JASMINE LANGFORD

THE CAKE SHOP

APPOINTMENT BOOKING SYSTEM

TABLE OF CONTENTS

A. Project Summary

A1. Business Problem

The Customer

The Cake Shop is a thriving business with co-owners, administrative staff, and a dedicated team of bakers, operating across two strategic east coast locations: Manhattan, New York, and Newark, New Jersey. Known for its delicious baked goods, The Cake Shop has expanded its services and now provides in-person and virtual baking classes. As their business continues to grow, they need an appointment application to manage appointments for custom cake consultations and baking class requests, along with providing visibility of the business's scheduling and booking processes. The long-term goal would be to increase revenue by effectively managing The Cake Shop's schedule and providing the bakers with convenient appointment booking options. The short-term goal would be to increase customer satisfaction and reduce appointment inaccuracies by implementing the appointment booking application to streamline its current scheduling process.

Business Case

Since opening its doors, The Cake Shop has maintained a steady stream of business as a direct result of its successful social media campaigns. However, with the growth that the business has seen in a short amount of time, along with the opening of the Newark shop, they have faced problems with managing customer appointments and baking class schedules. The Cake Shop has seen online reviews from customers expressing their frustrations regarding overlapping appointments and confusion among staff members, which have resulted in the business losing revenue. With the appointment booking application, the business can provide its staff with the ability to schedule appointments in a single system, improving order fulfillment. It will also provide real-time visibility into the business's scheduling and booking processes for both locations.

Fulfillment

The proposed application will provide users with a comprehensive appointment booking and management system featuring a user-friendly interface that ensures an enhanced user experience. The application will offer a secure login functionality, allowing only authorized users to access the system, and will also provide structured views of appointment schedules, customer details, and reports. Users can add, modify, and delete appointments and customer information, including contact details and specific requirements, as well as schedule appointments during business hours. There will also be access to various reports that provide valuable insights into the business's scheduling and booking processes.

A. Project Summary

A2. Existing Gaps

Since the current process for appointments and class bookings is primarily handled in a single Excel spreadsheet shared between the two locations, it is fairly common that important customer details are missed because of how data is being entered. There have been multiple occasions where there have been appointment overlaps and bookings for baking classes have been overlooked.

These two features will be implemented to help with the existing gaps:

- **User Input Forms:** The forms will be developed so that they are easy to complete, and instructions for entering the required information will be made understandable. This will also help streamline data collection processes and ensure data accuracy.
- **User Interface:** The user interface designed will provide the user with structured views regarding important appointment information.

A3. Software Development Life Cycle

To support our software development activities, we will be using Agile methodology for the development of this application. This approach allows for early and continuous delivery of working software because it uses iterative processes. The Agile methodology will also enable the developers working on this application to get feedback in order to make adjustments throughout the development process. By using this methodology, we will be able to involve the owners of The Cake Shop in order to prioritize features that will best benefit the business.

Agile Process:

1. Ideation: This stage is where we define the goals, objectives, and scope of the application. We also need to identify and prioritize the initial requirements, which include documenting user stories to capture desired functionality. These user stories are then broken down into actionable tasks known as a "sprint".
2. Development: This stage of the process is when teams are able to begin building the first iteration of the application based on the tasks assigned in the sprint.
3. Testing: This stage is important to maintain software quality. The team tests the application to identify errors and bugs to ensure its functionality.
4. Deployment: When the application is at this stage of the process, it becomes available to the customer through the cloud or on-premise server.
5. Operations: This stage is when continued work is provided to the application in the form of maintenance (fixing bugs, maintaining functionality, future improvements, etc.).

A. Project Summary

Project Deliverables

- Deliverables from the project
  - Sprint Backlog - This deliverable breaks down tasks to be done in a specific time period. This is updated daily in order to track progress.
  - Test Plans - This is a strategic outline for testing the application. It provides the development team with details and procedures for testing activities to make sure that the intended requirements and program features are correct.

P*roduct Deliverables*

- Deliverables to the client
  - User Interface Design Layout - This will include both low and high-fidelity layouts to show the design for the application.
  - User Guide - This documentation is provided to users and provides instructions, information, and guidance as to how to effectively use the application.

A4. Implementation Plan

In order to implement the appointment booking system, it will be deployed in the production environment in a phased approach, ensuring minimal disruption to normal business operations. The implementation timeline will be established in collaboration with the owners of The Cake Shop, taking into consideration their schedules and requirements.

During the implementation, a dedicated team will be involved. Developers will be responsible for coding and customization, while project managers will oversee the implementation process, coordinating tasks and making sure that the team is meeting the established timeline. Infrastructure and deployment aspects will be handled providing a smooth transition to the production environment. Training and support will also be presented to users to ensure that they understand how to properly use the application and to address any questions or issues that may arise.

A5. Validation and Verification

To establish the quality and reliability of the application, validation and verification processes will be done at various stages of the implementation. Multiple methods of testing will be performed to validate the functionality and performance of the system. Any issues or bugs identified during the testing phase will be quickly addressed and resolved.

A. Project Summary

Unit Testing:
- Test each individual component of the application to verify that each component performs its intended function, such as login and adding an appointment.
- Tests will be performed by the developers.

Integration Testing:
- Test how different components of the application work together, such as verifying that the login module communicates correctly with the home screen.
- Tests will be performed by the developers.

System Testing:
- Test the entire application as a whole to ensure it meets all the required functionality.
- Verify that all user interactions and workflows function correctly, such as creating, modifying, and deleting appointments, as well as managing customer information.
- These tests will be performed by the development/testing team, who will simulate real-world scenarios and user interactions to make sure the system works as expected.

Acceptance Testing:
- The Cake Shop's owners and staff will perform tests based on predefined scenarios and provide feedback on the functionality and usability of the system.

A6. Environments and Costs

Programming Environment

- Java Version 17
- MySQL Version 8
- Windows 11 OS

A. Project Summary

Environment Costs

There will be several startup costs involved in the release of the appointment booking application to The Cake Shop. Since the application needs to be accessible to both shop locations (New York and New Jersey), the application and database will need to be housed on the cloud. Monthly and annual licensing costs are relatively minimal and both locations already have a thin client they had been utilizing for booking purposes, so there is no need to purchase anything additional.

- Amazon Relational Database Service (Amazon RDS): approx. $364.60/month
- Amazon Elastic Compute Cloud (Amazon EC2): approx. $704.30 upfront cost for 2 instances

Human Resource Costs

The approximate costs for human resources are indicated below. Although documentation and user guides will be provided for step-by-step instructions on how to use the application, training can also be provided at a cost.

- Developers (2): $45/hour per developer
- Project Manager(1): $30/hour
- Optional Training: $25/hour, plus travel/hotel accommodations

A. Project Summary

A7. Project Timeline

Phase	Milestone/ Task	Deliverable	Description	Dates
Requirements gathering and analysis	Task 1	Requirements	Meeting with client to discuss application scope and requirements	4/10/2023 - 4/14/2023
Design	Task 2/ Design files	1) Class Design 2) Low fidelity wireframe and high fidelity mockup	1) Create class design 2) Create UI according to UX/UI best practices	4/17/2023 - 04/21/2023
Application Development	Task 3 / Develop application	Functioning application	Development of core features and functionality of application	4/24/2023 - 5/15/2023
Testing and Debugging	Task 4/ Test and fix any bugs or issues	Test cases	Using different testing methods to identify issues that need to be fixed	5/15/2023 - 5/31/2023
Documentation and Training	Task 5/ Final release and training on new application	User guide	Training users on the new application and provide final release of application	5/31/2023 - 6/15/2023
Maintenance	Task 6/ Maintain application and future feature developments	Continued support for application	Developers will support application maintenance and any future developments	6/15/2023- 6/15/2024

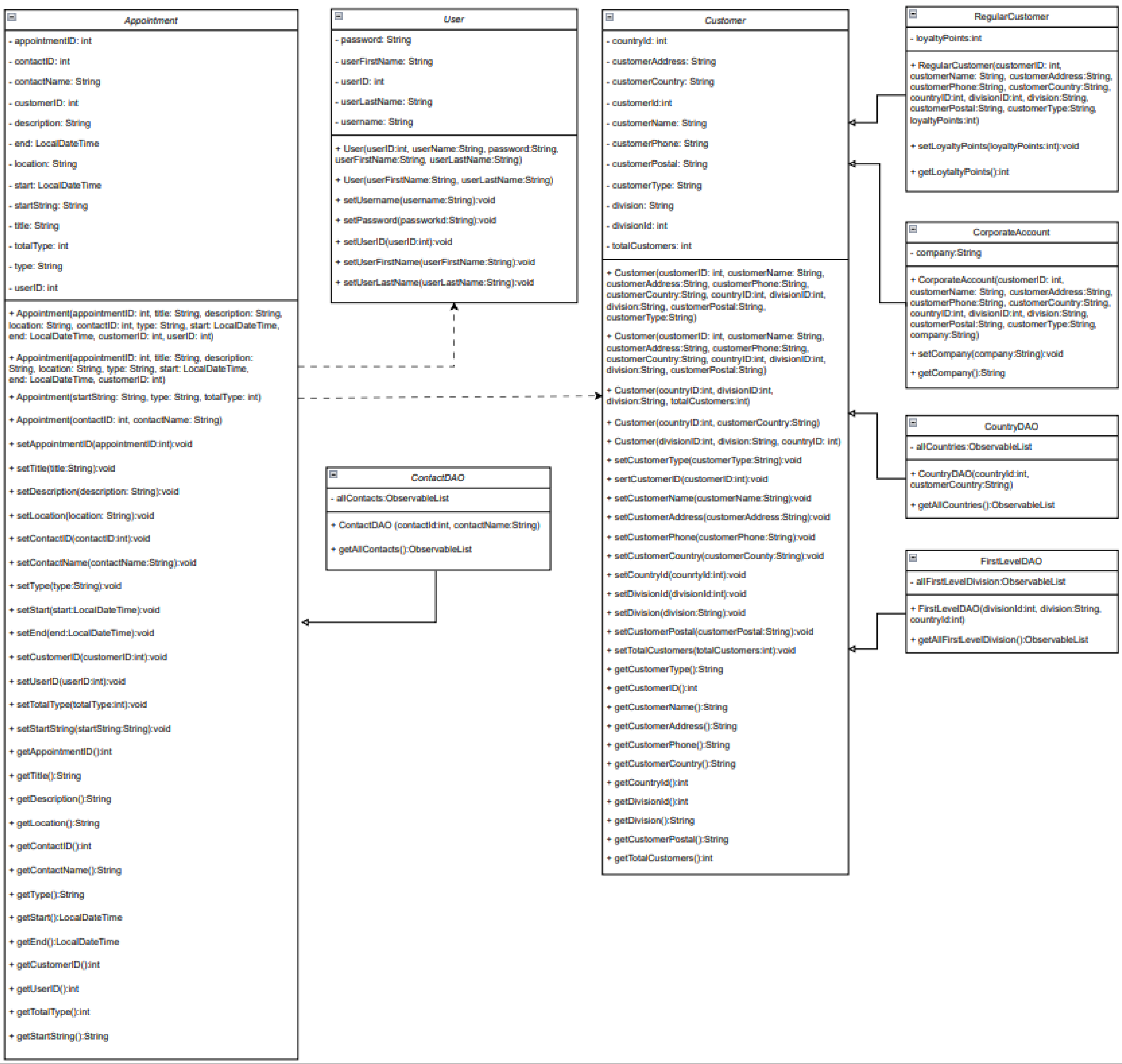
B. Development

See source code file attached to submission

C. Application Design & Testing

C1. Class Design

The Entity-Relationship Diagram (ERD) below will provide a visual representation of the appointment booking application. The primary classes in the application are shown here *Appointments, Customers, and Users.* There are also several query classes, which run queries in the database, and subclasses extending from superclasses. The ERD also shows the get and set methods for each class.



C. Application Design & Testing

C2. User Interface Design

The user interface for the appointment booking application consists of 22 total low-fidelity and high-fidelity user interface layouts: Login, Appointments, Add and Modify Appointments, Customers - Regular Customers, Customers - Corporate Accounts, Add and Modify Regular Customers, Add and Modify Corporate Accounts, and Reports. The designs were constructed in Figma.

The login screen provides input fields for the user to enter credentials along with a display of the timezone based on the user's system.

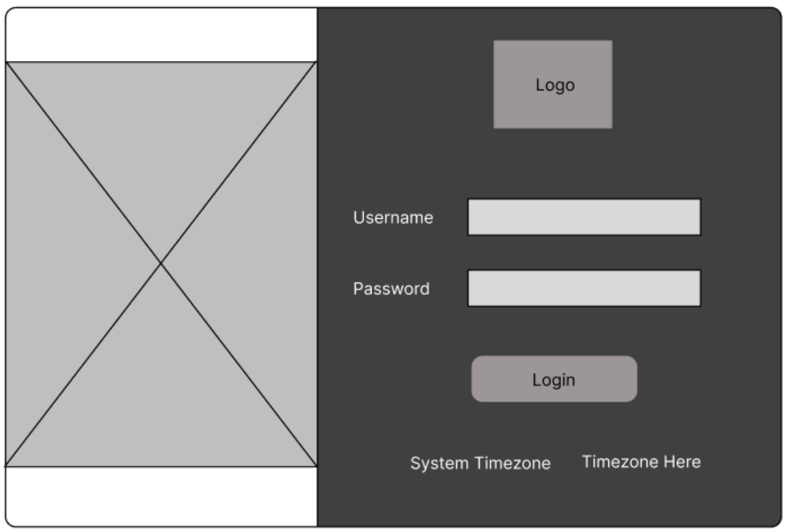


Figure 1: Low Fidelity Login Screen

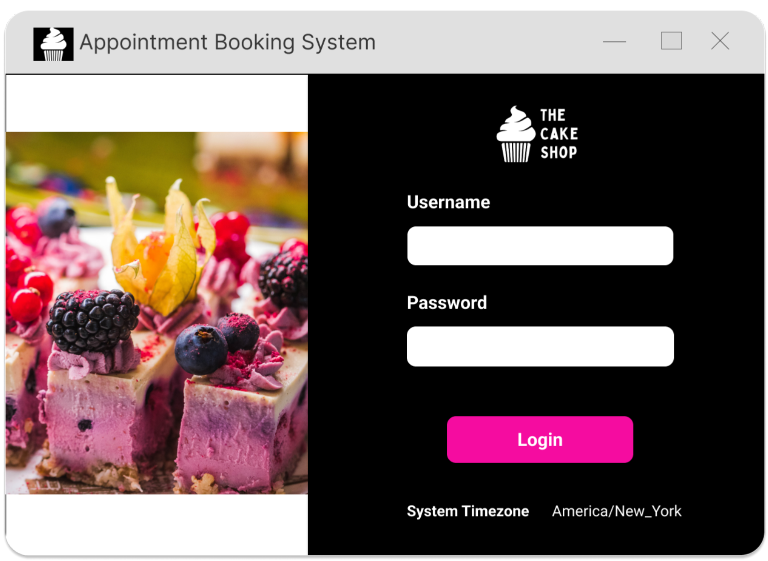


Figure 2: High Fidelity Login Screen

C. Application Design & Testing

C2. User Interface Design

The Appointments screen is the first screen that the user will see upon successful login into the system. The screen displays a custom welcome display for the user and the current date. There is a left panel navigation bar to direct users from each primary screen (Appointments, Customers, and Reports) and an option to log out of the application. Appointment information will be shown on the table with the ability to filter appointments via radio buttons. Actions such as adding a new appointment, viewing appointment details and canceling an appointment buttons are also present on this screen .

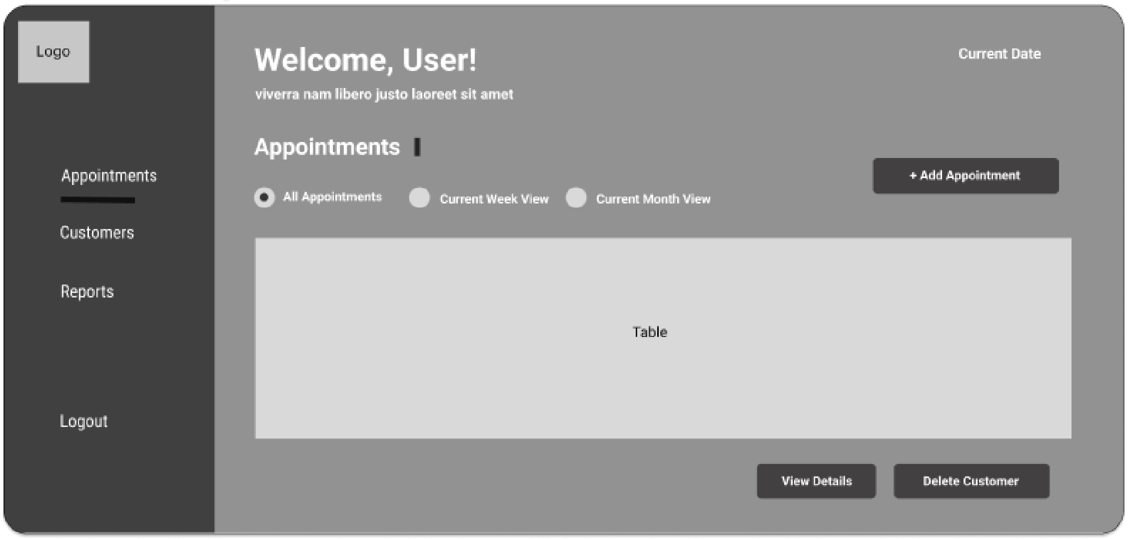


Figure 3: Low Fidelity Appointment Screen

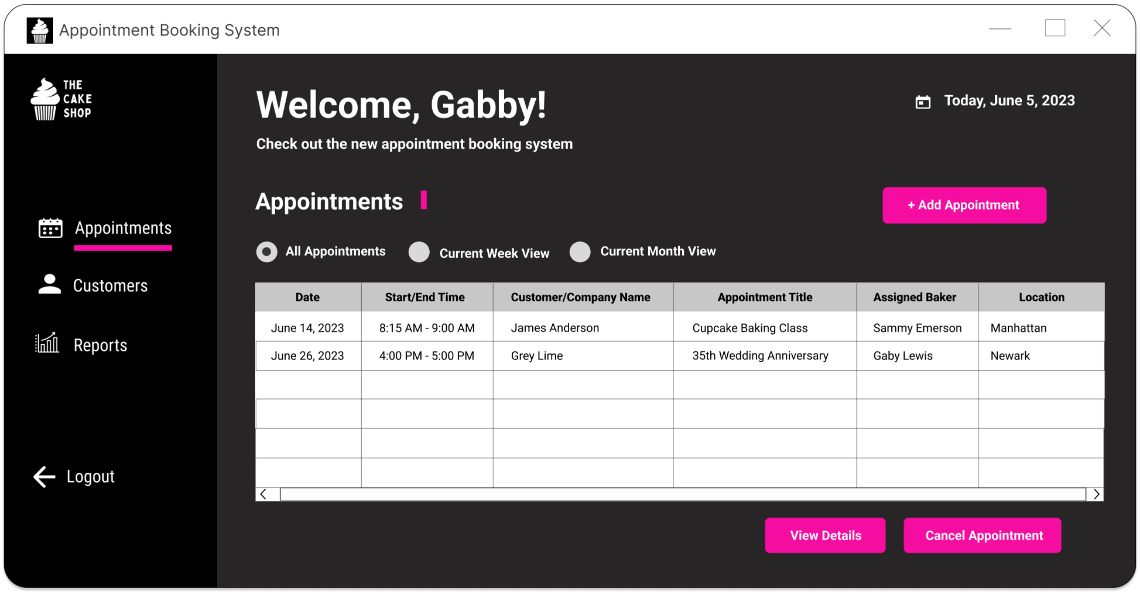


Figure 4: High Fidelity Appointment Screen

C. Application Design & Testing

C2. User Interface Design

This *Add Appointment* screen provides the user with a form with input fields where text can be entered or a drop-down option is selected. The *Save* and *Cancel* buttons are at the bottom of the screen for users to click when appropriate. For the user's convenience, the company's business hours will be provided in the top right corner of the screen.

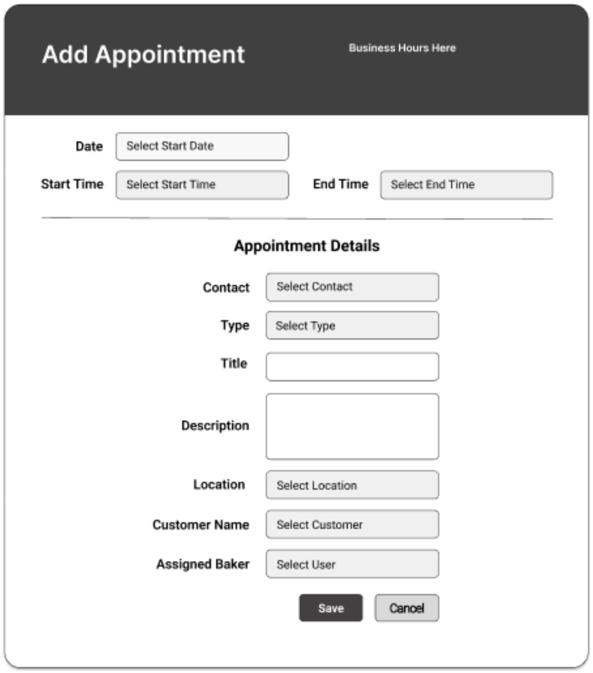


Figure 5: Low Fidelity Add Appointment Screen

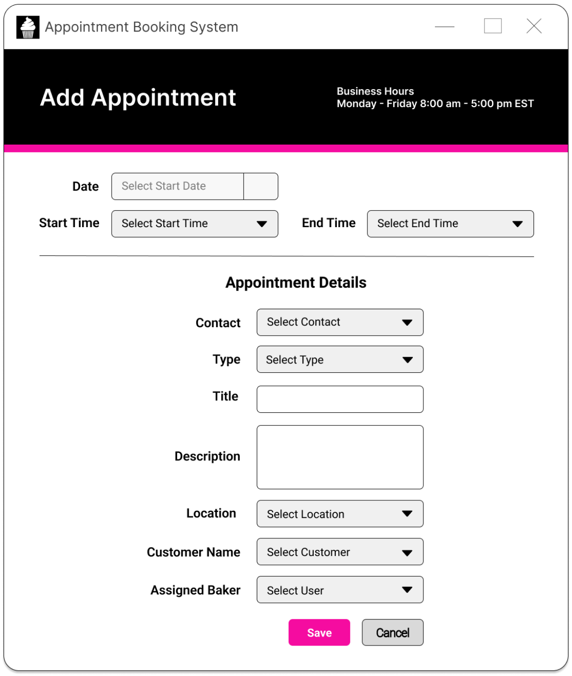


Figure 6: High- Fidelity Add Appointment Screen

C. Application Design & Testing

C2. User Interface Design

This *View Appointment Details* screen contains the information about the selected appointment. The user will be able to modify the appointment information and save changes by using text and drop-down options. The *Update* and *Cancel* buttons are at the bottom of the screen to use when appropriate.

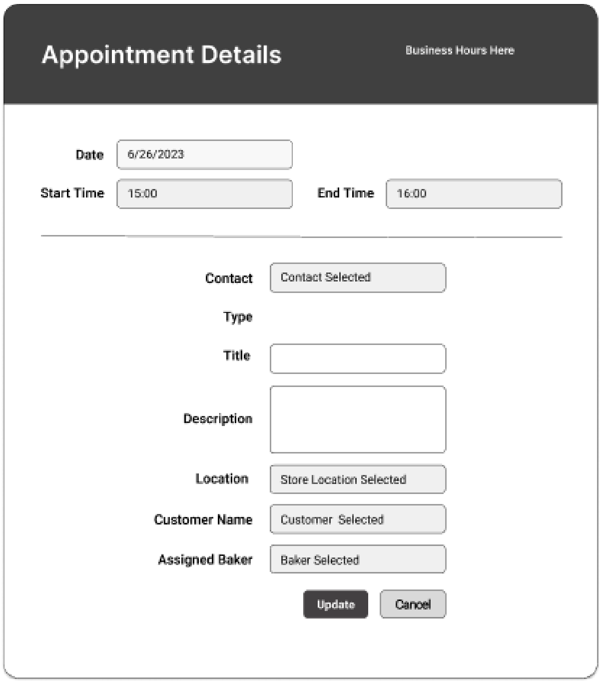


Figure 7: Low Fidelity Modify Appointment Screen

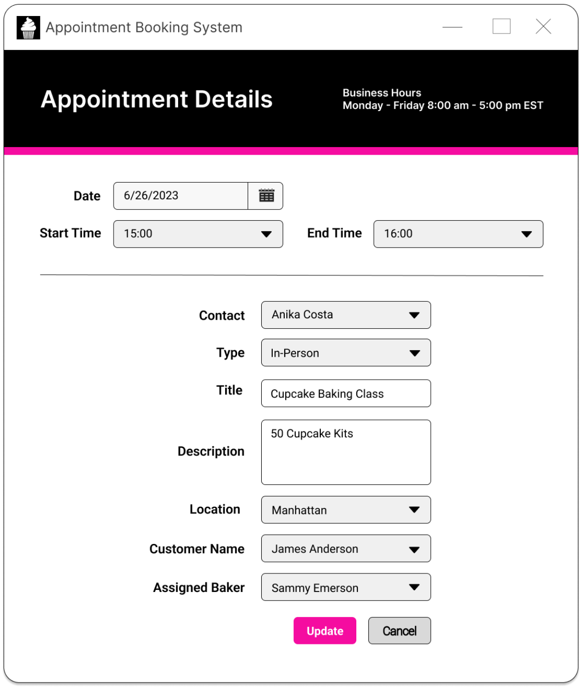


Figure 8: High- Fidelity Modify Appointment Screen

C. Application Design & Testing

C2. User Interface Design

The *Customers* primary screen (left navigation panel option) shows customer information on a table that can be filtered by Regular Customers (default display) and Corporate Accounts by utilizing the radio buttons. A search feature is also present for users to search through the respective options. The user will also be able to add new customers, view customer details, and delete customers.

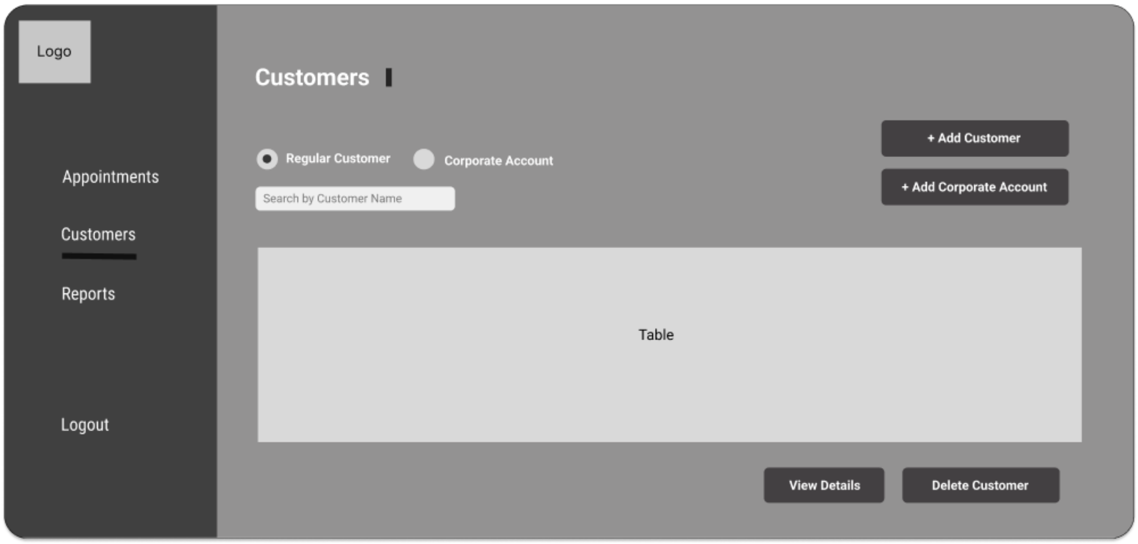


Figure 9: Low Fidelity Customer - Regular Customer Screen

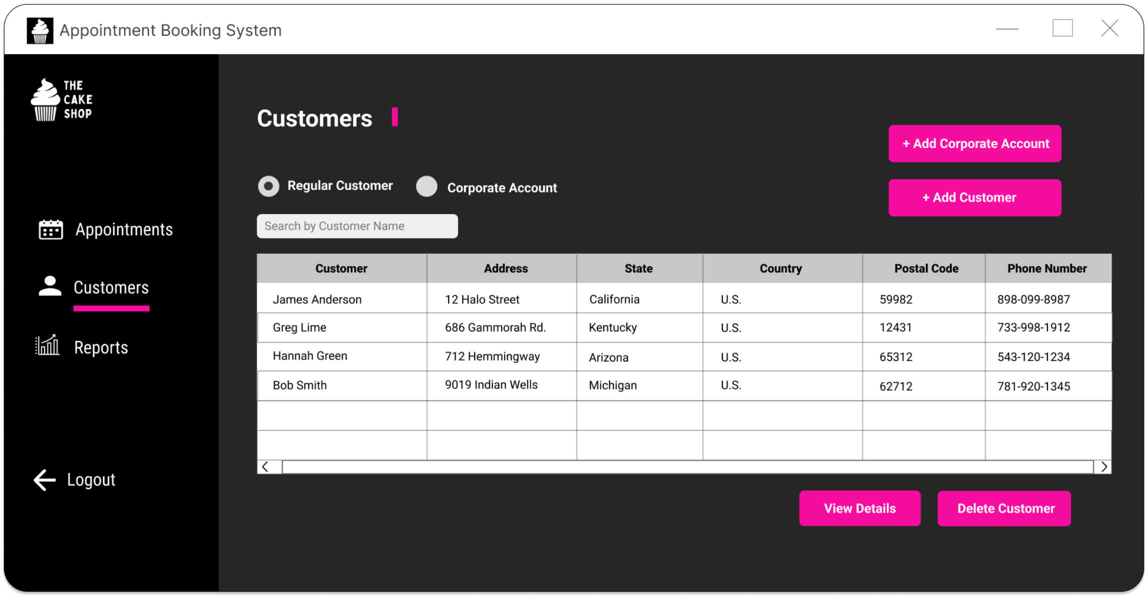


Figure 10: High Fidelity Customer - Regular Customer Screen

C. Application Design & Testing

C2. User Interface Design

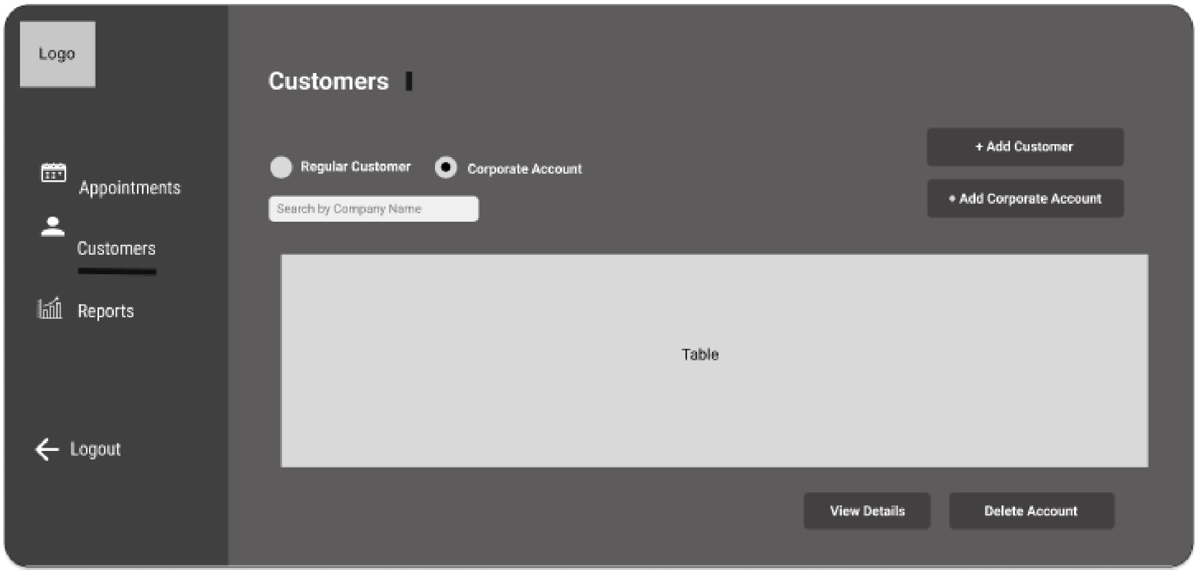


Figure 11: Low Fidelity Customer - Corporate Account Screen

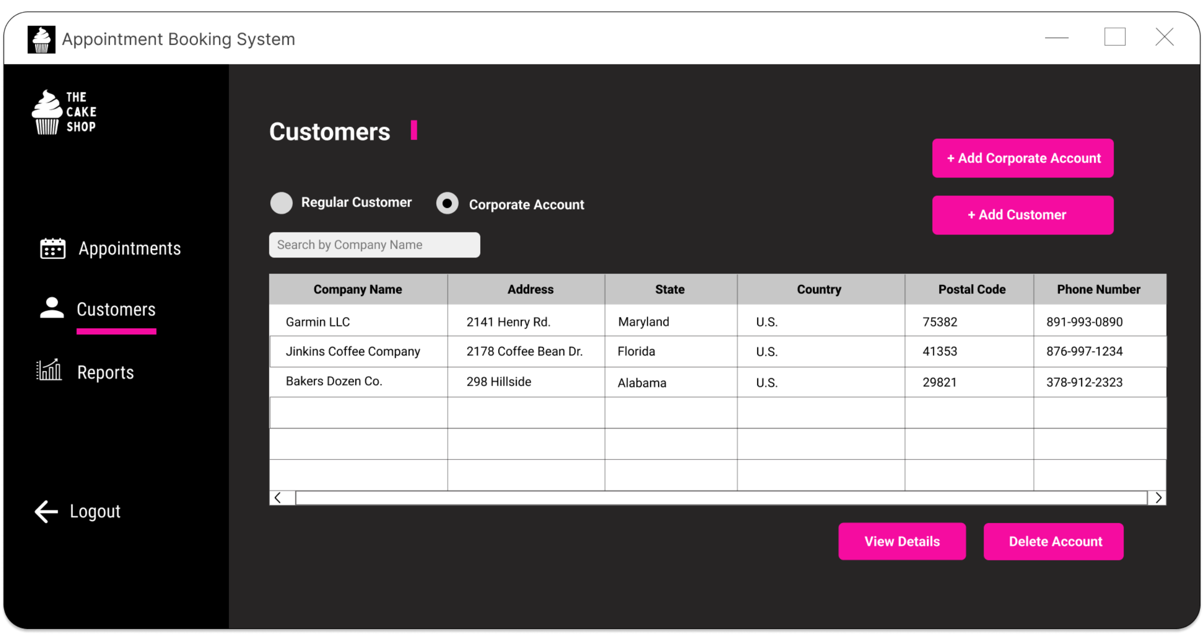


Figure 12: High Fidelity Customer - Corporate Account Screen

C. Application Design & Testing

C2. User Interface Design

This *Add Customer* (Regular Customer) screen provides the user with a form with input fields where text can be entered or a drop-down option is selected. The *Save* and *Cancel* buttons are at the bottom of the screen for users to click when appropriate.

Figures 15 & 16 - *Add Corporate Account* also functions in the same manner.

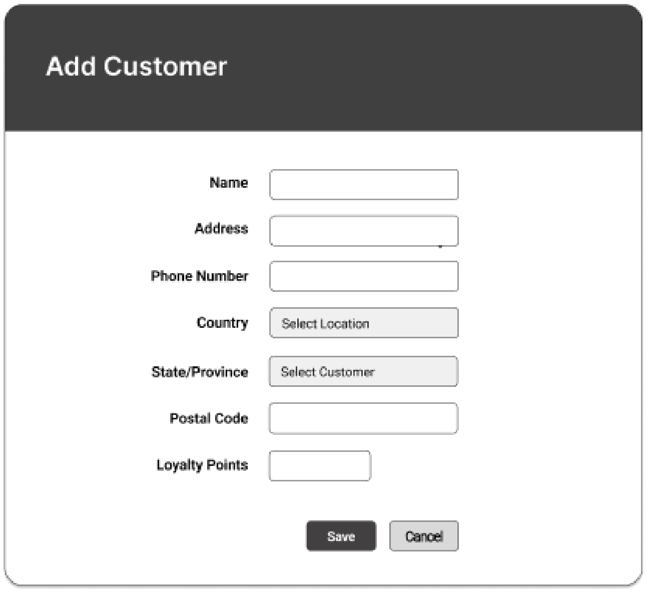


Figure 13: Low Fidelity Add Regular Customer Screen

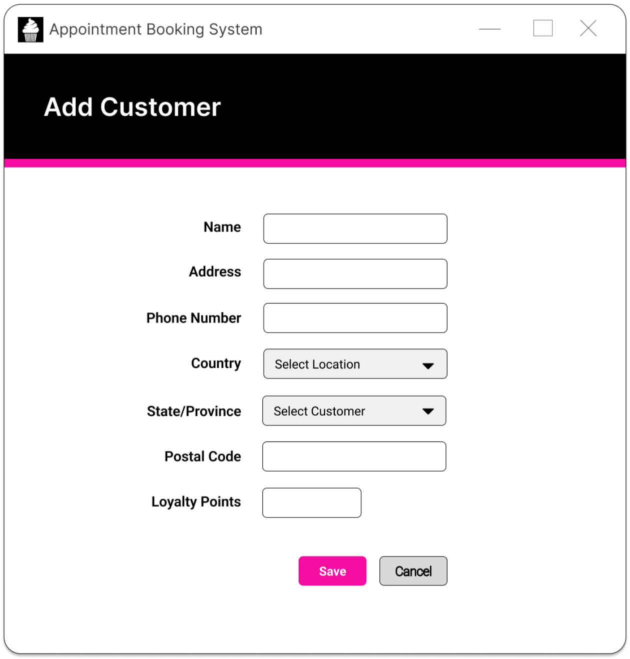


Figure 14: High Fidelity Add Regular Customer Screen

C. Application Design & Testing

C2. User Interface Design

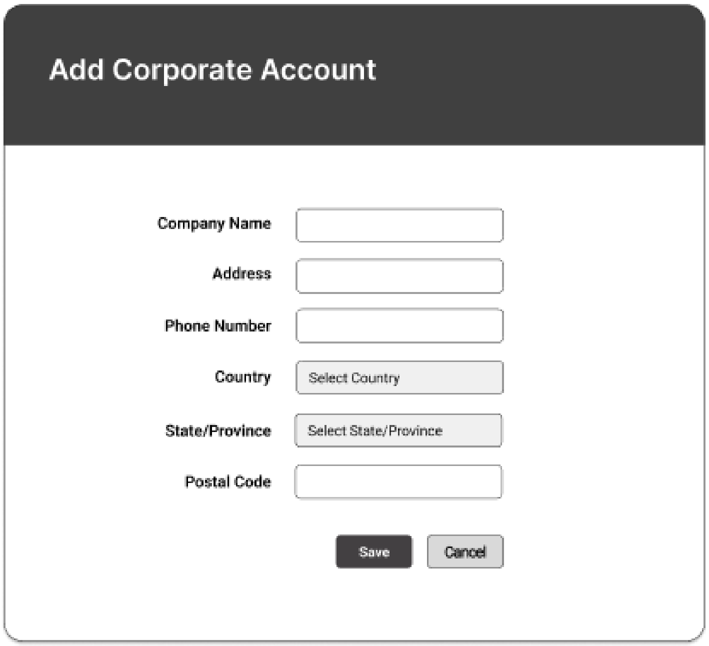


Figure 15: Low Fidelity Add Corporate Account Screen

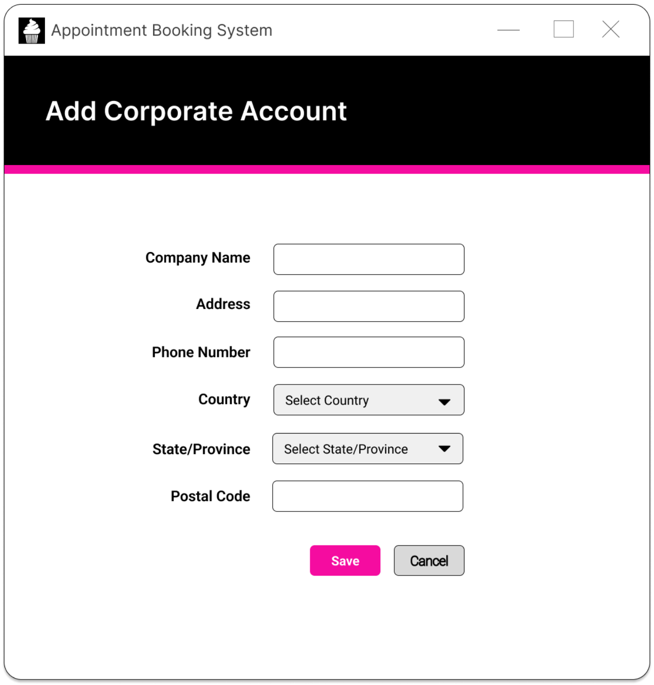


Figure 16: High Fidelity Add Corporate Account Screen

C. Application Design & Testing

C2. User Interface Design

This *View Customer Details* screen contains information about the selected customer. The user will be able to modify the customer information and save changes by using text and drop-down options. The Update and Cancel buttons are at the bottom of the screen to use when appropriate.

Figures 19 & 20 - Modify Corporate Account also functions in the same manner.

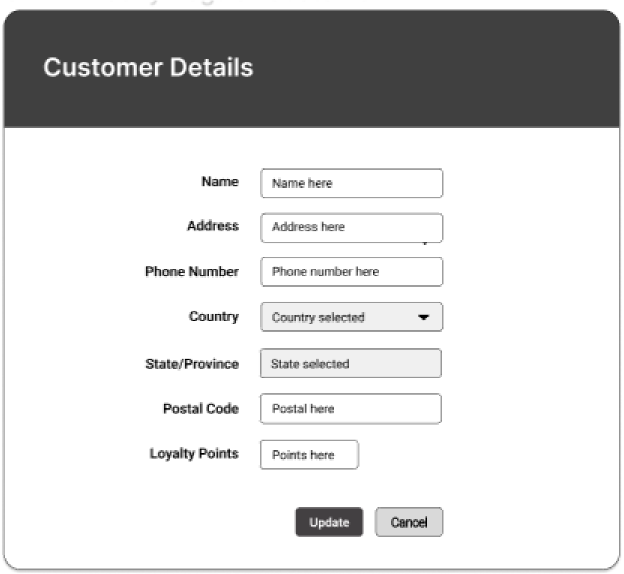


Figure 17: Low Fidelity Modify Regular Customer Screen

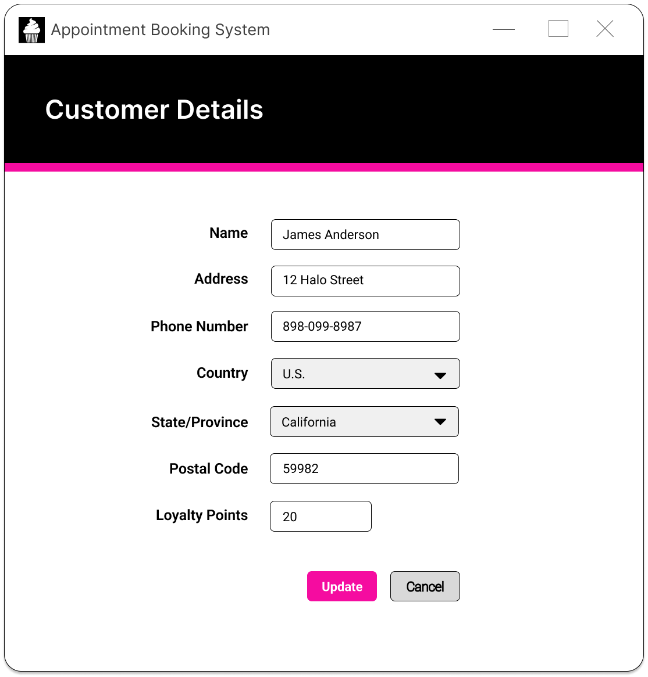


Figure 18: High Fidelity Modify Regular Customer Screen

C. Application Design & Testing

C2. User Interface Design

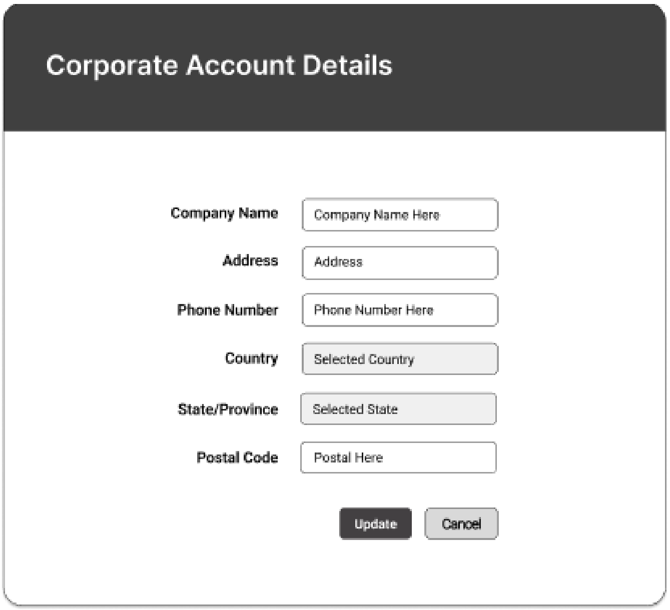


Figure 19: Low Fidelity Modify Corporate Account Screen

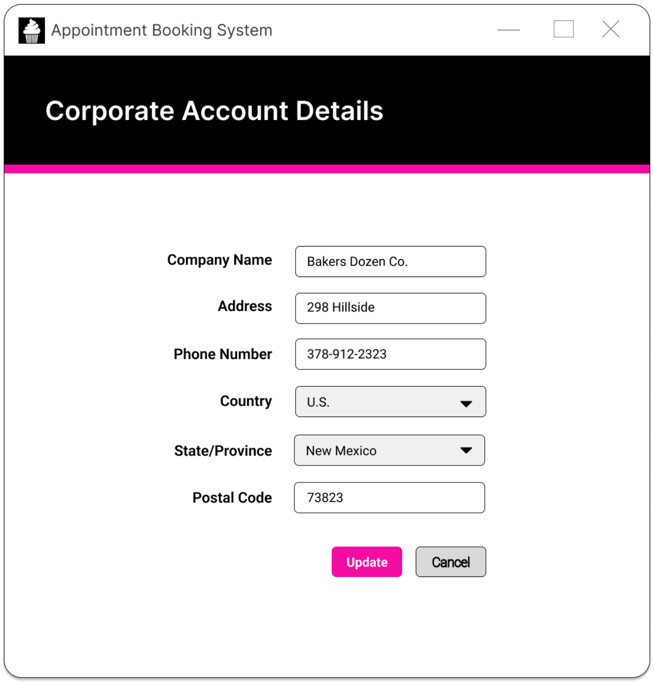


Figure 20: High Fidelity Modify Corporate Account Screen

C. Application Design & Testing

C2. User Interface Design

The *Reports* primary screen (left navigation panel option) shows reports structured on multiple tables, along with a filter option for the first report.

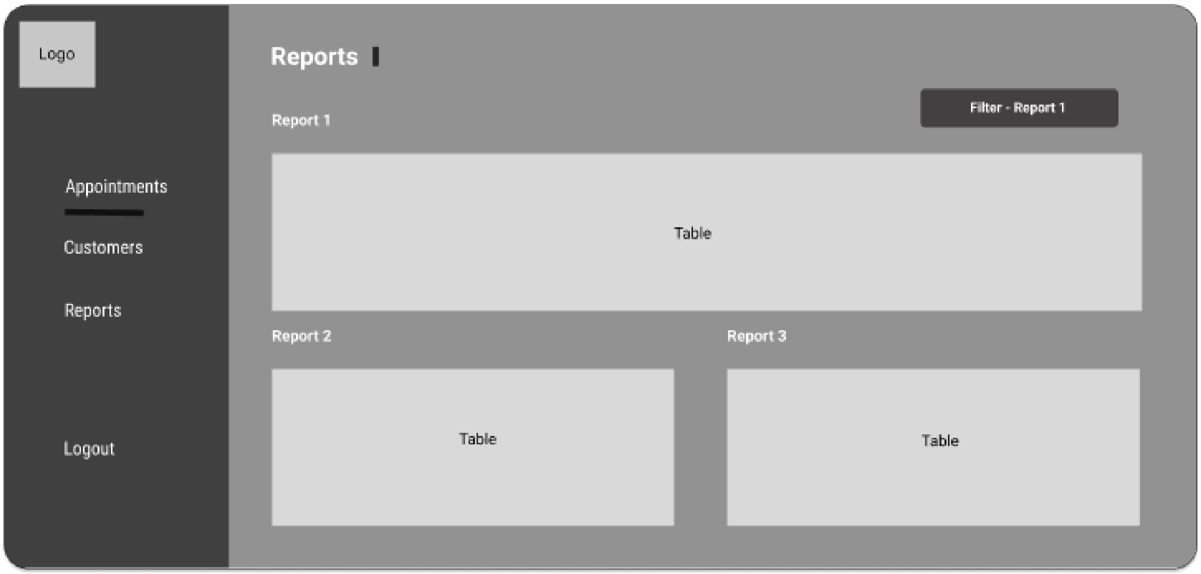


Figure 21: Low Fidelity Reports Screen

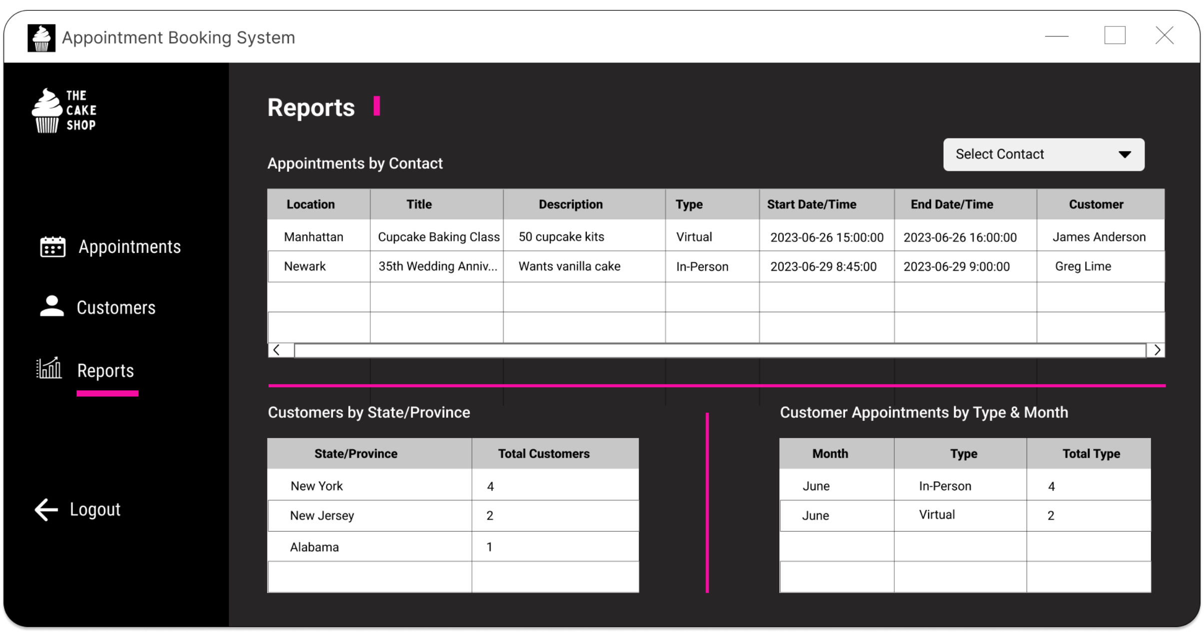


Figure 22: High Fidelity Reports Screen

C. Application Design & Testing

C3. Unit Test Plan

Purpose

During the development of the application, various testing methods were used to ensure its quality and functionality. Black Box testing was used predominantly on this application for functionality testing. For example, the login feature was tested to validate user credentials against the stored data in the MySQL database. Similar approaches were followed for testing the add, modify, and delete functionalities for appointment and customer management, and reporting features.

If issues were identified during testing (mainly through test cases), the appropriate steps were taken to remediate those issues. An example of this would be the use of debugging procedures to find the cause of application failure. Once the issue was fixed, the test case would be executed again to verify that the issue was fixed and that the feature was correctly functioning.

Overview

The test plan for adding an appointment for the appointment booking system involves verifying the functionality of the appointment addition process. The test will be focused on ensuring that appointments can be successfully added with the correct information and that appropriate validation checks and error-handling procedures are in place. The test approach used will consist of unit testing using black box techniques with test cases being executed to verify the behavior of adding a new appointment.

Test Plan

- **Items**
  - Test data for appointments (Title, Description, Location, Type, Start, End, Create Date, Created By, Last Update, Last Updated By, Customer ID, User ID, and Contact ID)
  - Test environment (appointment booking application is installed and running) along with login credentials
  - Access to MySQL database for storing and retrieving data
- **Features**
  - UI for adding appointments
  - Database operations in order to store appointment data
- **Deliverables**
  - Test Results/Reports
  - Test procedures/guidelines (developer documentation)

C. Application Design & Testing

- **Task**
  - Set up the required test environment, including the installation of the appointment booking application and configuring the MySQL database for access
  - Create test cases that are designed to validate the functionality of adding a new appointment
  - Execute the steps identified in the test cases
  - Document any issues that were encountered during testing and their outcomes
- **Needs**
  - Application being tested
  - Java Version: 17
  - MySQL Connector Driver Version: 8.0.31 (MySQL Community Server-GPL)
  - Windows 11
- **Pass/Fail Criteria**
  - Pass
    - The appointment is added to the system when valid data is provided.
    - The application provides the appropriate validation and error messages when missing or invalid data is provided.
    - The user receives a confirmation message upon the successful addition of an appointment.
    - The appointment data is accurately stored in the MySQL database.
    - The test results align with the expected outcomes of each test case.
  - Fail
    - The appointment fails to be added to the system when valid data is provided.
      - Remediation: Proper validation checks need to be implemented to ensure that all fields are filled with valid data prior to adding the appointment to the system. Appropriate error messages need to be displayed indicating invalid or missing data.
      - Documentation: Document validation rules and error messages for missing information in the developer documentation.
    - The application does not display the appropriate error messages when missing or invalid information is provided.
      - Remediation: Review error handling and messaging logic to provide clear error messages for missing or invalid data input.
      - Documentation: Error messages and their corresponding scenarios are documented in the developer documentation.
    - The test cases produce unexpected errors or exceptions.
      - Remediation: Identify the cause of the errors or exceptions by reviewing the application code and logs. Debug and fix the issues.

        - Documentation: Errors and exceptions, along with their resolutions, should be documented in the developer documentation.
    - The test results do not align with the expected outcomes of each test.
        - Remediation: Analyze the difference between the actual and expected results of the tests. Identify the cause and fix issues in the code.
        - Documentation: Document differences found, causes, and solutions that were implemented.
- **Procedures**
    - Testing with Valid Data:
        - Test Process: Enter valid appointment details into user input fields. Click the "Save" button to save the appointment.
        - Expected Result: The appointment is successfully added to the database and can be viewed in the database table called "appointments". The user receives a confirmation message and the new appointment can be viewed on the table of appointments on the Appointments screen.
        - Pass/Fail Criteria: Pass
    - Testing with Invalid Data:
        - Test Process: Omit one or more fields empty or provide invalid data. Click the "Save" button to save the appointment.
        - Expected Result: An error message is displayed indicating missing or invalid information, and the appointment is not added to the system.
        - Pass/Fail Criteria: Pass

Results

Black box testing was used for testing the functionality of units like adding an appointment. However, this method was also utilized throughout the application to establish software quality. Once the unit tests were cleared with a pass in the pass/fail criteria, regression testing needed to be completed in order to ensure that all units were working cohesively with the rest of the application.

C. Application Design & Testing

C4. Source Code

See zip file along with this submission named:
C868_V1_JasmineLangford.zip

C5. User Guide

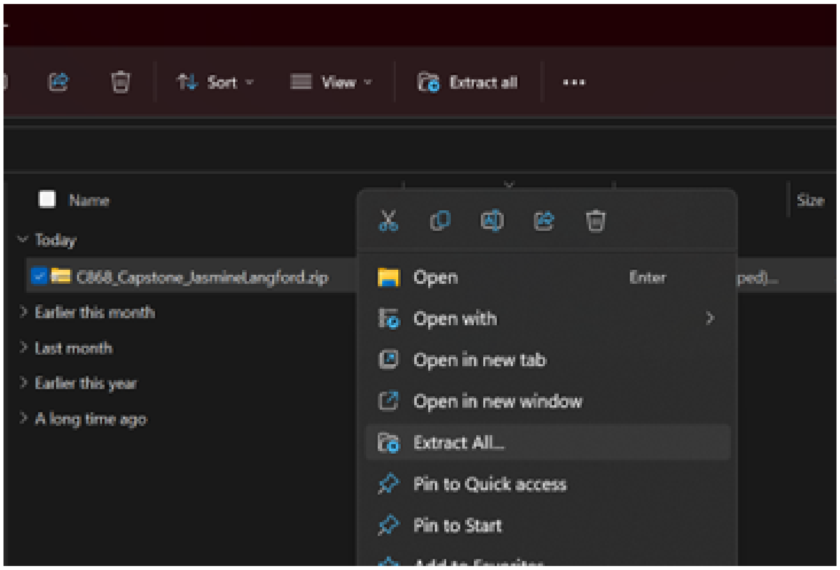
Introduction

This user guide provides a comprehensive description of the content and functionality of the appointment booking application. It includes detailed instructions on how to install the program, log into the application, and utilize its functions. The installation process will be given to ensure that the user can successfully set up the application. In addition, the guide will provide step-by-step instructions for logging into the system with the appropriate credentials and how to effectively use the application's features.

Installation and Using the Application

1. System requirements will need to be met prior to installing the application. These requirements include Windows 11, Java Development Kit (JDK) 17, and the ability to access the MySQL database. The MySQL connector driver was used as a bridge between this Java application and the MySQL database.

2. Extract the zip file to a location in File Explorer. This can be done by right-clicking on the zip file and selecting *Extract All*.



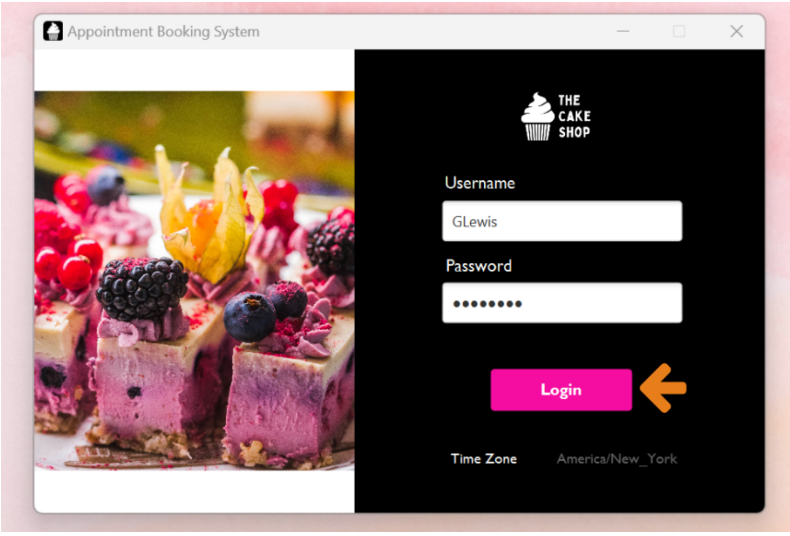
C. Application Design & Testing

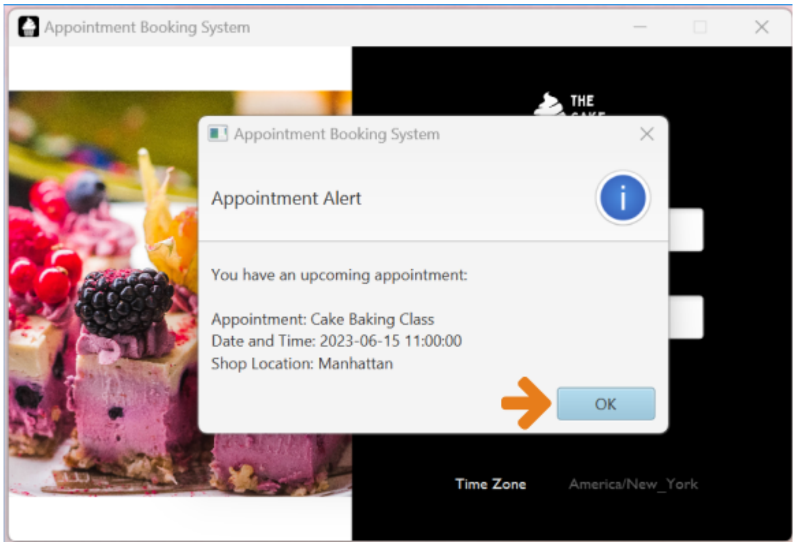
3. Click on the extracted file and navigate to the executable JAR file. This file will launch the application and a login page will appear.

Login

4. The login screen will provide input fields to enter credentials. Enter the following username and password and click the *Login* button to access the application. If input fields are left blank or invalid credentials have been entered, the system will provide an input validation message to enter valid credentials. Upon successful login, an appointment alert will be displayed (even if there are no appointments within 15 minutes of logging in). Click *Ok* to be directed to the *Appointments* screen.

Username: GLewis
Password: cakery101

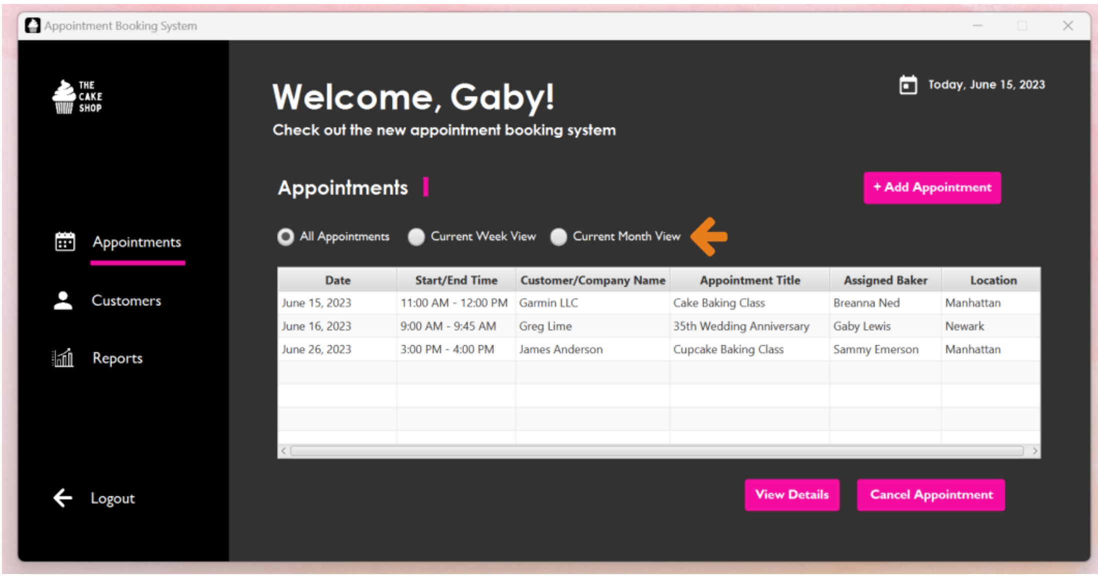




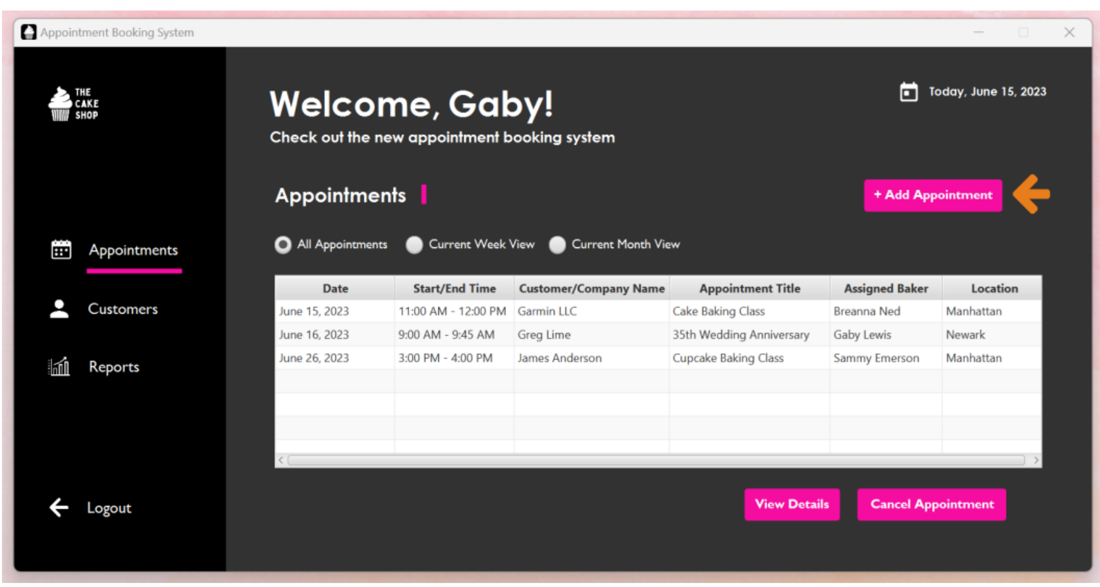
C. Application Design & Testing

Appointments

5. The Appointments screen displays appointment information. Choose the desired radio button above the table of appointments to filter appointments by all appointments (default display), current week, or current month.



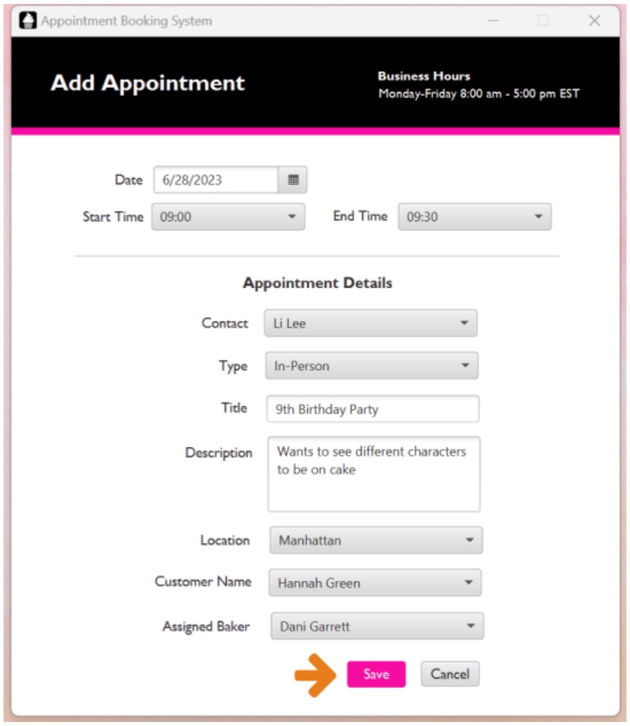
6. Click on the *Add Appointment* button near the top right of the screen to add a new appointment. Once clicked, the *Add Appointment* form will load on another screen.

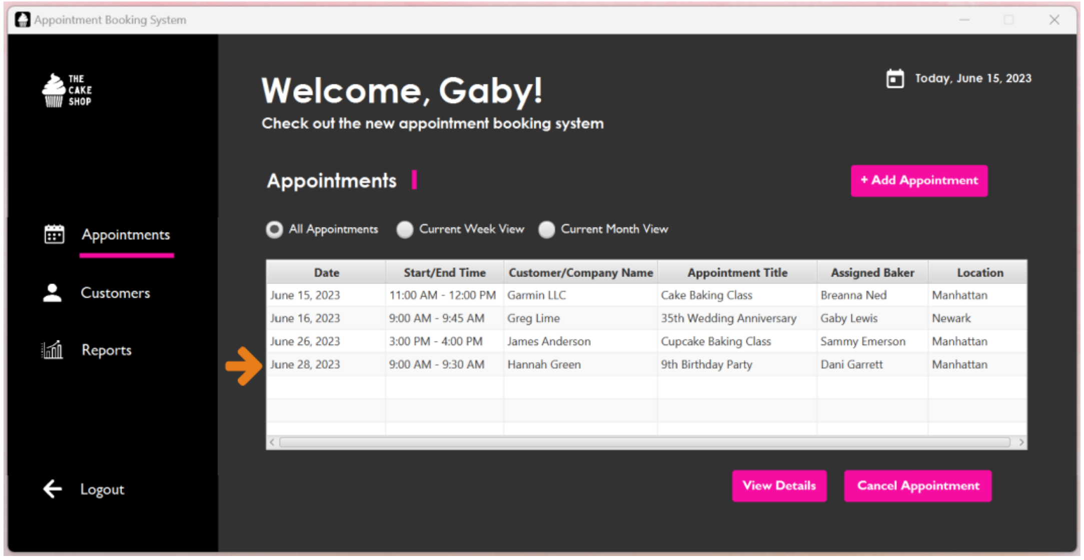


C. Application Design & Testing

Appointments

7. Fill in the *Add Appointment* form with text in the input fields and select the desired option from the drop-down menus. Click the *Save* button to add the new appointment. An alert will be displayed stating a new appointment was added. The new appointment will then be displayed on the *Appointments* table on the next screen. If invalid information is entered, input validation messages will be shown with additional directions.

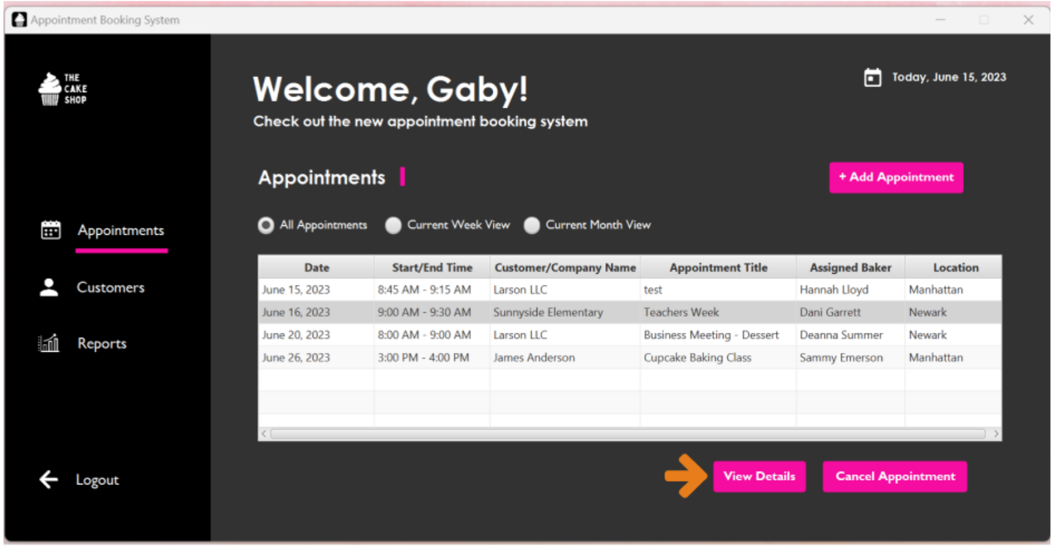


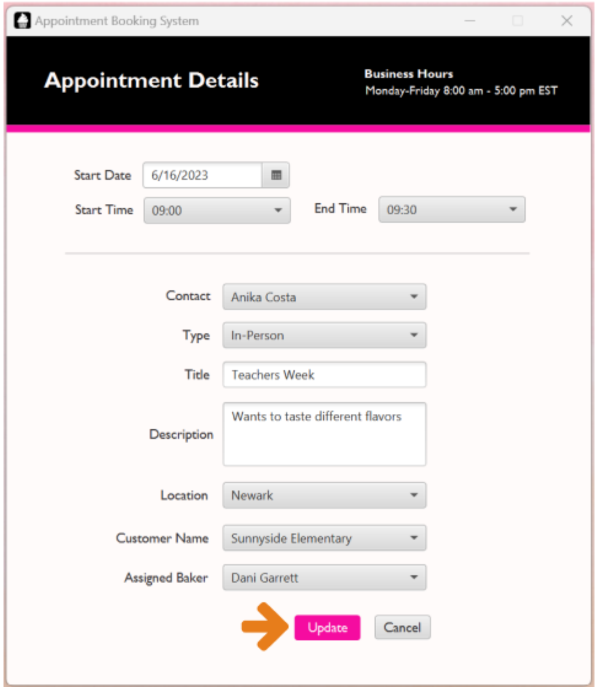


C. Application Design & Testing

Appointments

8. Select an appointment on the table and click the *View Details* button located under the *Appointments* table. The *Appointment Details* screen will populate the information for the selected appointment. Make the desired changes and click the *Update* button to modify the appointment. Changes to the appointment will be reflected on the table on the *Appointments* screen. If there are no changes to the appointment details, click on the *Cancel* button to be directed back to the *Appointments* screen.

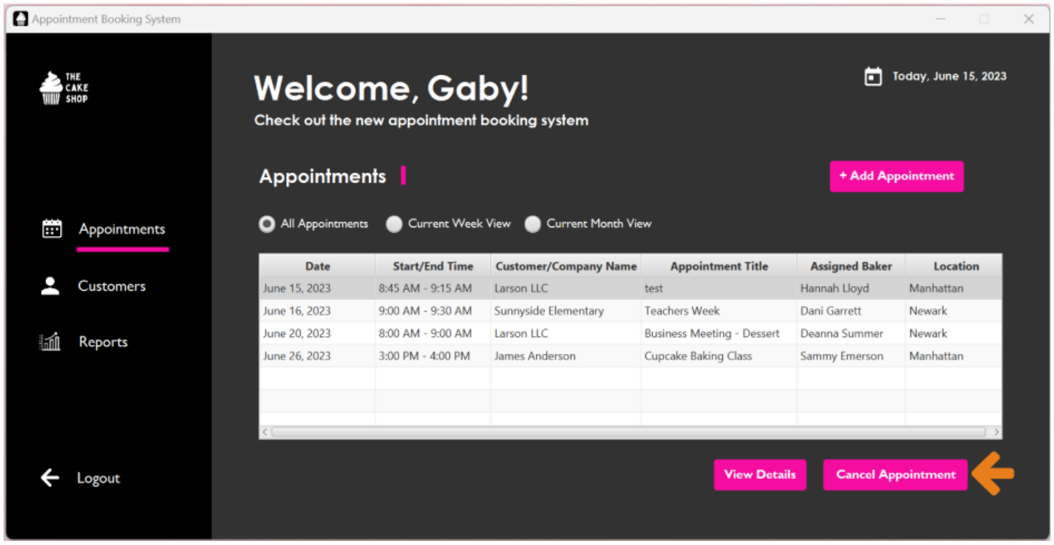


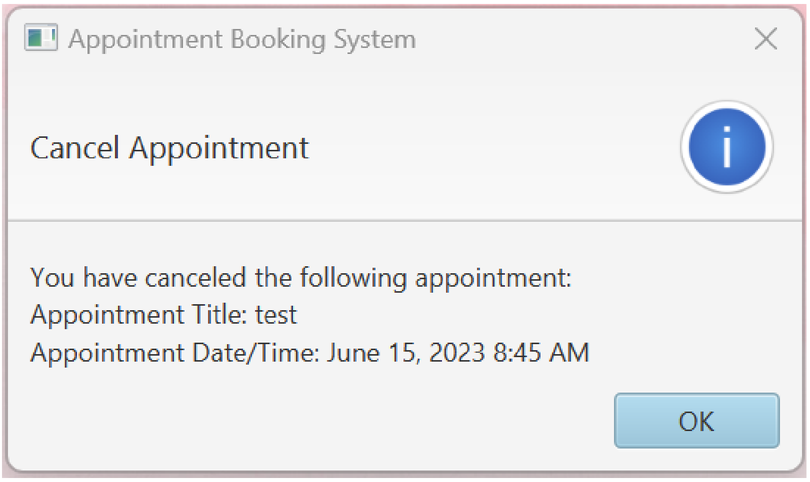


C. Application Design & Testing

Appointments

9. Select an appointment and click the *Cancel Appointment* button located under the *Appointments* table to delete the selected appointment. A confirmation message will be prompted. Click *Yes* to delete the appointment. An alert will be displayed indicating that the selected appointment was canceled.

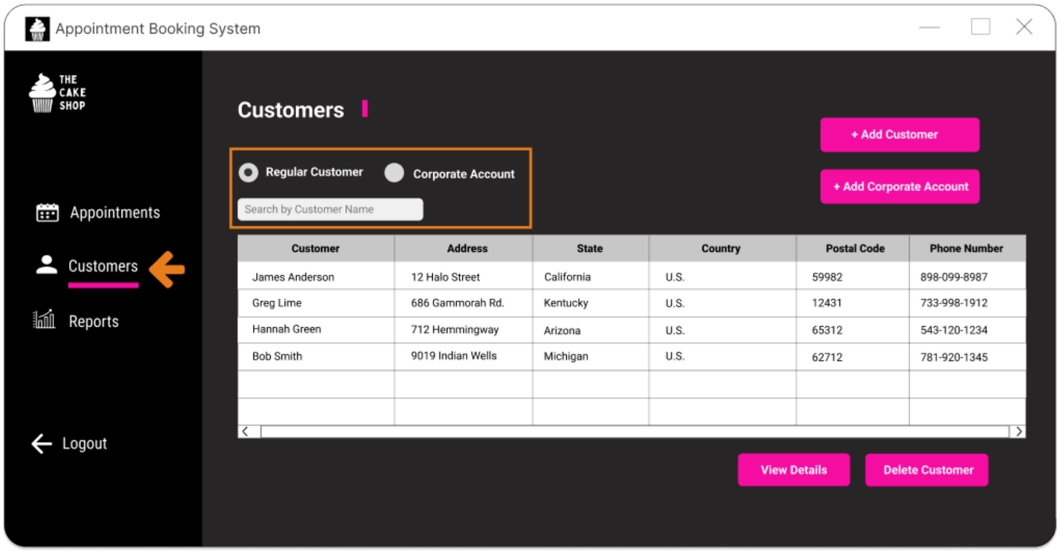


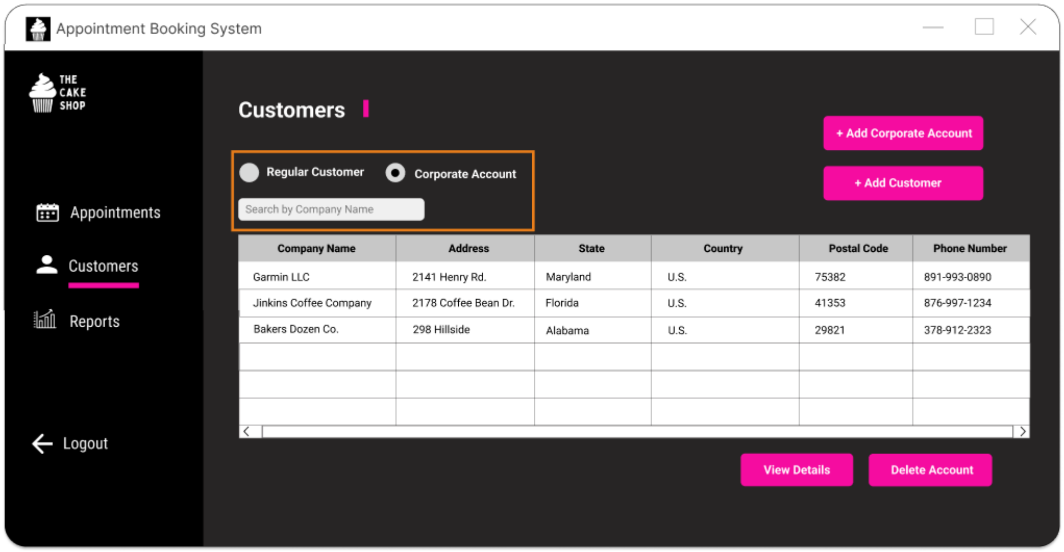


C. Application Design & Testing

Customers - Regular Customer

10. Navigate to the *Customers* screen by utilizing the left panel navigation bar. This screen displays customer information. Click the radio buttons above the search bar to filter customers by Regular Customers (default display) and Corporate Accounts. Once filtered, searching for Regular Customers by customer name or Corporate Accounts by company name can be done by typing in the search bar.

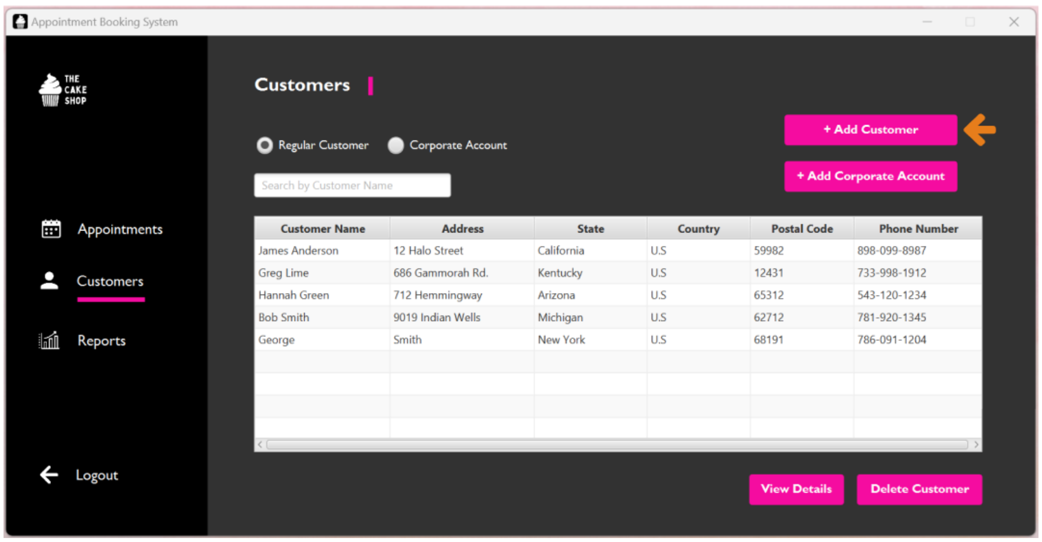


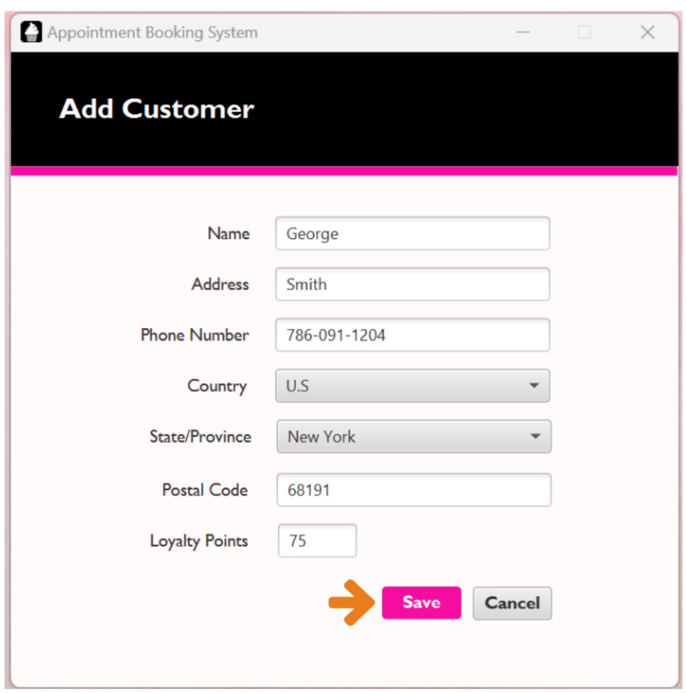


C. Application Design & Testing

Customers - Regular Customer

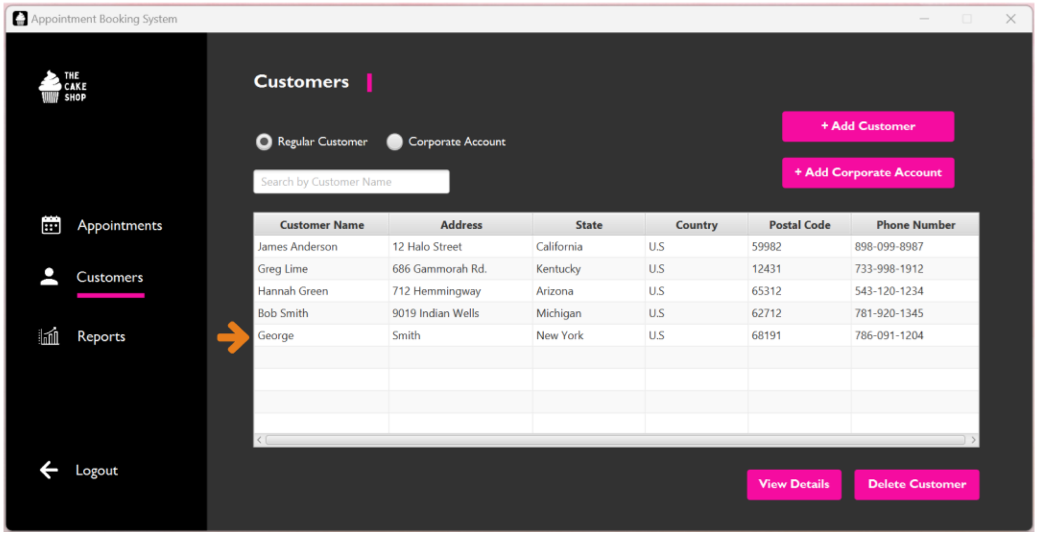
11. Click on the *Add Customer* button near the top right of the screen to add a new regular customer. Once clicked, the *Add Customer* form will load on another screen. Fill in the *Add Customer* form with text in the input fields and select the desired option from the drop-down menus. Click the Save button to add the new customer. An alert will be displayed stating a new customer was added. The new customer will then be displayed on the Regular Customers table on the next screen. If invalid information is entered, input validation messages will be shown with additional directions.



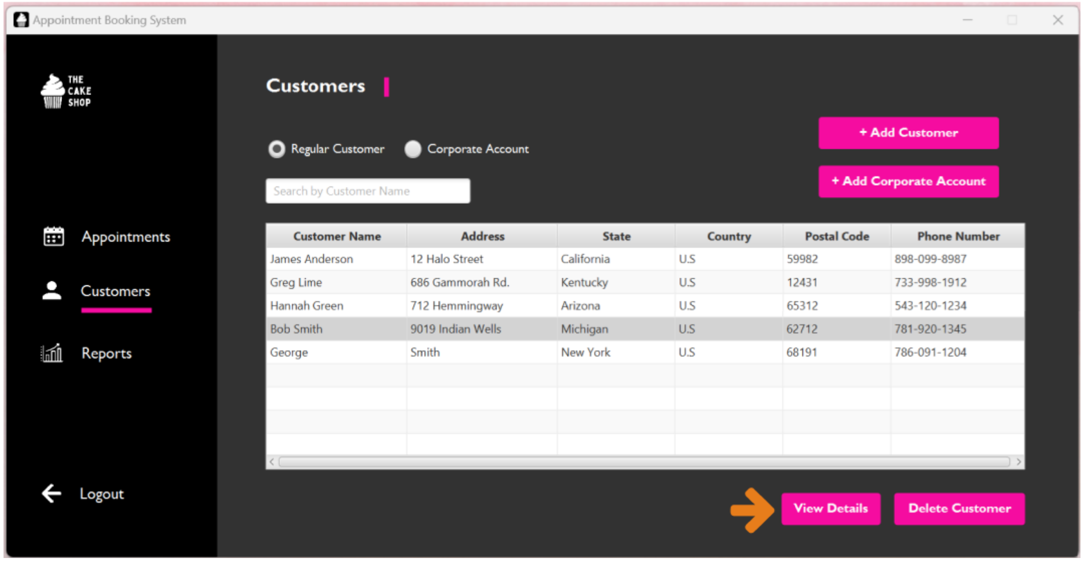


C. Application Design & Testing

Customers - Regular Customer

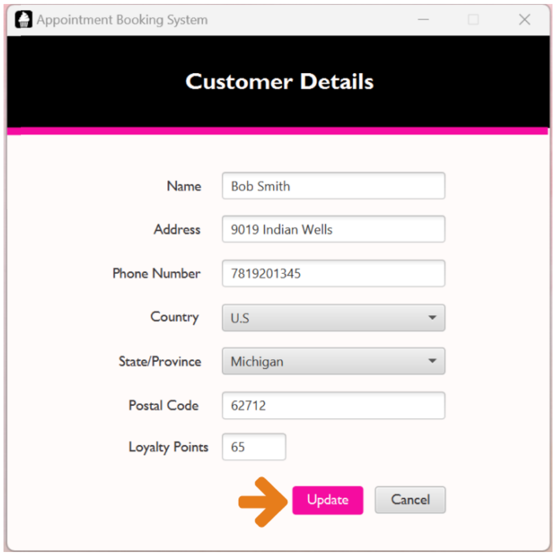


12. Select a customer from the table and click the *View Details* button located under the Customer table to navigate to the Customer Details screen where information on the selected regular customer can be viewed in its entirety. Make the desired changes and click the *Update* button to modify the customer. Changes to the customer will be reflected on the table on the Regular Customer screen. If there are no changes to the customer details, click on the Cancel button to be directed back to the Regular Customer screen.

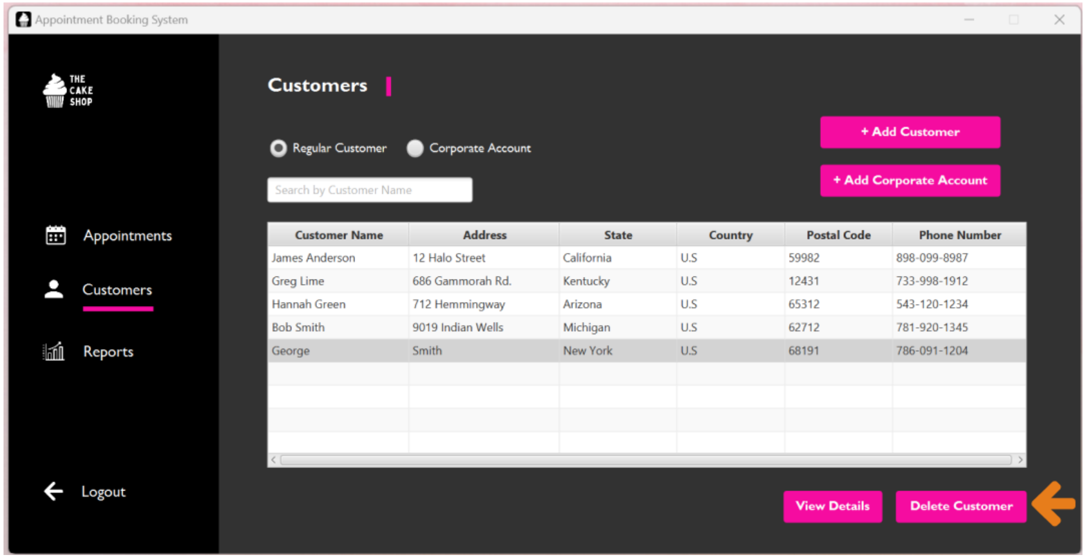


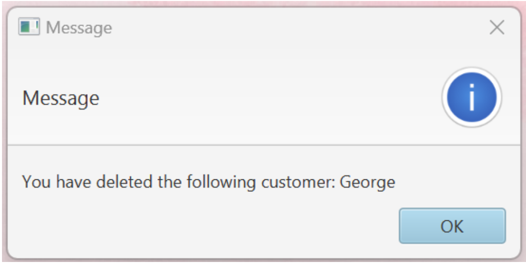
C. Application Design & Testing

Customers - Regular Customer



13. Select a customer and click the *Delete Customer* button located under the Regular Customer table to delete a selected customer. A confirmation message will be prompted. Click *Yes* to delete the customer. An alert will be displayed indicating that the selected customer was deleted.

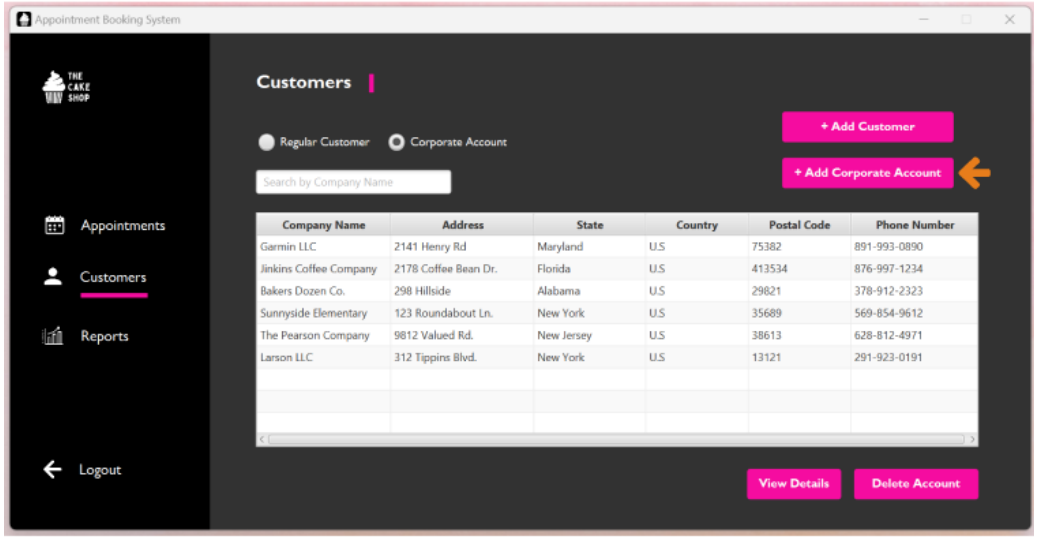


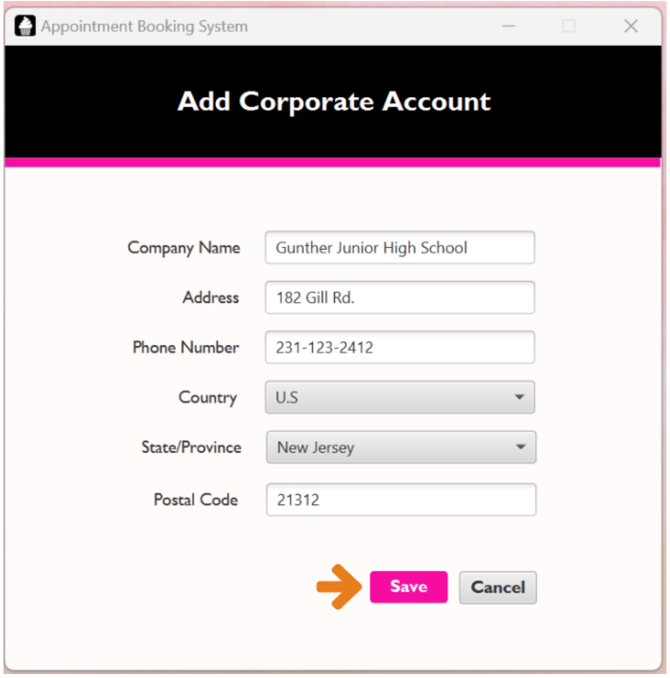


C. Application Design & Testing

Customers - Corporate Accounts

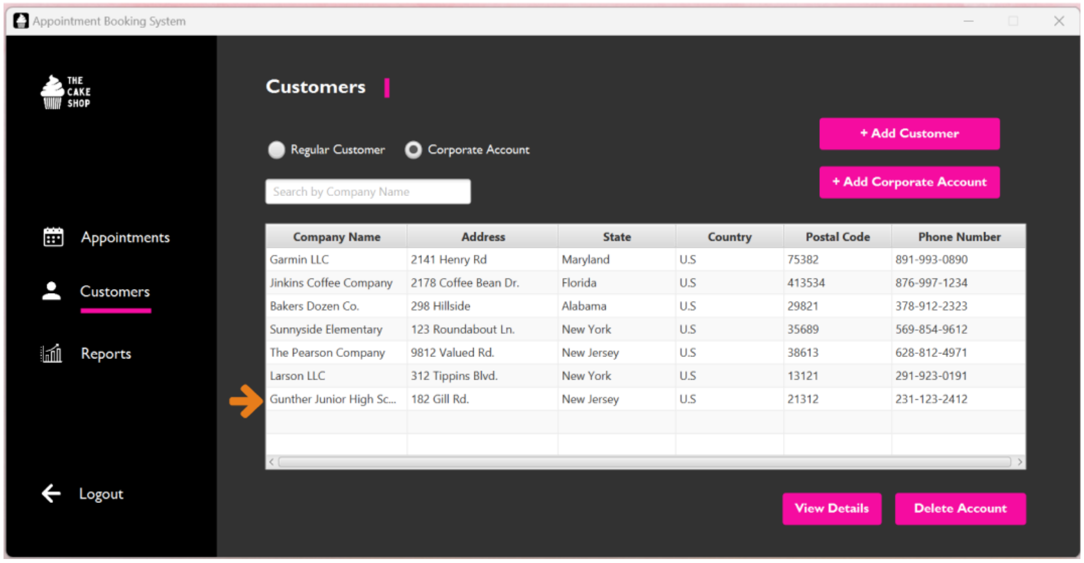
14. Click on the *Add Corporate Account* button near the top right of the screen to add a new corporate account. Once clicked, the *Add Corporate Account* form will load on another screen. Fill in the *Add Corporate Account* form with text in the input fields and select the desired option from the drop-down menus. Click the Save button to add the new corporate account. An alert will be displayed stating a new customer was added. The new corporate account will then be displayed on the Corporate Accounts table on the next screen. If invalid information is entered, input validation messages will be shown with additional directions.



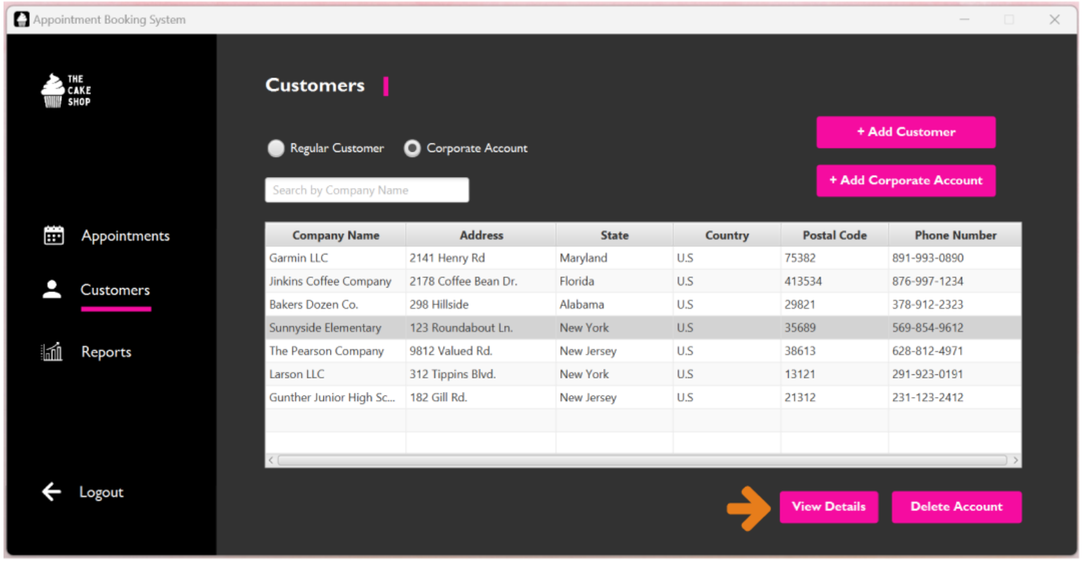


C. Application Design & Testing

Customers - Corporate Accounts

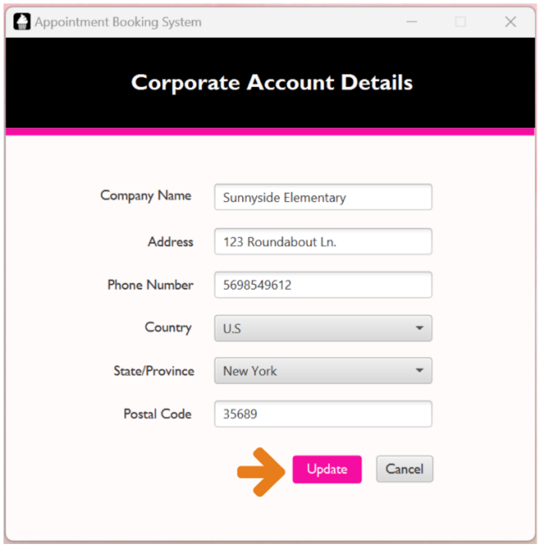


15. Select an account from the table and click the *View Details* button located under the Customer table to navigate to the Corporate Account Details screen where information on the selected corporate account can be viewed in its entirety. Make the desired changes and click the *Update* button to modify the account. Changes to the corporate account will be reflected on the table on the Corporate Accounts screen. If there are no changes to the account details, click on the *Cancel* button to be directed back to the Corporate Accounts screen.

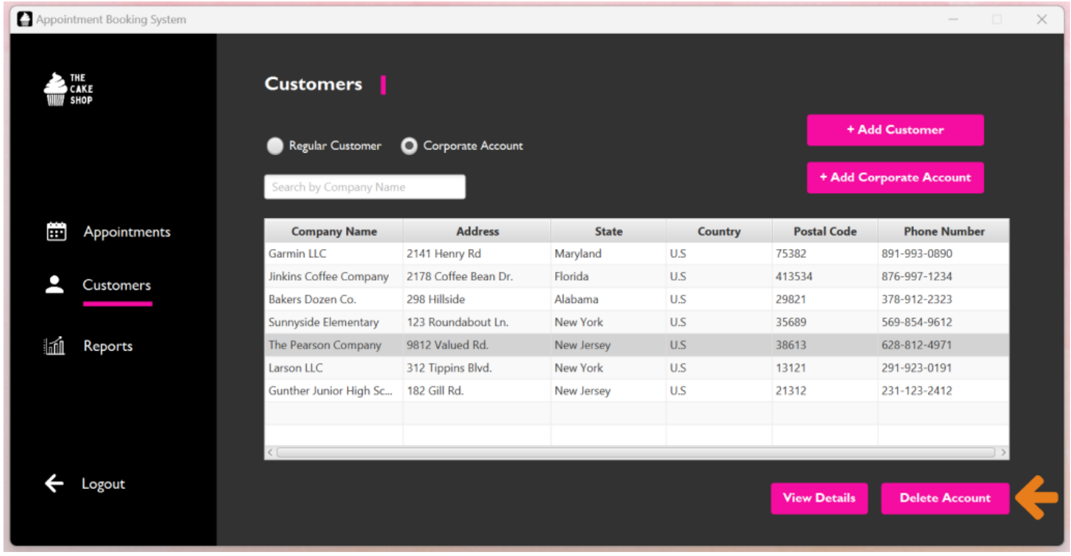


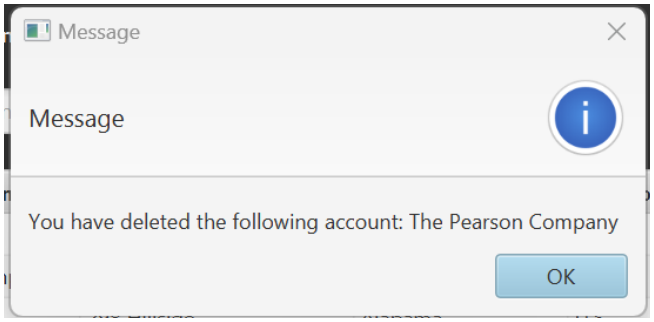
C. Application Design & Testing

Customers - Corporate Accounts



16. Select an account and click the *Delete Account* button located under the Corporate Account table to delete a selected corporate account. A confirmation message will be prompted. Click Yes to delete the corporate account. An alert will be displayed indicating that the selected account was deleted.

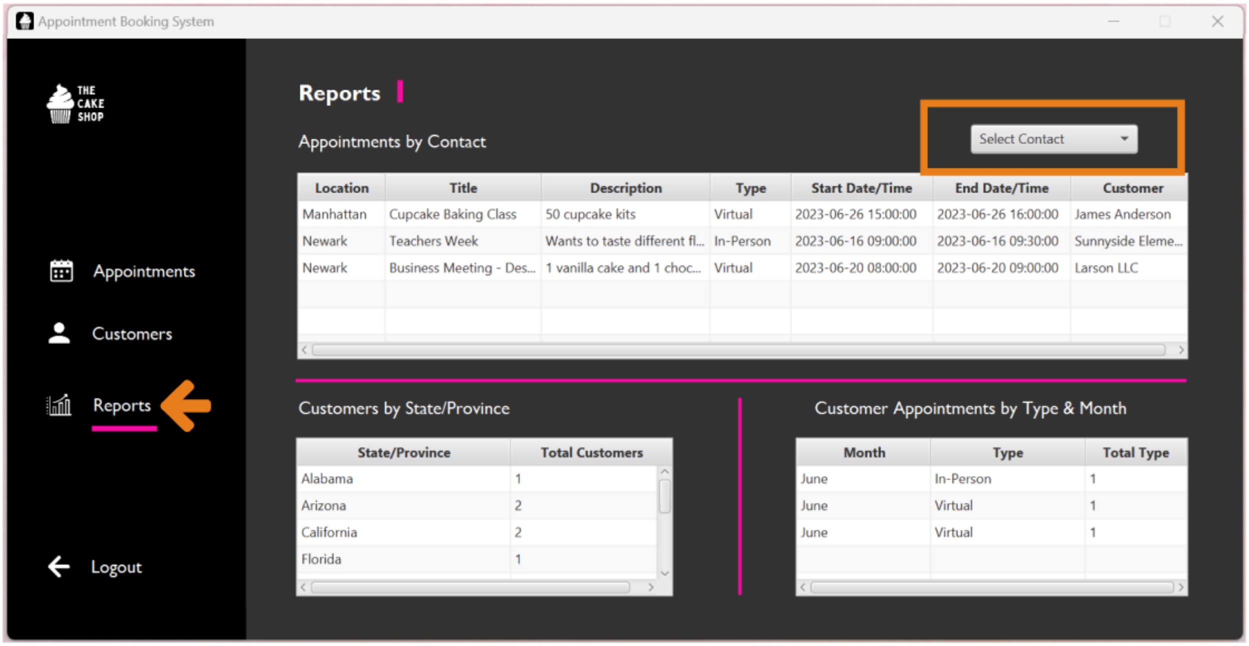




C. Application Design & Testing

Reports

17. Navigate to the *Reports* screen using the left panel navigation bar. This screen displays three reports in a structured table view. The first report, Appointments by Contact can be filtered per contact using the drop-down menu in the top right of the screen. The second (Customers by State/Province) and third (Customer Appointments by Type & Month) reports are displayed at the bottom of the screen.



D. Sources

Hoek, J. van der. (2023, January 24). The 5 stages of the Agile Software Development Lifecycle. Mendix. https://www.mendix.com/blog/agile-software-development-lifecycle-stages/

Joseph, F. (2021, December 6). Understanding java inheritance and polymorphism. Section. https://www.section.io/engineering-education/understanding-java-inheritance-and-polymorphism/

PractiTest. (n.d.). Black Box vs White Box Testing. PractiTest. https://www.practitest.com/qa-learningcenter/resources/black-box-vs-white-box-testing/#:~:text=The%20Black%20Box%20Test%20is,is%20carried%20out%20by%20testers.

Scrum for one: Adapting agile scrum methodology for individuals. Lucidchart. (2020, July 22). Retrieved April 6, 2023, from https://www.lucidchart.com/blog/scrum-for-one#:~:text=Also%2C%20in%20adapting%20Agile%20for,sprint%20to%20make%20this%20easier.

Photo Source:
Login - Photo by Valeria Boltneva: https://www.pexels.com/photo/cake-with-raspberry-toppings-827516/