CS6360 - Project 1 - Design Document

Submitted By:
Bhushan Vasisht - bsv180000

October 11, 2020

Introduction

This project is a Full Stack application for maintaining a dynamic contacts list application.The application consists of a web UI develeped with Reactjs, a backend REST service provider with Nodejs and uses MySQL database for data store.

The application supports all the generic CRUD operations that are listed in the Functional Requirements and and more for application maintenance. Some of the notable features are:

- Viewing all the contact details in tabular form on the home page.
- Adding new records/updating old records/deleting records directly from the UI.
- Custom searching from a single google like search box

System Architecture and SQL Schema

1. Database Tables

- CONTACT

Field	Type
Contact_id	NUMBER
Fname	VARCHAR(30)
Mname	VARCHAR(20)
Lname	VARCHAR(30)

- ADDRESS

Field	Type
Address_id	NUMBER
Address_type	VARCHAR(10)
Address	VARCHAR(512)
City	VARCHAR(30)
State	VARCHAR(30)
Zip	NUMBER(5)

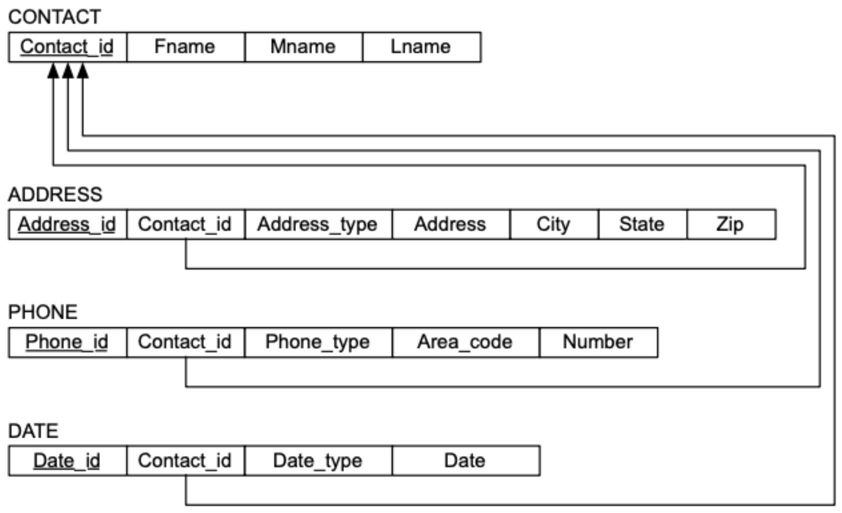
- PHONE

Field	Type
Phone_id	NUMBER
Phone_type	VARCHAR(10)
Area_code	NUMBER(3)
Number	NUMBER(7)

- DATE

Field	NUMBER
Date_type	VARCHAR(10)
Date	DATE

2. Inter-table Relationship



Design Decisions and Assumptions

To be able to understand the design decisions, we should take a look at the services offered by the application.
The end user services seen from the frontend include:

- Display Contacts
  - The base design is in the form of a table. The headers have the name of the field.
  - Each row has a specific entity's contact details. It was decided to save space and display related fields together.
  - Special purpose buttons are placed in each row to facilitate modify/delete the entry
  - A loading animation will be rendered until the data is processed and rendered.
- Add/Modify/Delete Contacts
  - The base design is in the form of a HTML form element.
  - Simple field names are taken as text inputs.
  - The application supports adding variable number of address, phone and date components. A special button is placed which allows to add more such components.
  - Composite fields are rendered as table. Each simple field is taken as text input
  - Each composite field can be deleted by clicking the corresponding delete button
- Search Contacts
  - Search services are provided over a single text search like google.
  - Users can search on any parts of contact, addresses, or phone numbers components;
  - The hits are displayed in the form of a table. Here, only the full names of the entities are displayed.
  - Buttons are placed to modify or delete an entity.
  - A clear button is placed for clearing the search results.
  - Users can modify the text and search for different results.

Backend Services

- Get every contact detail
  - Method: GET
  - Route: /all
  - Return: [JSON]
- Get a specific contact detail
  - Method: GET
  - Route: /contact
  - Query Parameter: contact_id
  - Return: JSON
- Get search results
  - Method: GET
  - Route: /search
  - Query Parameter: key
  - Return: [JSON]
- Add/Modify a contact
  - Method: POST
  - Route: /addNew
  - Body: JSON
- Delete a contact
  - Method: PUT
  - Route: /delete
  - Query Parameter: contact_id