C868 – Software Capstone Project Summary

Task 2

Capstone Proposal Project Name: Scheduling Application

Student Name: Kenneth Gillingham

Table of Contents

Contents

Section A

A1: Buisness Problem

Customer

KennyG.Tech is a startup talent acquisition agency. They manage multiple contacts and customers in a centralized database. They are spread across the United States and France and are in multiple different time zones. They are looking to expand their buisness and generate more revenue for their upcoming IPO on the US Stock Market. They want to have everything in place before going public to ensure a smooth transition, keep returning customers, and help entice new customers to try their products.

Problem

Currently KennyG.Tech uses a combination of Google Suite software's and various other free applications to manage all their contacts, customers, and appointments. They need a centralized software that is connected to a centralized database to store all the needed information. Currently with their tech stack, appointments can easily get lost, deleted, or modified without the creator knowing. Having a software that is connected to a database that includes a user login functionality would help keep all appointments, contacts, and customers information safe and readily available for any employee without disrupting another employee's progress.

Solution

KennyG.Tech needs a solution for their appointment scheduling that includes the ability to link appointments to respective contacts and customers. The software solution needs to be able to adjust to the user's time zone automatically and change languages based on the user's location. The software must be able to have user login capabilities and to alert the user upon logging in if there is an appointment within 15 minutes or not. KennyG.Tech needs the ability to generate different reports for different use cases as well, this includes a report of appointments per month, appointments per contact, and group by appointment types. This software application needs to have a clean and easy to use Graphical User Interface (GUI) for which the employees will interact with and control the application. The application needs to be linked to a database that will house all the information and the application can pull from, add to, delete from, or modify from the database.

A2: Existing Gaps

KennyG.Tech is currently using a combination of Google Suite applications and various other free applications. This creates an issue of employees not being able to work in the same software at the same time and have the data be synced. The software solution allows employees a centralized software to handle all appointment scheduling and tracking of customer and contact information. This software is connected to a database so users can not change or delete appointments while another employee is altering. This software application aims to make handling all needed information easier for the customer.

A3: Software Development Life Cycle

The chose Software Development Life Cycle for this project is the Waterfall Method. This method is most familiar to the developers and therefore would not require any major changes to the development team. This method follows a few major phases with sub steps within each phase.

The first main phase is the Requirements Phase. In this phase the development team meets with the customer to determine the main requirements for the software solution. All requirements should be clearly stated and understood by both teams before moving forward. If requirements are found later in the life cycle, there is a chance they may not be met until the next major software update.

The second main phase is the Design Phase. In this phase the development team begins to make wireframes of the GUI, the Entity Relationship Diagrams (ERDs), and all other rough drafts needed for the software.

The third main phase is the Implementation Phase. Feedback from the previous phase will help guide the development team in creating the software. They will use all rough drafts from the Design Phase in their creation along with any feedback from the customer. During this phase Unit tests are performed whenever a major part of the software application is built to ensure as little bugs as possible at the end. It is easier to test each component as they are created rather than waiting until the end to test everything.

The fourth main phase is the Verification Phase. This is the phase where the software is tested against the requirements and verified by the customer that all needs are met. In

coordination with the customer, more test plans are created in this step to further test the application to clear as many bugs as possible.

The fifth main phase is the Development Phase. The development team will create the application to be used in the desired environment as stated by the customer. The development team will ensure the software can be ran within the desired environment. A development timeline is also created to give everyone dates for when specific components will be built and when the final release will occur.

The last main phase is the Maintenance Phase. This phase includes software updates as set intervals to continuously clear bugs and ensure the software runs as stated. Improvements by the customer can also be added to a list and added to the software using a priority system within the set software update intervals.

A4: Deliverables

Along the way there will be a few main deliverables from the team. This includes:

- Project Timeline
    - A timeline for the development lifecycle with dates
- Test Plans
    - A set number of test items to ensure as little bugs as possible
- Wireframes
    - Wireframes of the GUI to verify with customer of layouts
- Rough Drafts
    - Rough Drafts of the GUI to approve with the customer
- Final Software Application
    - Final software that meets all requirements

A5: Implementation Plan

Implementation will follow the waterfall methodology explained above. The development team will work to complete each phase in order and ensure quality work is done along the way. This will require collaboration with the customer and constant communication to ensure no requirements get missed and everything is up to the customer's standard. The development team needs a clear plan and path moving forward to ensure the software is developed correctly and on schedule.

A6: Validation and Verification

Along the implementation and creation lifecycle there will be tests performed to ensure quality and customer satisfaction. The initial wireframes and rough drafts will be shown to the customer and any feedback will be changed to them. During development unit tests will be used to test specific features and units during the creation. After the software is developed, more tests will be used to test the final release and ensure all requirements are met and the customer is satisfied with the software.

A7: Programming Environments

Programming Environments

The environments that will be used includes:

- Windows 10 or MacOS Ventura
- Java 17
- MySQL Version 8

Environment Costs

KennyG.Tech currently uses a variety of free applications. Due to this, there may be some initial costs at the beginning to buy some software licenses to run the software application. The database will be hosted on a physical Windows PC so no cloud-based subscriptions will be needed. Based on the company's current technology situation, a new computer may be required to host and run the database software.

A8: Project Timeline

Phase	Milestone/Task	Deliverable	Description	Dates
Pre-development	Gather all Requirements	Requirements	Meet with Customer to create clear and accurate list of Requirements	01/01/2023 – 01/08/2023
Design	Outline of Database	ERD	Create the Database outline and schema	01/15/2023 – 01/22/2023
Design	Wireframes of the GUI elements	Wireframes	Create basic wireframes for GUI elements	02/01/2023 – 02/15/2023
Design	GUI rough Drafts	GUI rough Drafts	Create basic GUI rough drafts for customer approval	02/16/2023 – 03/01/2023
Implementation	Software Rough Draft	Software Rough Draft	Create rough draft of software application.	03/01/2023 – 04/01/2023
Verification	Unit testing of software rough drafts	Completed unit testing results	Begin unit testing on software application based on test plans	04/01/2023 – 04/15/2023
Development	Development of Software in Customer environment	Bug report and feedback from Customer	Begin developing software application for the Customer's environment	04/16/2023 – 04/30/2023
Maintenance	Customer feedback and final testing	Final bug report and customer feedback	Hands on feedback from the customer for last minute bug and feature adjustments	05/01/2023 – 05/22/2023
Maintenance	Software updates	Major Software updates on agreed upon intervals	Continuous software updates as agreed upon with the Customer. Future improvements will be tied with the updates	06/01/2023 – End of Software Life

Section C

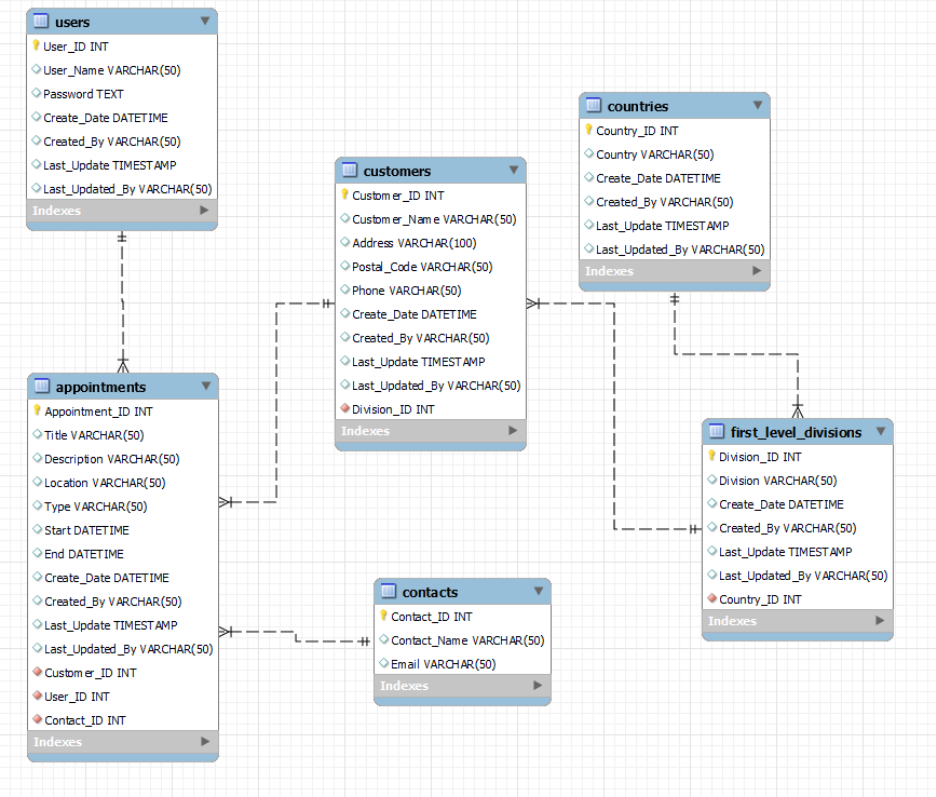
C1: Class and Design Diagrams

Entity Relationship Diagram (ERD)

Below is the ERD for the database that will be used in conjunction with the software application. This diagram shows the connections and relationships between the different tables within the database schema. Along with this database schema, there are multiple classes within the source code that handles the interactions within the application.

The JDBC class handles the code needed to connect to the database. There will be a copy of the script within a .txt file in case it is needed in the setup process. This is referenced whenever a connection to the database is needed.

There are multiple database classes that handle the Create, Read, Update, and Delete aspects for the respective table within MySQL. Each table has its own "DAO" class within the Database package in the source code. The main classes are, Appointments, Contacts, Country, Customer, firstLevelDivision, and Users. There is also a class called Search that handles all of the code related to the search fields for both the appointments table and the customer's table.

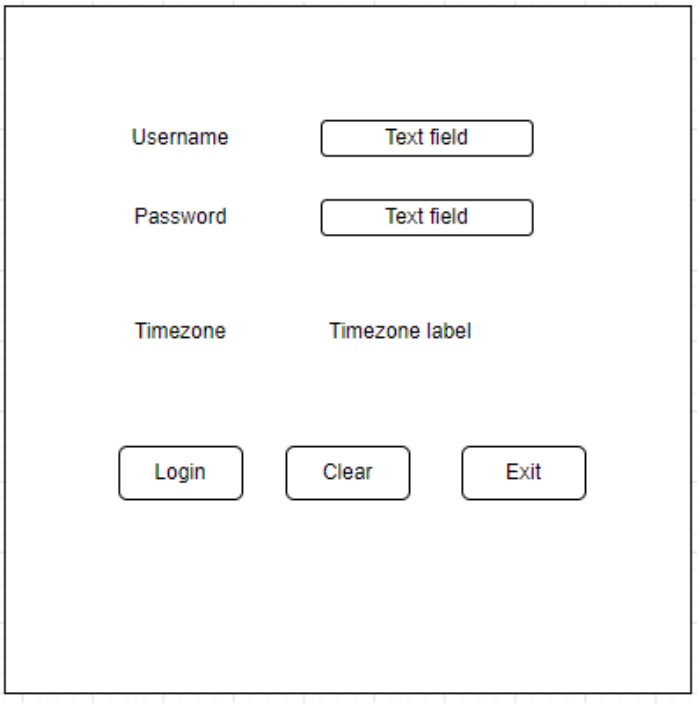


Graphical User Interface Wireframes and Rough Drafts

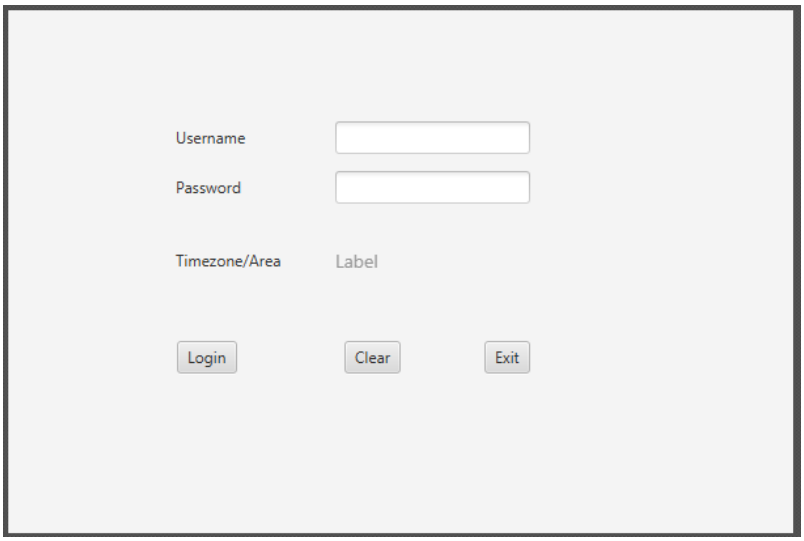
Below are the wireframes and rough drafts of the GUI screens.

Login Screen:

Wireframe:

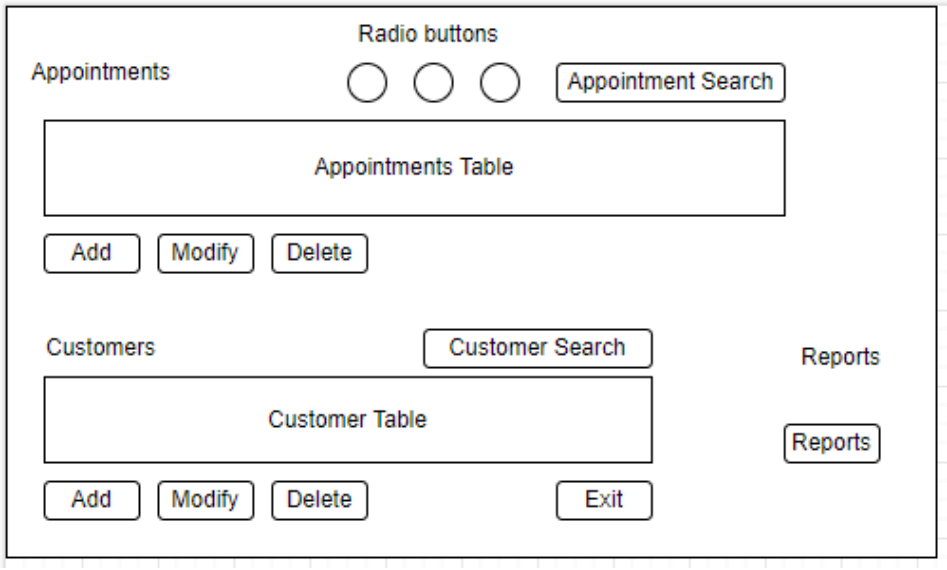


GUI:

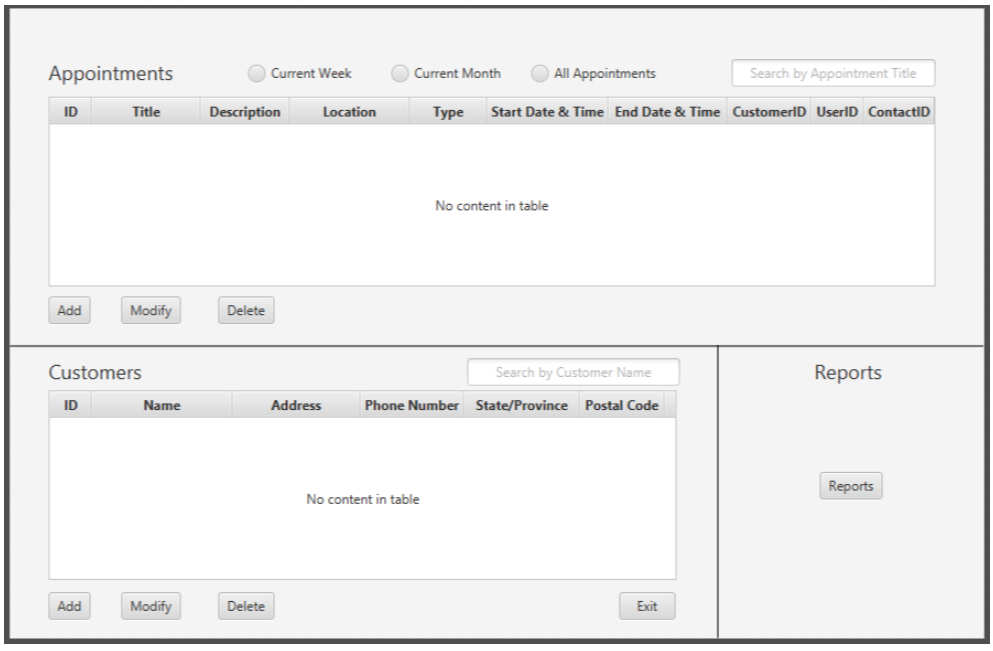


Main Screen:

Wireframe:

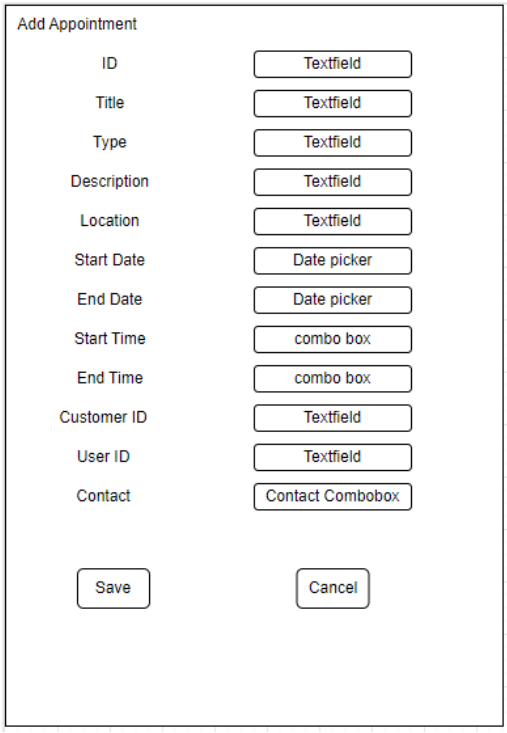


GUI:

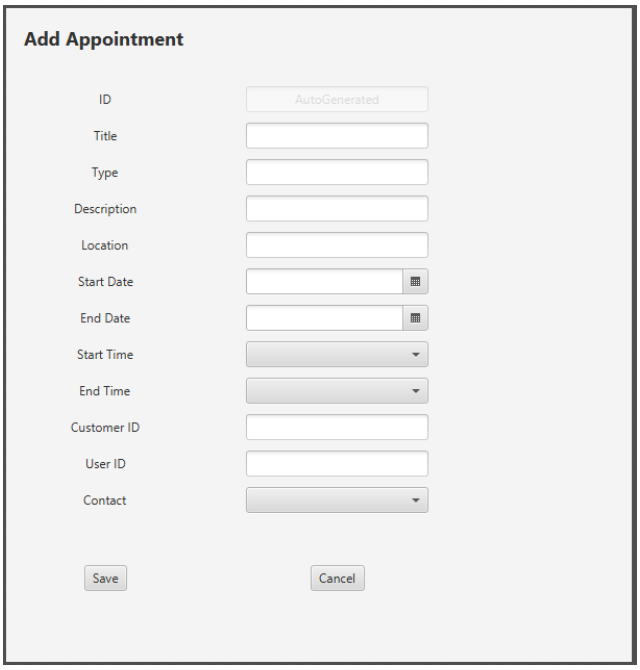


Add Appointments:

Wireframe:

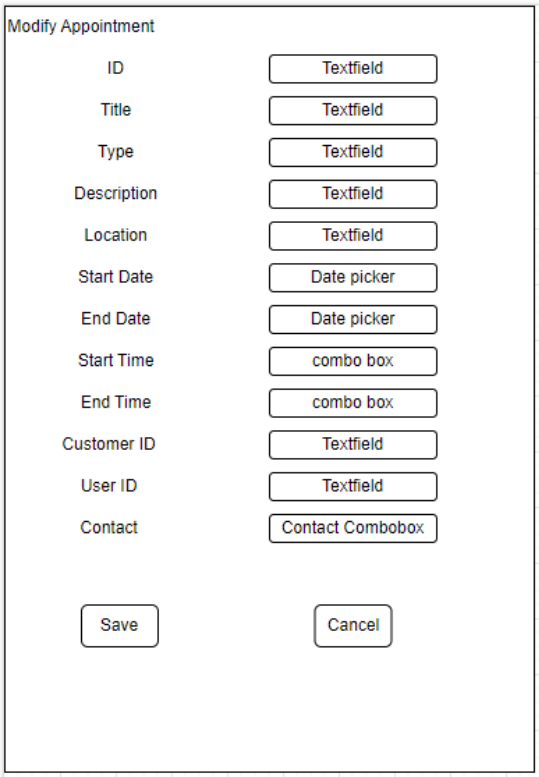


GUI:

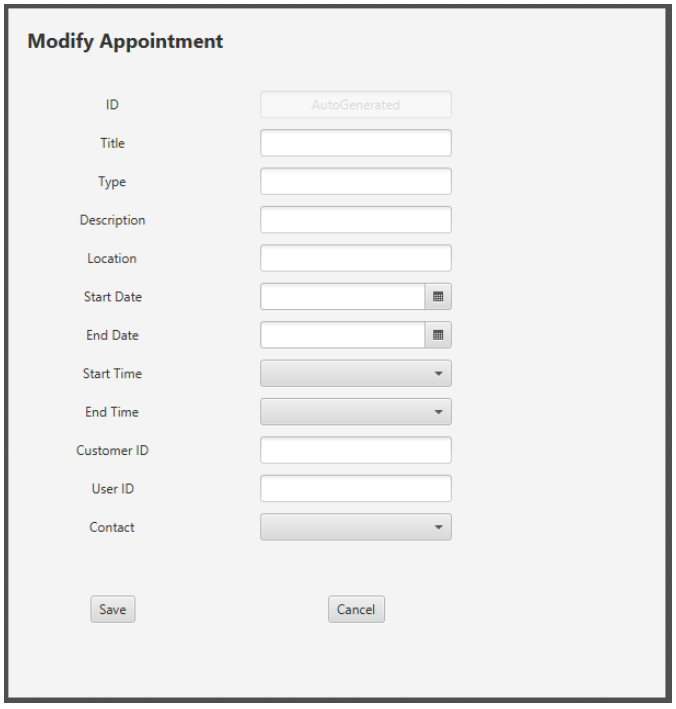


Modify Appointments:

Wireframe:

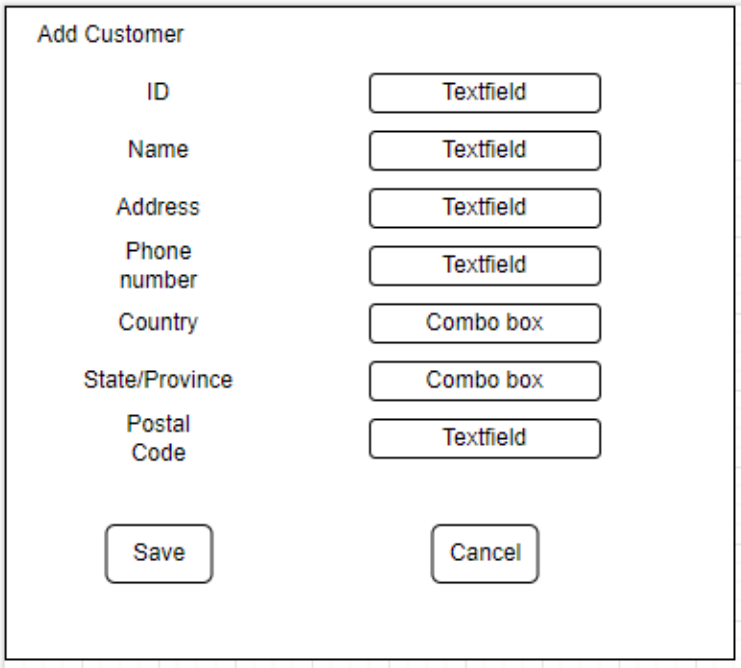


GUI:

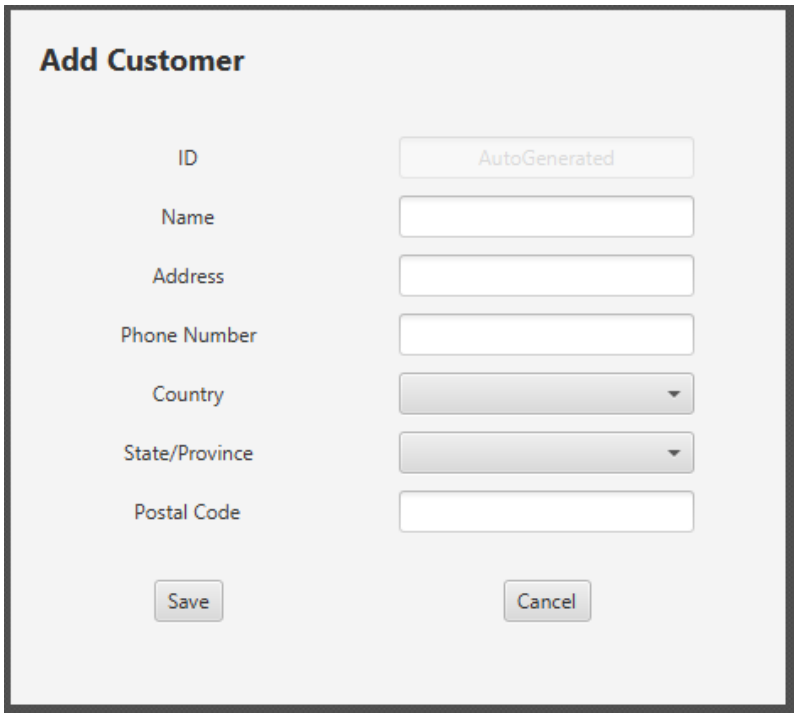


Add Customer:

Wireframe:

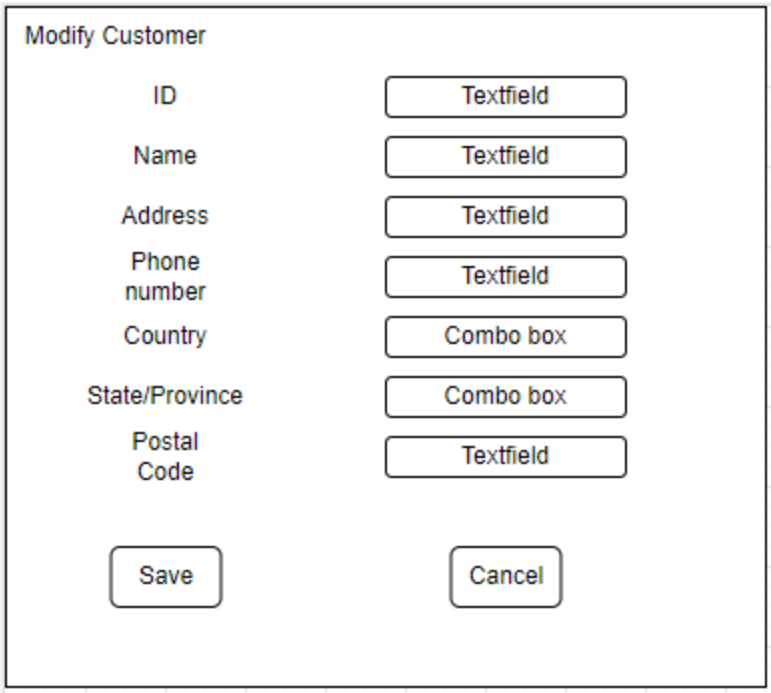


GUI:

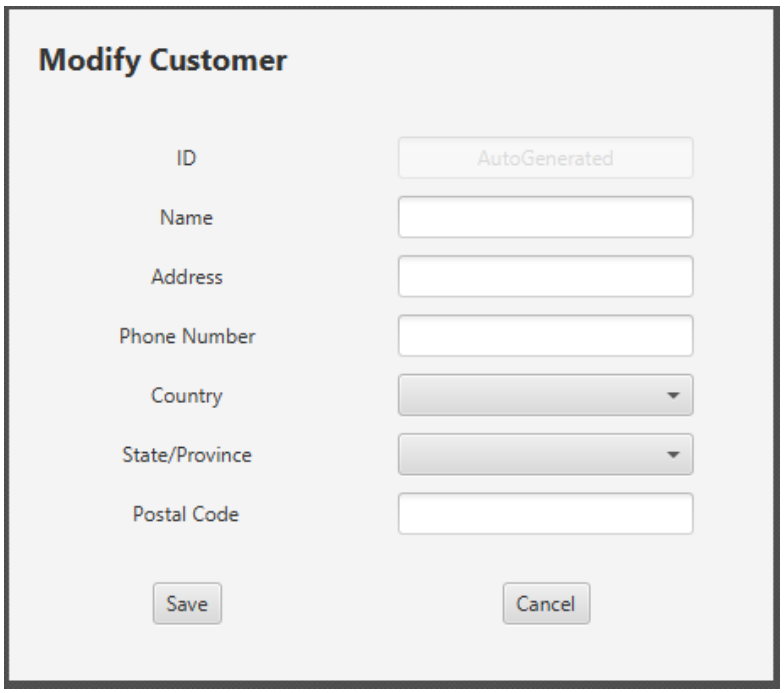


Modify Customer:

Wireframe:

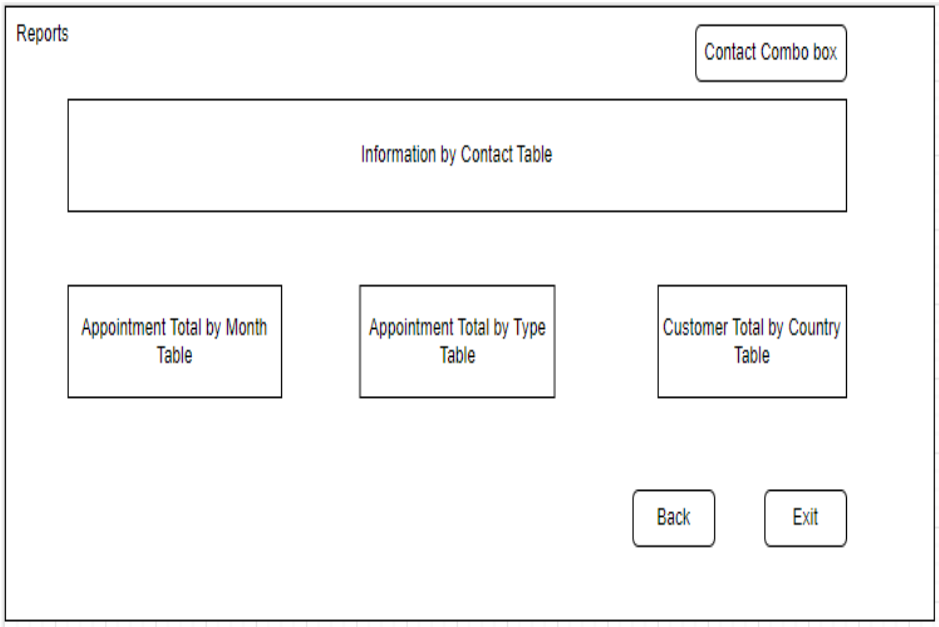


GUI:

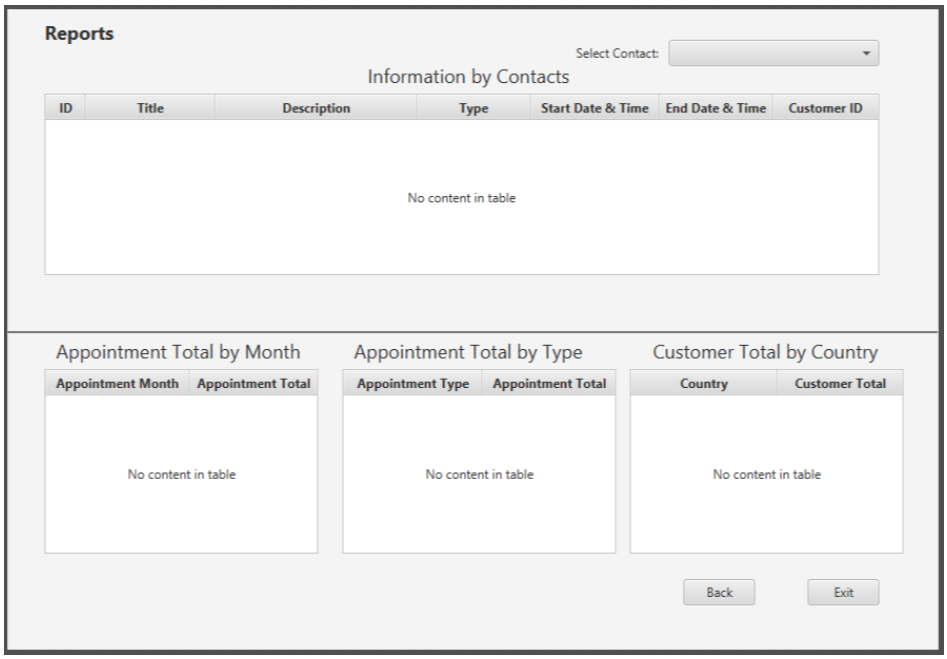


Reports:

Wireframe:



GUI:



C2: Test plan

The test plan below captures a main unit test from each of the main screens or sections. In addition to the tests planned below, the development team will also test each addition to ensure as few bugs as possible. Each button, radio button, and GUI element will be tested during development to ensure proper functionality and minimize the number of bugs found later in the development life cycle.

Test Case	Description	Test Steps	Expected Results	Actual Results	Pass or Fail
#1	Correctly login with appropriate credentials	1: Enter username “test” 2: Enter password “test” 3: Click the “Login” button	Successful Login	Successful Login	Pass
#2	Appointment Month Filter	1: Select the Appointment Month Radio Button	Appointment table displays all appointments within the current month	Appointment table displays all appointments within the current month	Pass
#3	Customer Search Field	1: Select the customer search field and type a section of one of the customer’s names	The table should display only the customer with the section of name typed in the field	Only the customer with matching name appears	Pass
#4	Reports Button	1: Select the “Reports” button	The application should switch to the “Reports” screen	The application shows the “Reports” screen	Pass
#5	Exit Button	1: Select the “Exit” button from the main screen	The application should display a confirmation message and then close the application upon confirming	The application shows a confirmation message and closes upon confirmation	Pass

C3: Test Results

Test results for the above testing can be seen in the table above. All manual unit tests passed, as well as each functional element passed during the development testing. At the end the completed software was put through extensive testing to ensure all elements worked together cohesively and all requirements were met, and customer satisfaction was achieved.

C4: Source Code

Source code for the application is provided within the Zip folder. Instructions on how to open and run the source code is provided in the following guide in “C5”.

C5: User Guide for Setting Up Applications

Introduction:

The following guide will show the proper steps to download, extract, and run the software application. There will also be steps on how to ensure the MySQL workbench is properly set up and the IntelliJ IDEA is properly set up as well. All the needed code will be inside the .zip folder and can be copy/pasted when it is needed. This guide follows all the needed steps that were needed for the Software 2 project. The following guide will address the use of the application more than the initial set up.

Prerequisites:

Ensure that IntelliJ IDEA 2022.3.1 (Community Edition) and MySQL Workbench 8.0 are both installed on the device. Also ensure that the computer is running either Windows 10 or MacOS Ventura. Verify that Java 17 SDK is also installed on the computer so it can properly compile and run the code.

Installation:

1. Download the .zip folder that contains all the necessary information.
2. Extract the .zip folder to the location of your choosing.
    a. This can be done by right-clicking on the .zip folder and selecting the "extract all" option.
    b. A popup will appear that allows you to select the location where the folder will be extracted to.
3. Once the folder is extracted there are a couple of different ways to open the application.
    a. You can right-click on the "capstone" folder and choose the "open folder as IntelliJ IDEA Community Edition Project". This will open IntelliJ and populate with all the data.
    b. You can open IntelliJ, select file, open, and then navigate to the capstone folder and open it that way.
    c. Or you can open the capstone folder, open the "src" folder, open the "main" folder, open the "java" folder, open the "Gillingham" folder, open

the "capstone" folder, this will give you access to the Java files for the source code.

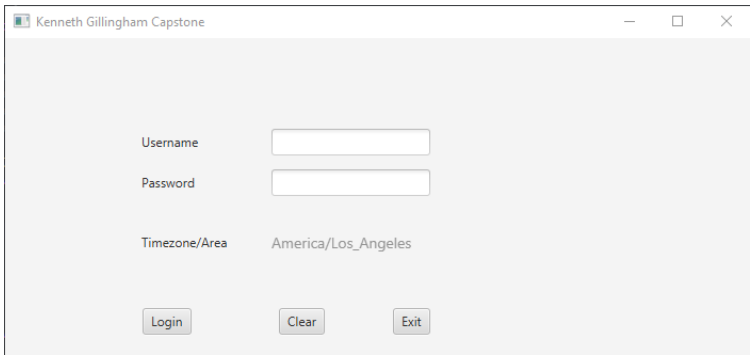
4. Once the project is opened in IntelliJ, there are a few things that need to be verified.
    a. In IntelliJ, go to file -> Project Structure -> Project. The SDK should be set to openjdk -19 or similar. The Language level should be set to "16 – Records, patterns, local enums, and interfaces". If these two don't match, the project may not run.
    b. In IntelliJ, go to file -> Project Structure -> Modules. The "module sdk" should be set to openjdk -19. If this does not match the project may not run.
    c. Ensure the mysql-connector-j-8.0.32 module is installed. This can be installed using the same instructions from Software 2. This can also be downloaded from the internet if needed.
5. Once everything is verified within IntelliJ, we can move on to verifying everything with MySQL workbench.
    a. If needed, you can download MySQL workbench from the internet and install version 8.0 to match this project.
    b. Follow the same steps as Software 2 when creating the database schema.
    c. There is a .txt file for the DDL script if needed.
    d. There is a .txt file with the script to populate all the tables needed for the database.
6. Once IntelliJ and MySQL workbench are all verified, open the project within IntelliJ.
7. Navigate the project tree to the Main.java class.
8. In the top of IntelliJ software, click on the "hammer" icon to build the project.
9. In the top of IntelliJ software, click on the "play button" icon to run the application.
10. This should open the Login screen to the application. If there are any errors, please refer and make sure all needed adjustments are made and everything else

matches. If needed, refer to the Software 2 class setup and ensure everything matches.

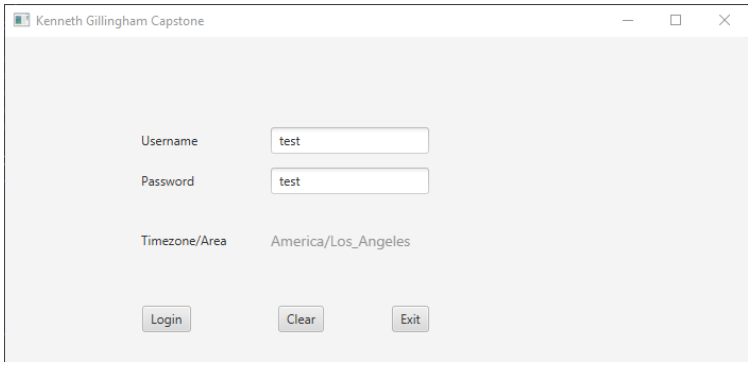
C6: User Guide for Running Application

Login Screen

1. Upon finishing the above installation guide and running the Main.java file, you should be prompted with the Login Screen of the application.



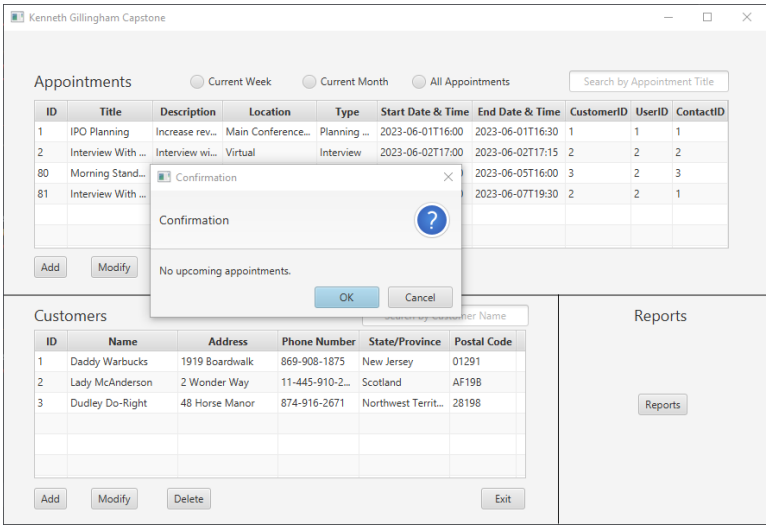
2. On this screen, please enter a username of “test” and a password of “test”. This will allow you to successfully login and view the main screen.



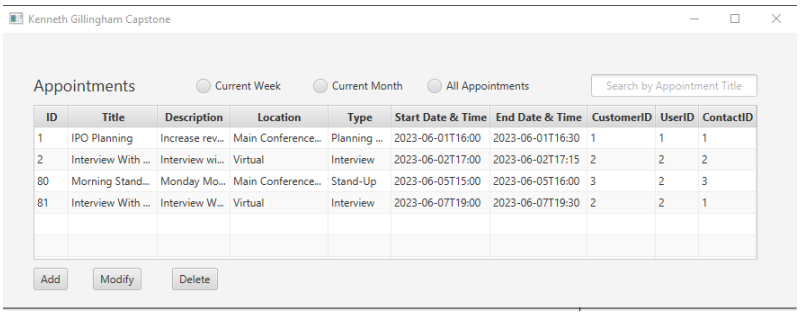
3. Other features on this screen are the time zone and area label that will update automatically based on your computer’s settings. All labels will change to French if the computers language is set to French. The “clear” button will clear both username and password fields. The “exit” button will display a popup confirming to exit the application, if that popup is confirmed, the application will close, if the popup is not confirmed it will remain on this screen.

Main Screen

1. Upon successful login, the main screen will be displayed and the login activity .txt file will be updated with the username and a timestamp of when that user logged in. Also, upon logging in, a popup will alert the user if there are appointments within 15 minutes of their current time or not.



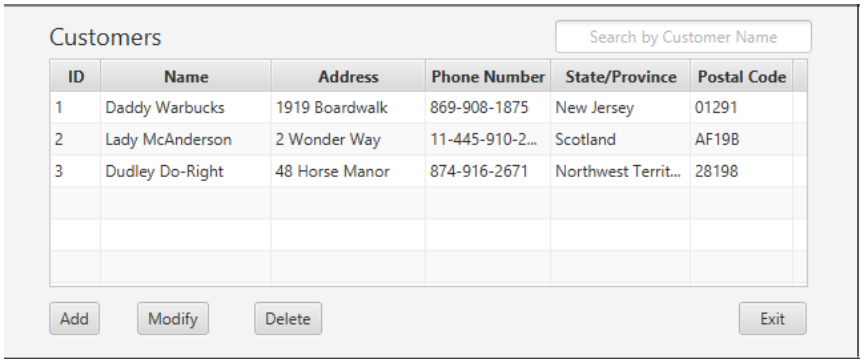
2. On this screen there is a lot of information. First, we are going to focus on the Appointments section. There is a main table that displays all the Appointments information. There are also 3 radio buttons that can be used to filter the table based on the labels of the radio buttons. For example, if you select the “current week” radio button, only appointments in the current week will be displayed. There is also a search field that can be used to search by appointment title.



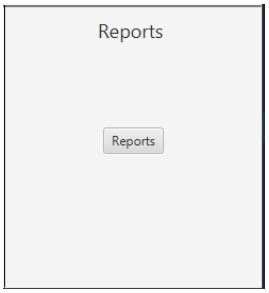
3. Pressing the “Add” button will open the “Add appointments screen”.
4. Pressing the “Modify” button will open the “Modify appointments screen”.
5. Pressing the “Delete” button will open a popup confirming if the user wants to delete the selected appointment. If the popup is confirmed, the appointment will

be deleted, if the popup isn't confirmed, then the popup will go away, and the appointment will not be deleted.

6. Secondly there is the Customer Section, there is a main table that displays all the needed information for all the customers. There is a search field that can be used to search by Customer name and will display only customers that match what is typed into the search field.

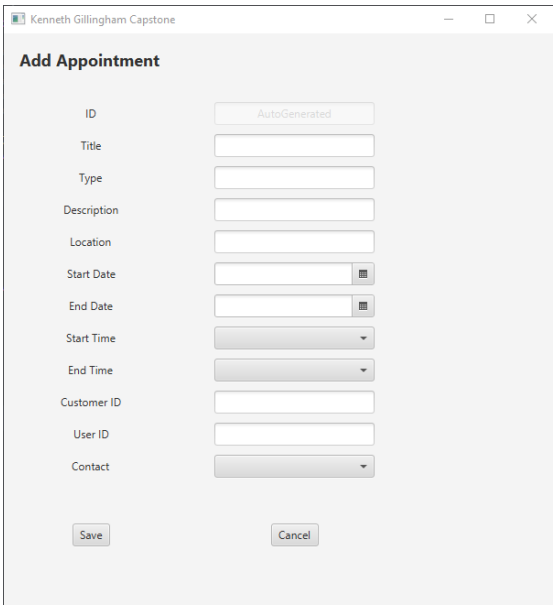


7. Pressing the "Add" button will open the "Add Customer Screen".
8. Pressing the "Modify" button will open the "Modify Customer Screen".
9. Pressing the "Delete" button will open a popup confirming if the user wants to delete the selected customer. If the popup is confirmed, the selected customer will be deleted. If the popup is not confirmed, the popup will go away, and the customer will not be deleted.
10. Pressing the "Exit" button will open a popup confirming if the user wants to close the application. If the popup is confirmed, the application will close. If the popup is not confirmed, the popup will go away, and the application will not close.
11. Thirdly, there is the "Reports" section of the Main Screen. There is simply a button that when pressed will take you to the "Reports" screen.



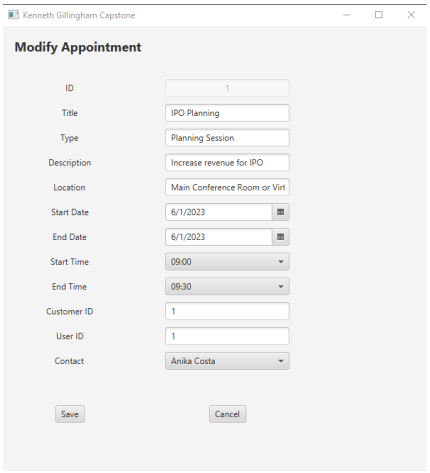
Add/Modify Appointments

1. Pressing the “Add” button below the Appointments table will open the Add Appointments Screen.



2. The “ID” field is auto generated upon creation of adding new appointments.
3. The “Title”, “Type”, “Description”, “Location”, “Customer ID”, and “User ID” are text fields to be inputted by the user.
4. The “Start Date” and “End Date” use calendar selection tools to choose the appropriate dates needed.
5. The “Start Time” and “End Time” use combo boxes for the time using 15-minute intervals.
6. The “Contact” uses a combo box of all contacts that are populated within the database data.
7. Pressing the “Save” button will save all the information that the user inputs.
8. Pressing the “Cancel” button, will cancel out of this screen, and return the user to the Main Screen.
9. There are exceptions built in that will not allow the appointment to be saved if any of the fields are blank, if the “End Time” is before the “Start Time”, if the “End Date” is before the “Start Date”, or if the selected times are outside business hours.

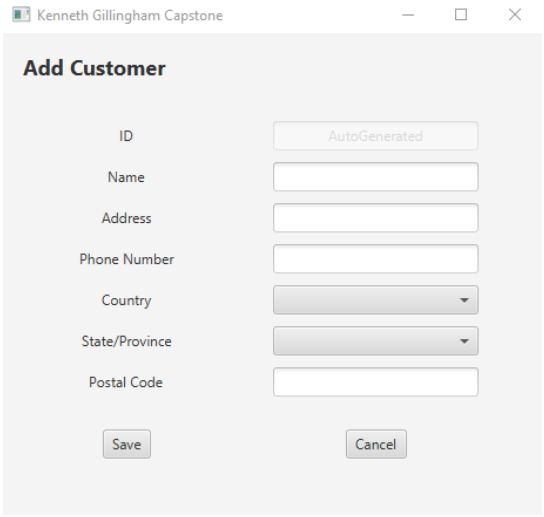
10. The user inputs the dates and times in their time zone, the data is then automatically changed to UTC when stored in the database, but it is then converted back to the users' time zone if the "Modify" button is pressed.
11. Pressing the "Modify" button below the Appointments table will open the Modify Appointments Screen.



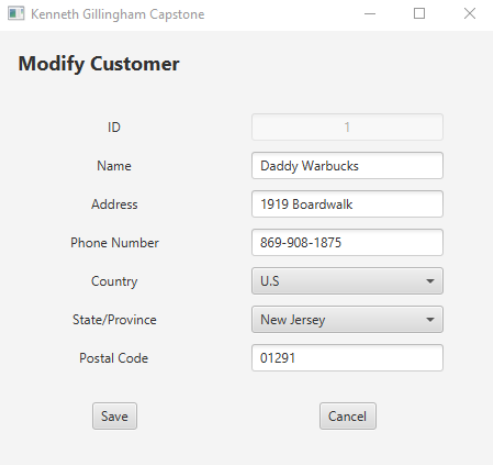
12. The data is all populated from the database, and everything is converted to the users' time zone.
13. Any changes can be made, and then pressing the "Save" button will modify the appointment in the database with the changes.
14. Pressing the "Cancel" button will return the user to the Main Screen.
15. The same exceptions apply from the Add Appointments Screen.

Add/Modify Customer

1. Pressing the “Add” button below the Customers table will open the Add Customer Screen.



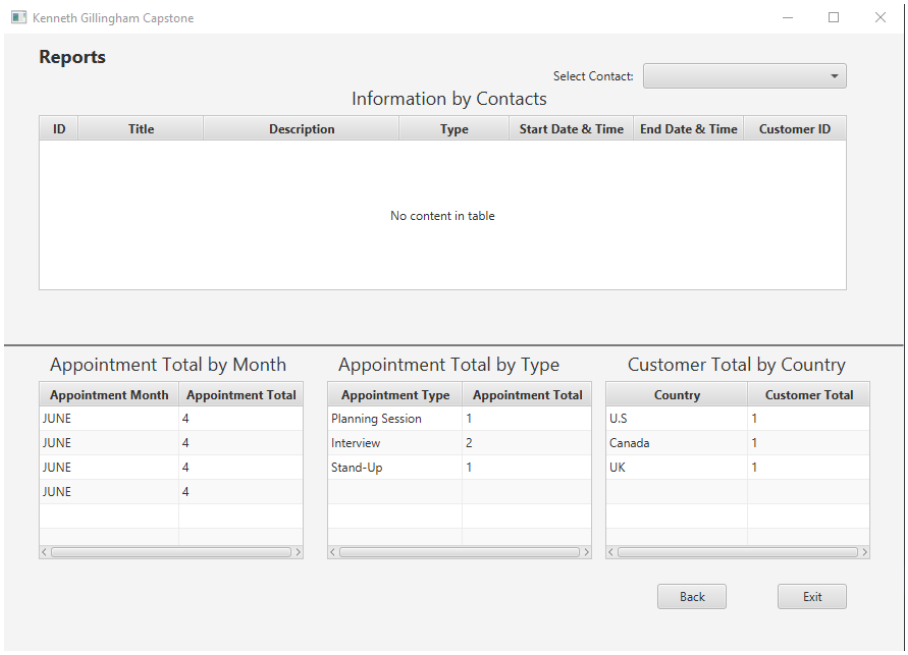
2. The “ID” field is auto generated upon creation of adding new customers.
3. The “Name”, “Address”, “Phone Number”, and “Postal Code” are text fields for the user to input the information.
4. The “Country” and “State/Province” are combo boxes with information populated within the database.
5. Pressing the “Save” button will save the customer information to the database.
6. Pressing the “Cancel” button will return the user to the Main Screen.
7. There are exceptions put in place that will not allow the user to save the information if any field is blank.
8. The “State/Province” combo box will change its selections based on the selection made in the “Country” combo box.
9. Pressing the “Modify” button below the Customers table will open the Modify Customer Screen.



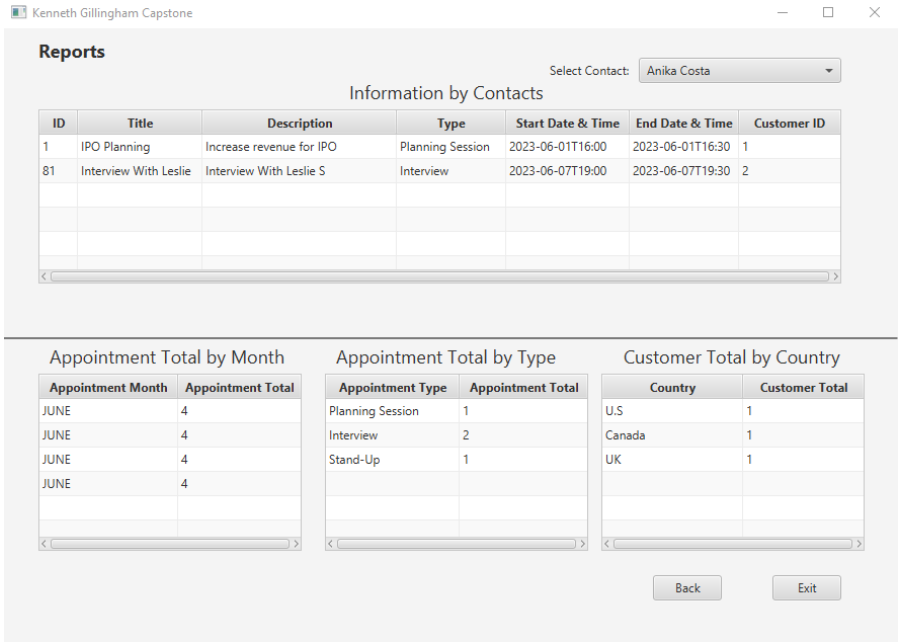
10. The data is populated from the information that is stored within the database.
11. Any changes can be made, and then pressing the “Save” button will modify the customer in the database with the changes.
12. Pressing the “Cancel” button will return the user to the Main Screen.
13. The same exceptions apply from the Add Customer Screen.

Reports

1. Pressing the “Reports” button on the Main Screen will open the Reports Screen.



2. At the top is the table that will populate data based on the selected Contact from the drop down at the upper right corner.



3. The bottom left table shows a summary of total appointments for each month.
4. The bottom middle table shows a summary of total appointments based on the type of appointment.

5. The bottom right table shows a summary of total customers by country.
6. Pressing the “Back” button will return the user to the Main Screen.
7. Pressing the “Exit” button will open a popup confirming if the user wants to close the application. If the popup is confirmed, the application will close. If the popup is not confirmed, the popup will go away, and the application will not close.