|  |  |  |  |
| --- | --- | --- | --- |
|  | Gonzales Reyna Co.  Science & Technology Building, Preston Smith Rd, Odessa, TX 79762  Phone (432) 894 - 4726 e-mail cesar.a.reyna@gmail.com  Phone (432) 741 – 0225 e-mail gonzales\_c52743@utpb.edu |  |  |
| ADDRESS BOOK Management | | | | |
| Title | | | | |
|  | | | | |

|  |  |  |
| --- | --- | --- |
| Report Date | Project Name | Prepared By |
| 11/27/2023 | Address Book Management System | Reyna C. and Gonzales C. |

|  |
| --- |
| Overview |
|  |

The Address Book Management System is a desktop application developed in Java that allows users to create, search, update, and delete contact information from an SQLite database. The application features a graphical user interface (GUI) for easy interaction.

|  |
| --- |
| Objective |
| To develop a user-friendly application that efficiently manages contact information using database operations, and to demonstrate the application of object-oriented programming and database connectivity in Java. |
| Relation To Class |
| This project encompasses several key concepts discussed in the course, such as:   * Software Development Life Cycle (SDLC): The project went through the planning, design, implementation, and testing phases. * Object-Oriented Modeling: The system was designed using classes and objects, encapsulating data and functions related to address book entries. * Database Management: The application integrates with an SQLite database, demonstrating Update, Delete, Insert operations in a relational database. * Relation Model: Our database was a relational database, and with that being said it allowed us to perform actions such as cascade delete, being able to pull from 4 tables at once, and inserting in a table with a corresponding primary key. The benefit of implementing a relational database in our application is scalability, tracking the entries, and an ease of processing the action we are calling upon the application. |
| Main Concepts |
| Classes and Methods:   * DBconnection class handles database connectivity and operations like insert, search, delete, and update. * GUI class creates the user interface and responds to user actions. * AddressBook class contains the main method to run the application. * Database Operations: Implemented using SQL queries in Java, facilitated by JDBC (Java Database Connectivity). * Exception Handling: The application includes robust error handling for database and input/output operations. |
| |  | | --- | | Difficulties encountered | | * Handling SQL Exceptions and ensuring database integrity during database operations. * Version control * Lack of planning led to poor code readability. | | Pros/Cons | | * Pros: Simplifies relational database management for the user, intuitive gui, error handling.   + Learning Github and being able to transfer the code between the two of us. * Cons: limited to a desktop environment, lacks network capability for remote access, some potential for SQL injection. | | FUTURE Enhancements | | * Implement network functionality for remote database access. * Implement prepared statements on all database operations to prevent SQL injection. * Add features like sorting and advanced search criteria | | CONCLUSIONS/RECOMMENDATIONS | | The Address Book Management System is a successful demonstration of applying object-oriented principles, database management, and relational database skills to create a functional desktop application. It simplifies relational database operations for the end user through a simple GUI. It effectively demonstrates the utility of the knowledge acquired in class in a practical and usable business system. | |