY(phone_num) REFERENCES Main_table(phone_num)  
ON DELETE CASCADE ON UPDATE CASCADE,  
CHECK (shop_type in  
( 'Jewellery', 'Electronics', 'Grocery', 'Florist', 'Mall' ))  
);
```

6) Individual

```
CREATE TABLE Individual(  
phone_num bigint PRIMARY KEY,  
FName varchar(20),  
LName varchar(20),  
FOREIGN KEY(phone_num) REFERENCES Main_table(phone_num)  
ON DELETE CASCADE ON UPDATE CASCADE  
);
```

Screenshots

Conclusion

Although the all the code works and the application itself runs reasonably well, it must be said that there is scope for significant improvements in terms of design, code and database design quality. Although it started off with merely 5 categories, the code is structured in a way that makes it possible to expand on it and add further categories easily. In the future, we aim to add functionalities that allow the users themselves too create new categories and also the ability to delete outdated entries. From the conception of the idea to the writing of this report, this project has been an absolute pleasure to work on and we come out saying that we've learned quite a bit!

References
