**C868 – Task 2 – Section C**

|  |  |
| --- | --- |
| **Capstone Proposal Project Name:** | http://www.idevnews.com/views/images/uploads/general/wgu_logo.png  Project Punctual |
| **Student Name:** | Richard Menz |

**Table of Contents**

Contents

[**Table of Contents** 2](#_Toc71146530)

[Application Design and Testing 4](#_Toc71146531)

[Design Documentation 4](#_Toc71146532)

[Class Design 4](#_Toc71146533)

[UI Design 11](#_Toc71146534)

[Unit Test Plan 17](#_Toc71146535)

[Introduction 17](#_Toc71146536)

[Purpose 17](#_Toc71146537)

[Overview 23](#_Toc71146538)

[Test Plan 24](#_Toc71146539)

[Items 24](#_Toc71146540)

[Features/Functions 24](#_Toc71146541)

[Deliverables 24](#_Toc71146542)

[Tasks 24](#_Toc71146543)

[Needs 25](#_Toc71146544)

[Pass and Fail Criteria 25](#_Toc71146545)

[Source Code 25](#_Toc71146546)

[User Guide 26](#_Toc71146547)

[*Introduction* 26](#_Toc71146548)

[*Installation and Using the Application* 26](#_Toc71146549)

[*Logging In* 30](#_Toc71146550)

[*Navigation-Main Menu* 31](#_Toc71146551)

[*Customers records screen* 32](#_Toc71146552)

[*Appointments Screen* 35](#_Toc71146553)

[*Add/Edit Appointment screen* 36](#_Toc71146554)

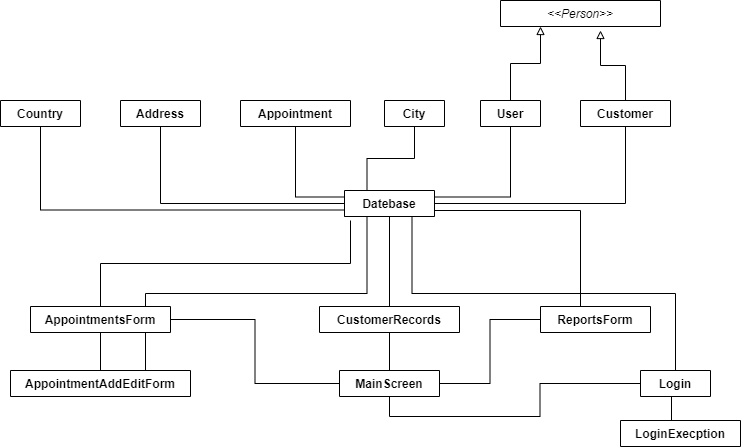
[*Reports Screen* 38](#_Toc71146555)

# Application Design and Testing

# Design Documentation

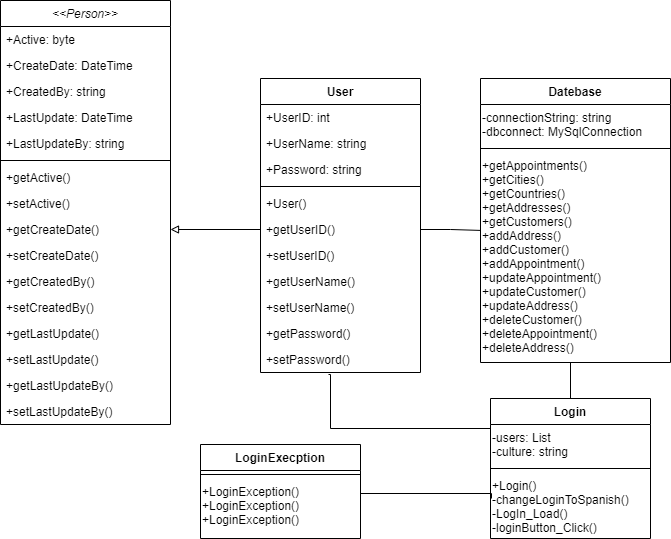
## Class Design

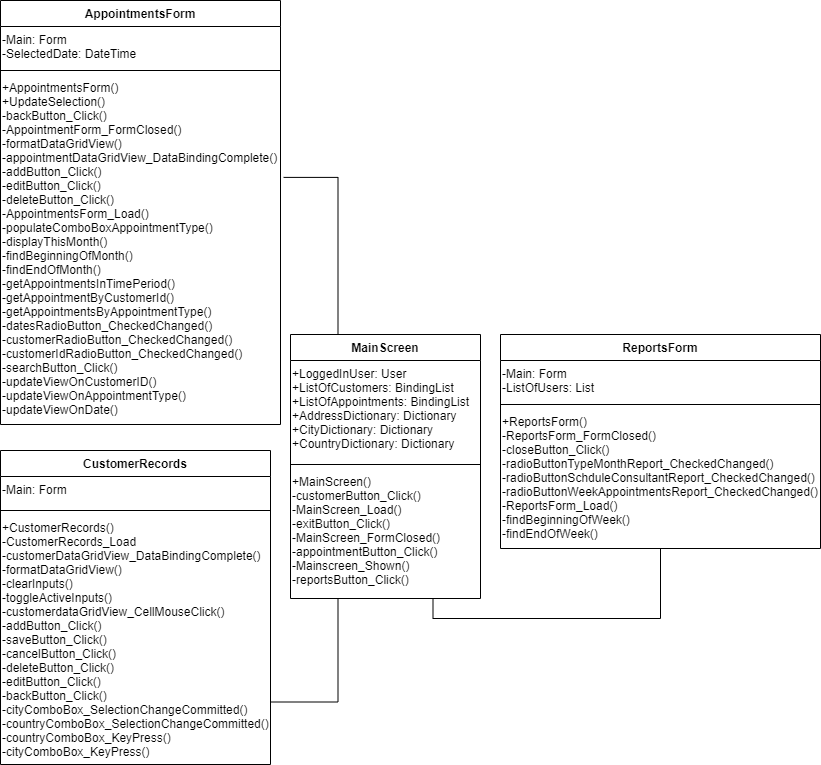
Punctual, the customer scheduling software, will contain fifteen classes, one of which is an abstract class. Of the fifteen classes included seven of them are models to be used as either a base class or by the database class to gather and store within the application Punctual. The other eight classes are either used as an UI element or to manipulate the data calling the database class to store the changes made. See the diagrams listed below for the reference diagram and related data points. The fifteen classes in the program are Address, Appointment, City, Country, Customer, Person, User, AppointmentAddEditForm, AppointmentsForm, CustomerRecords, Database, Login, LoginExpections, MainScreen, and ReportsForm.

Person is used as a parent class of Customer and User; these two classes have many differences, but both have five properties they share. Both classes have an indicator if they are active. Besides that, they have information about who created them, when they were created, when they were last updated, and who did that update. They share the base class for possible future reports client request can both take in Customer and User without modification. This polymorphic nature will help with generated reports on shared properties. How the user and customer class differ is the user has login information to get into the application while the customer has address information to be able to be contacted. The functionality to add new users from the application or edit users is not necessary for the Punctual application for now. This will be handled by the client IT staff most likely the database Administrator. Below is an image of how all classes work together in a UML diagram. After that each section of the diagram goes into more detail and is explained.

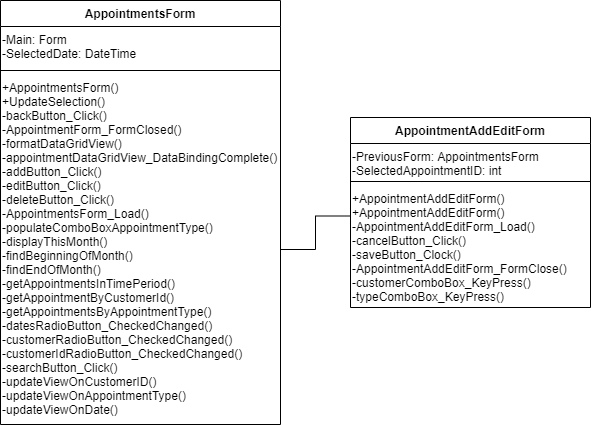
From what you can see from above the program has a complete path from starting the login which then leads to the main screen. From the main screen you have the three main UI elements AppointmentsForm, CustomerRecords, and ReportsForm. All of these make calls to the database which inputs the data into the modules shown above it so it can be used more easily by the UI sections of the program.

The start of the application is from the login. From the diagram below you can see the login form you have users and culture property to make it possible to update the UI in the proper language and have a variable user to verify off to make sure someone that should have access is accessing the application. A separate login expectation class was made to handle expectations of the class which is placed in LoginExpection.

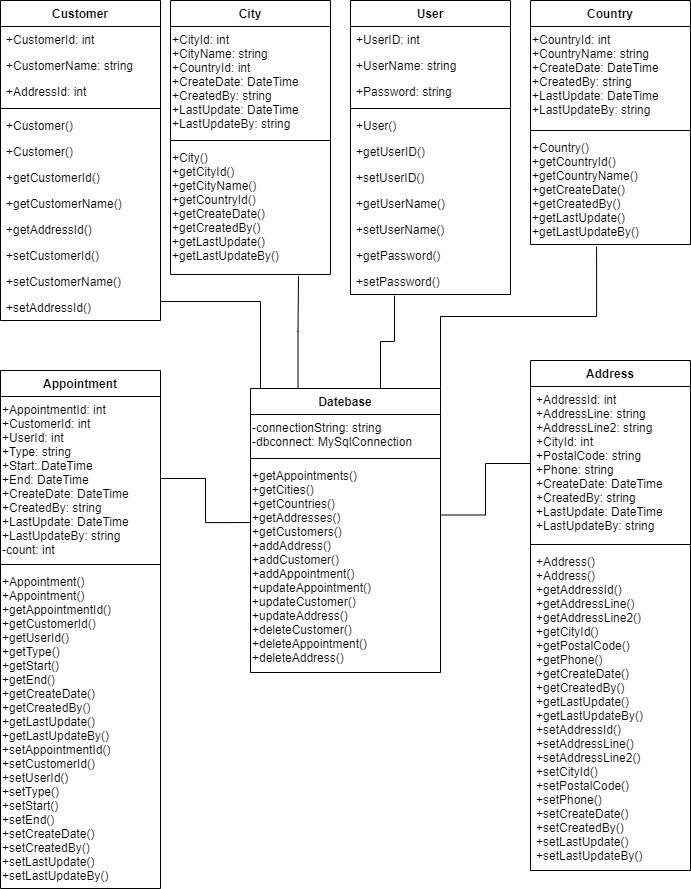


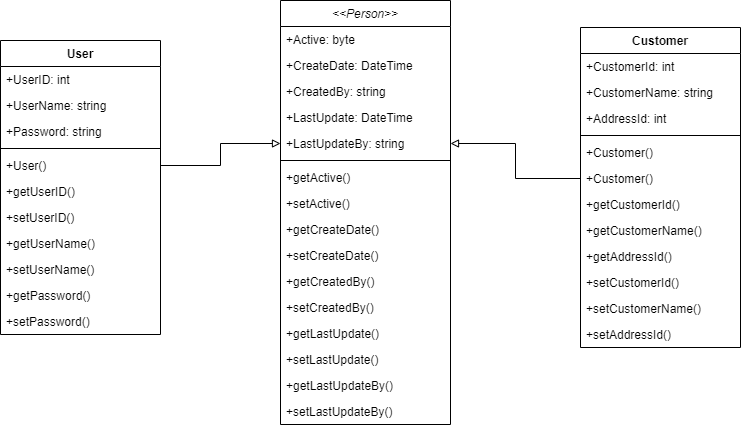
Once the user is signed in, they have access to the main screen class. From this class is a bridge to the other UI classes in the diagram. From the main screen you can get to the AppointmentsForm, CustomerForm, and ReportsForm. On the next section we will go into more detail on the AppointmentsForm. Using the methods customerButton\_Click and reportsButton\_Click makes you able to into the appropriate form to get the data one needs. While in these forms you can add, edit and delete customers from one and the other class is there to show per defended reports for the user.

For the appointment button it will bring you to the appointment form. This form also has the functionality of the pervious CustomerForm but operations are completed on appointments. The functionality of adding and editing appointment was moved to a separate AppointmentAddEditForm because the main AppointmentForm can search which if both on the page could cause overcrowding. The radio button classes for check change are in place to make it possible for user to do more then one type of search on the one form so they can find the information they need.



All the add, edits, and deletes from the form must be sent to the database class so the database can be updated with the new information. Besides being invoked by the main 3 forms AppointmentForm, CustomerForm, and ReportsForm it is directly used with the model classes above it. These models play a vital role to keep data being place in organized object while also making it simpler to update the information in the database from the object.

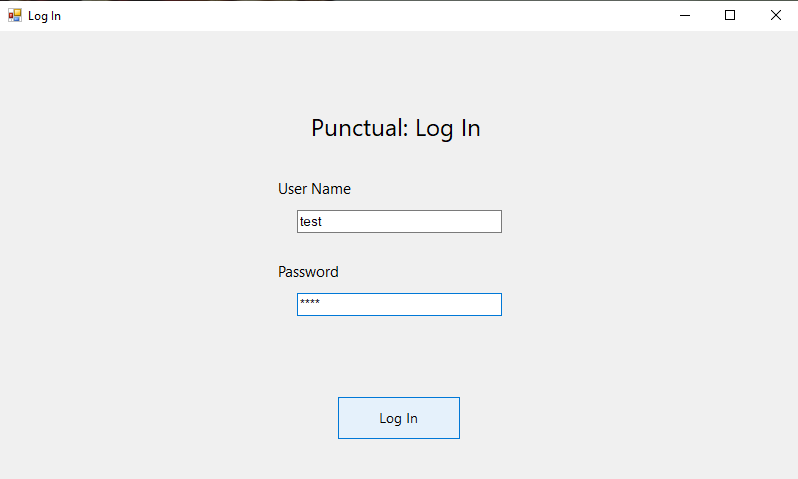


Lastly, we have two special models’ classes User and Customer. How these two are special is because they inherit from the abstract class Person. This is done because they share many sections of methods of code that can be handled similarly. Besides that, you can use both in place of a Person object polymorphically for the methods and properties they share. This is good because code can be used on what they have in common with Person class by passing in their respected objects and no other modification would be needed. Now any unique properties or methods cannot be used for each of these classes. Ergo a customer and user is a person but a user is not a customer.

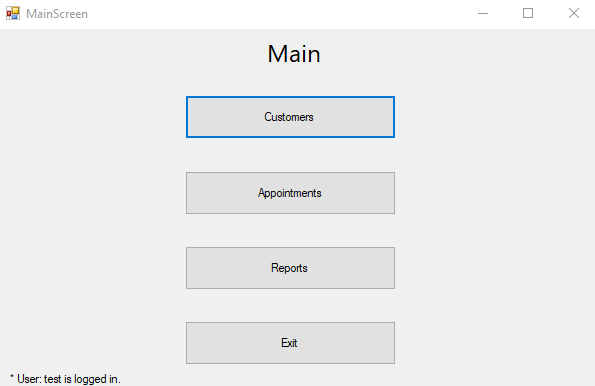
## UI Design

The design of the screen focus on simplicity because that is one of the most important parts of UI development. All screens are also like make it easier for users to navigate and use. Easy to use and intuitive controls are included to further bring out the power of the application. Listed below are high fidelity renders of the UI.

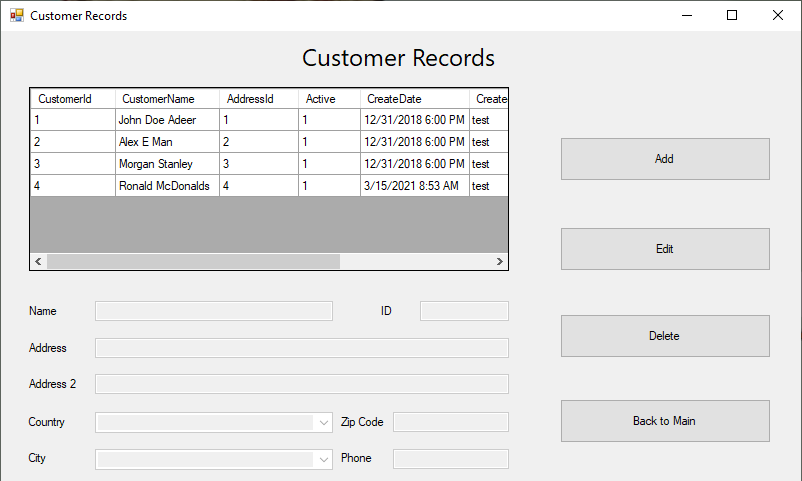
The first screen of the application is the login screen. This screen was designed to make is self-aware of what the user needs to do. The labels suggest the text needed for the text boxes. Once filled out the button signifies what the user needs to do which is login but clicking such button. Feedback is given directly to the user on the UI to inform them of invalid credentials.

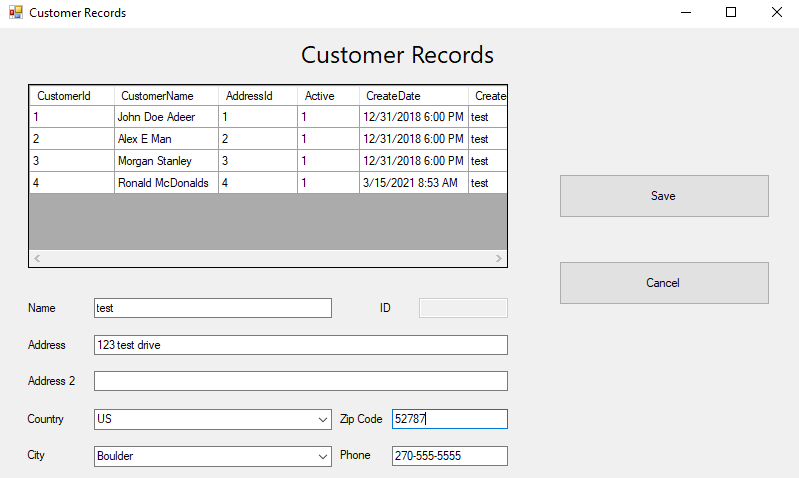


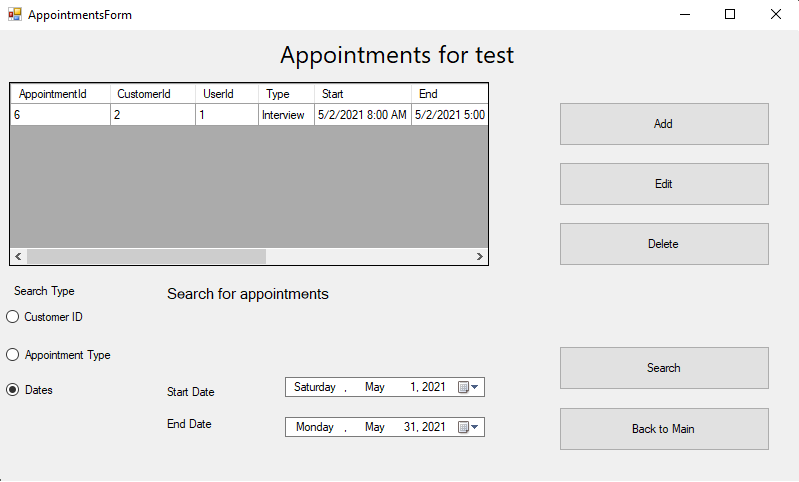
Once the user login they will be brought to the Main Screen which has three options available to navigate to different screens within the UI, and one option to exit the application. The first button will take the user to the customer records screen. The second button will bring them to the applications screen. The third button will take them to the reports screen.



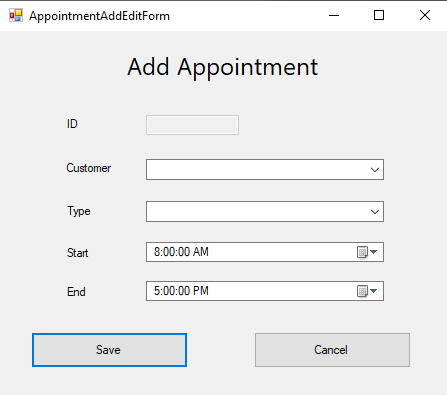
If the user selects the customer button, they will be brought to the customer records screen. It contains a large DataGrid View that will list all the customers. From this screen the user has the option to Add, Edit, and Delete customers. When a user selected the add or edit option the text boxes below will no longer be greyed out and be allowed to enter text to complete of adding a record. Once they put the information in, they can select the save button to save the record.

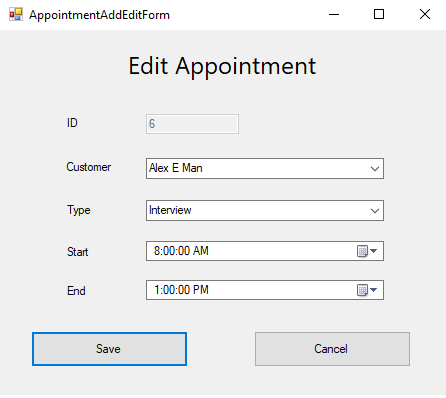




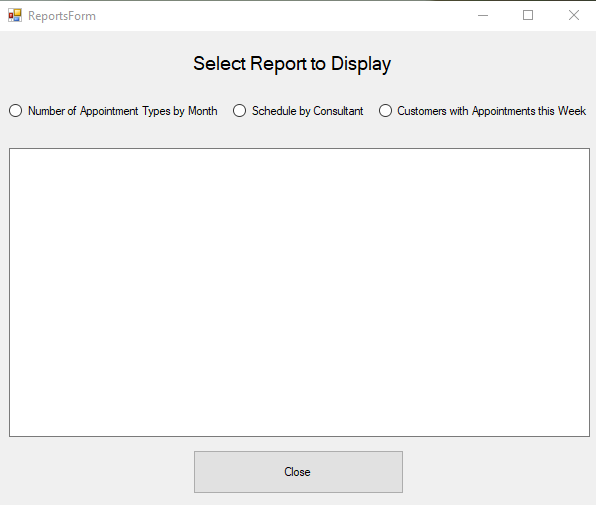
If the user selects the appointments button from the main screen, they will be brought to the appointments screen. This page contains a large DataGrid view where the results for any searches are displayed. On loading of the screen is will auto search all appointments this month. From this page they have the button options to add, edit, delete, and search for appointments. They also have a button back to main to go to the main page. The add and edit buttons bring up the add/edit appointment screen, while the delete button brings up a conformation box to delete the record. The user has three easy options on the left side for search that changes what fields are displayed they can search by.

The add and edit buttons from the appointments bring up the add/edit appointments but with slightly different displayed information. The add appointment will have blank information with labels to let the user know what to fill out but the edit will bring up a screen with the data already filled out where you can make changes. Both views of the same screen have a save option that the user can use to save and a cancel button that induvial bring the user back to the appointments page.





If the user selects the reports button from the main screen, they will be brought to the reports screen. On load they will see a big blank text back without being filled out. If they click on the three radio button options it will populate with a report with a title, date time stamp, and data relevant to the title of the button in multiple rows and columns. The user will have a intuitive button name close which will bring them back to the main screen.



# Unit Test Plan

## Introduction

### Purpose

Most of the testing done on this application was though the user interface manually. The test complete was to check for data validation, error handling, and base functionally. Each screen was tested individual as a unit before being added into the application. Once all forums have been made and added to the application, a final test as done to make sure all units worked together without any observable issues. Below are details for each test done on a unit. Any errors found were noted and given to the development team to fix. Once all updates have been made. One more test was performed to make sure everything functions together without issues. No errors were found during the finial full test turn.

**Login Screen**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| **Test number** | **Test Description** | **Test Data** | **Result Expected** | **Result Actual** | **Pass or Fail** |
| 1 | Login with test account | User: test  Password: test | Login Success | Login Success | Pass |
| 2 | Login with null entry | User: null  Password: null | Login Failed  Default error message | Login Failed  Default error message | Pass |
| 3 | Login with typo | User: test Password: test1 | Login Failed  Default error message | Login Failed  Default error message | Pass |

**Main Screen**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| **Test number** | **Test Description** | **Test Data** | **Result Expected** | **Result Actual** | **Pass or Fail** |
| 1 | Exit Button | None | Program closes self | Program closes self | Pass |
| 2 | Customer Button | None | Customer Records screen opens | Customer Records screen opens | Pass |
| 3 | Appointments Button | None | Appointments screen opens | Appointments screen opens | Pass |
| 4 | Reports Button | None | Reports screen opens | Reports screen opens | Pass |

**Customer Records Screen**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| **Test number** | **Test Description** | **Test Data** | **Result Expected** | **Result Actual** | **Pass or Fail** |
| 1 | Back to Main Button | None | Return to Main Screen | Return to Main Screen | Pass |
| 2 | Add Button | None | Let you enter data in the fields below | Let you enter data in the fields below | Pass |
| 3 | Save a real entry | Name: Ronald Mcdonaldsssss  Address: 888 BigMac Drive  Address 2: French Fry Cove  Country: Canada Zip Code: 97458  City: Sanata Barbara  Phone: 32152365898 | Add user into database with listed information | Add user into database with listed information | Pass |
| 4 | Save entry missing name | Name:  Address: 888 BigMac Drive  Address 2: French Fry Cove  Country: Canada Zip Code: 97458  City: Sanata Barbara  Phone: 32152365898 | Error popup informing you must enter customer name | Error popup informing you must enter customer name | Pass |
| 5 | Save entry missing address | Name: Ronald Mcdonaldsssss  Address:  Address 2: French Fry Cove  Country: Canada Zip Code: 97458  City: Sanata Barbara  Phone: 32152365898 | Error popup informing you must enter address | Error popup informing you must enter address | Pass |
| 6 | Save entry missing country | Name: Ronald Mcdonaldsssss  Address: 888 BigMac Drive  Address 2: French Fry Cove  Country: Zip Code: 97458  City: Sanata Barbara  Phone: 32152365898 | Error popup informing you must enter country | Error popup informing you must enter country | Pass |
| 7 | Save entry missing City | Name: Ronald Mcdonaldsssss  Address: 888 BigMac Drive  Address 2: French Fry Cove  Country: Canada Zip Code: 97458  City:  Phone: 32152365898 | Error popup informing you must enter City | Error popup informing you must enter City | Pass |
| 8 | Save entry missing zip code | Name: Ronald Mcdonaldsssss  Address: 888 BigMac Drive  Address 2: French Fry Cove  Country: Canada Zip Code:  City: Sanata Barbara  Phone: 32152365898 | Error popup informing you must enter zip code | Error popup informing you must enter zip code | Pass |
| 9 | Save entry missing phone | Name: Ronald Mcdonaldsssss  Address: 888 BigMac Drive  Address 2: French Fry Cove  Country: Canada Zip Code: 97458  City: Sanata Barbara  Phone: | Error popup informing you must enter missing phone | Error popup informing you must enter missing phone | Pass |
| 10 | Cancel button pressed | None | Return to showing add, edit, delete, back to main buttons | Return to showing add, edit, delete, back to main buttons | Pass |
| 11 | Click Edit Button without record select | None | Popup you must select a customer to edit | Popup you must select a customer to edit | Pass |
| 12 | While editing record try to save with missing information | Any information missing from test above | Error popup informing you must enter missing data of the missing type | Error popup informing you must enter missing data of the missing type | Pass |
| 13 | While editing record try to save with updated information | Name: Ronald Mcdonalds  Address: 888 BigMac Drive  Address 2: French Fry Cove  Country: Canada Zip Code: 97458  City: Sanata Barbara  Phone: 32152365898 | Record update with new information | Record update with new information | Pass |
| 14 | Delete button pressed with no record selected | None | Error popup saying you need a record selected | Error popup saying you need a record selected | Pass |
| 15 | Delete button pressed with a record selected | CustomerID 5 record selected | Bring up confirm box to delete record | Bring up confirm box to delete record | Pass |
| 16 | Delete confirm box press no | CustomerID 5 record selected | Return back to customers records screen | Return back to customer records screen | Pass |
| 17 | Delete confirm box press yes | CustomerID 5 record selected | Delete record for customerID 5 | Delete record for customerID 5 | Pass |

**Appointments Screen**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| **Test number** | **Test Description** | **Test Data** | **Result Expected** | **Result Actual** | **Pass or Fail** |
| 1 | Back to Main Button pressed | None | Return to Main Screen | Return to Main Screen | Pass |
| 2 | Add Button pressed | None | Bring up Add Appointment Screen | Bring up Add Appointment Screen | Pass |
| 3 | Edit Button pressed with no appointment selected | None | Popup informing a record needs to be selected | Popup informing a record needs to be selected | Pass |
| 4 | Edit Button pressed with record selected | A record selected | Bring up edit appointment screen | Bring up edit appointment screen | Pass |
| 5 | Delete Button pressed with no record selected | None | Popup informing a record needed to be selected | Popup informing a record needed to be selected | Pass |
| 6 | Delete button pressed with record selected | Appointment with ID 7 selected | Popup asking if you want to delete yes or no | Popup asking if you want to delete yes or no | Pass |
| 7 | Select no on delete pop up | Appointment with ID 7 selected | Return back to appoints screen without deleting record | Return back to appoints screen without deleting record | Pass |
| 8 | Select yes on delete pop up | Appointment with ID 7 selected | Delete record with appointmentID 7 | Delete record with appointmentID 7 | Pass |
| 9 | Click radio button Dates | None | Show a search that you can search by date | Show a search that you can search by date | Pass |
| 10 | Search a valid range of dates | Start date of start of the month and end date of end of the month | Show all appointments for the month | Show all appointments for the month | Pass |
| 11 | Search a range of dates without appointments | Searched from May 1 to May 1 | Show no records on screen | Show no records on screen | Pass |
| 12 | Search a date range that is not valid | Searched from May 1 to April 1 | Popup information saying start date is later than end date | Popup information saying start date is later than end date | Pass |
| 13 | Click radio button Appointment Type | None | Show a search that you can search by Appointment Type | Show a search that you can search by Appointment Type | Pass |
| 14 | Search by appointment type of appoints that have that type | Select Lunch | Show all lunch results | Show all lunch results | Pass |
| 15 | Search by appointment type that doesn’t have any appointments | Select Scrum | Show a popup non appointments of selected type | Show a popup non appointments of selected type | Pass |
| 16 | Search by appointment type not listed | Put nothing into combo box | Show a popup non appointments of selected type | Show a popup non appointments of selected type | Pass |
| 17 | Select Customer Id radio Button | None | Show a search that you can search by customer ID | Show a search that you can search by customer ID | Pass |
| 18 | Search with a valid customer ID with appointments | 2 | Show all appointments for customer with that ID | Show all appointments for customer with that ID | Pass |
| 19 | Search with a customer ID 0 | 0 | Popup saying enter a value greater than 0 | Popup saying enter a value greater than 0 | Pass |
| 20 | Search with a customer ID that isn’t a number | A | Popup saying enter a value greater than 0 | Popup saying enter a value greater than 0 | Pass |

**Add Appointment Screen**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| **Test number** | **Test Description** | **Test Data** | **Result Expected** | **Result Actual** | **Pass or Fail** |
| 1 | Cancel button | None | Return to the Appointments Screen | Return to the Appointments Screen | Pass |
| 2 | Saved data with valid information | Customer: John Doe Adeer  Type: Presentation  Start: 8 AM today  End: 5 PM today | Create appointment with information provided | Create appointment with information provided | Pass |
| 3 | Save data missing customer | Customer  Type: Presentation  Start: 8 AM today  End: 5 PM today | Popup a customer must be selected | Popup a customer must be selected | Pass |
| 4 | Save data missing type | Customer: John Doe Adeer  Type:  Start: 8 AM today  End: 5 PM today | Popup a type must be selected | Popup a type must be selected | Pass |
| 5 | Save data appointment overlapping another appointment | Make same appointment as test 2 Customer: John Doe Adeer  Type: Presentation  Start: 8 AM today  End: 5 PM today | Popup saying you cannot have overlapping appointments | Popup saying you cannot have overlapping appointments | Pass |

**Edit Appointment Screen**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| **Test number** | **Test Description** | **Test Data** | **Result Expected** | **Result Actual** | **Pass or Fail** |
| 1 | Cancel button pressed | None | Return to the appointments screen | Return to the appointments screen | Pass |
| 2 | Change data to different valid information | Change appoint ID 6 customer from Alex E Man to John Doe Adeer | Update appointment with new customer ID | Update appointment with new customer ID | Pass |
| 3 | Change date to be overlapping with a different appointment | Change start and end time to be 8 AM to 5 PM 05/02/2021 | Popup information you cannot have overlapping appointments | Popup information you cannot have overlapping appointments | Pass |

**Reports Screen**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| **Test number** | **Test Description** | **Test Data** | **Result Expected** | **Result Actual** | **Pass or Fail** |
| 1 | Close button | None | Return to the Main Screen | Return to the Main Screen | Pass |
| 2 | Select Number of Appointment Type by Month Radio Button | None | Show Report of appoints by type for the previous, current, and next months | Show Report of appoints by type for the previous, current, and next months | Pass |
| 3 | Select Schedule by Consultant Radio Button | None | Show report of current months appoints for each consultant. | Show report of current months appoints for each consultant. | Pass |
| 4 | Select Customers with Appointments this Week Radio Button | None | Show report of customers with appoints this week with contact information. | Show report of customers with appoints this week with contact information. | Pass |

### Overview

Testing was done by the team for each of its units, with each set of the units added to the core application a finial core test was complete. Each individual section of the application was tested in view/controller pairs as each one was considered one unit. All errors where documents in the main bug report. After reported it was assigned to specialized team members from the development team to resolve the issue. All test were done within the user interface.

## Test Plan

### Items

* Visual Studio 2019
* .Net Framework 4.7.2
* Access to test account to access the Application

### Features/Functions

The functions which were tested where all methods from all controllers. The tests were complete using the application User Interface (UI). A complete use of the interface was done to completely test the methods. Date integrity was examined from the tests because validation of the data was done from the controllers. Also, any handling of errors and messages come from the controller interface from the application as well which was part of the test plan.

### Deliverables

The test completed in the tables above will show the pass and fail mark for each test completed. It will also give information of what type if test it was, what data was used, what results we expected to get, and the observed results of each test. The data was added to the document for referencing by the client.

### Tasks

1. Do the test case.
2. Log test failures and assign to the special members of the development team for remediation.
3. Complete test cases.
4. Repeat steps 1-3 until all test cases are passed.

### Needs

To complete the test there will be a need for a windows 10 environment of any version. The machine will also require visual studio 2019 and .Net framework 4.7.2 installed. The machine will need access to a network connection to reach the MySQL database.

### Pass and Fail Criteria

The tables above with the test plan will account for the pass and fail criteria within the actual results and expected results column. A pass will be recorded if both columns match. A fail will mean the actual and expected results did not match. Discrepancies will be reported in the bug report log and then assigned to the special members inside the development team to resolve.

# Source Code

The source code will be attached into the submission portal on WGU website.

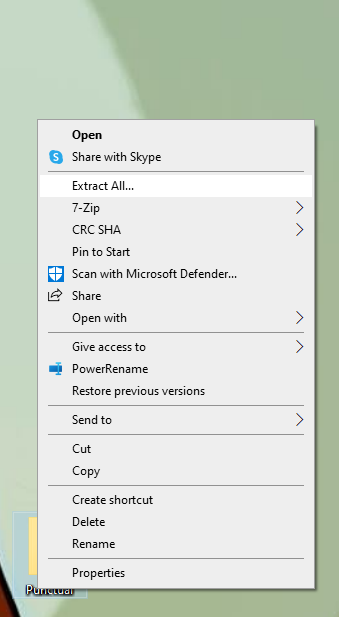
# User Guide

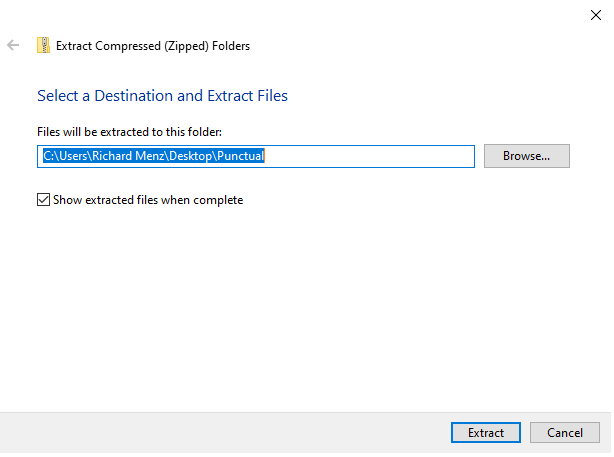
## *Introduction*

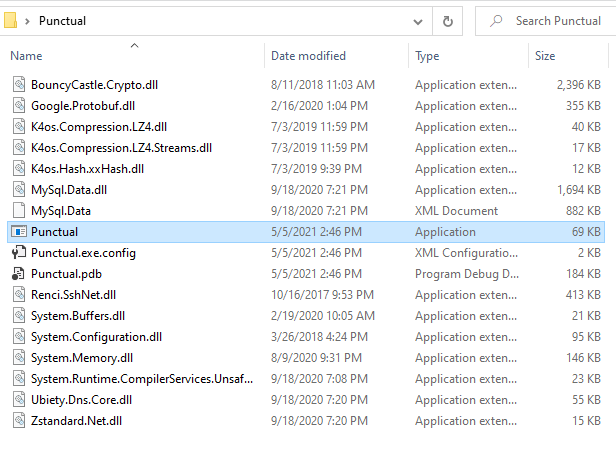
This is the guide for users of the Punctual Application. Punctual is a lightweight C# application designed for minimal backend setup and minimum install steps to get running. Follow the instruction for how to setup and use Punctual Application.

## *Installation and Using the Application*

1. We need to make sure the client machine meets requirements. Machine needs to be running any version of Windows 10 OS. Also, the .net runtime may need to be installed but most machine will have this installed for other windows applications. The administration team that will add users into the application will need MySQL Workbench, but it is not needed for a baseline application.
2. The application will be sent in a .zip format. Right click the zip file and click on extract all.

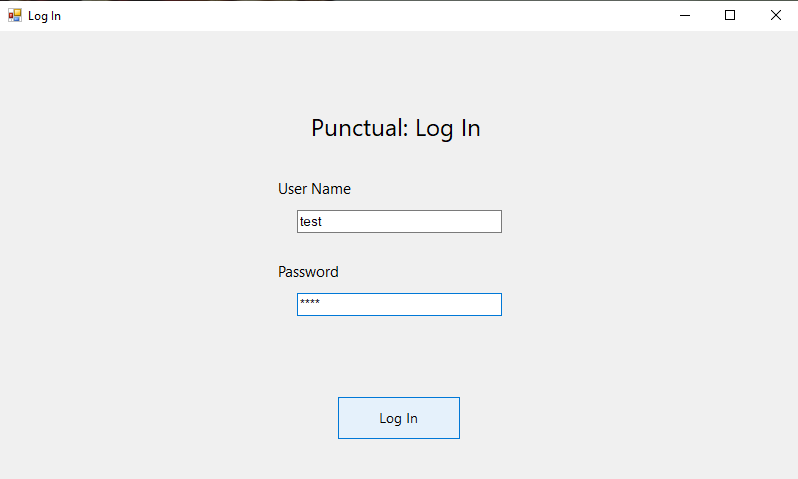


1. The extract dialog will come up. Pick the location you want to extract the file.
2. Open the folder Punctual you extracted and double click the application named Punctual.

**

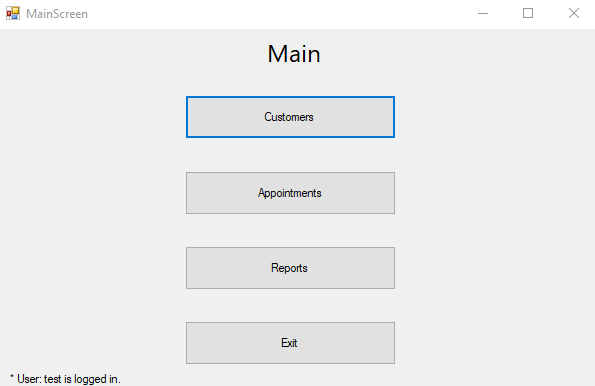
## *Logging In*

1. You will need to fill in the text field with the username and password.
   1. **For the test the username will be ‘test’ and the password will be ‘test’. You will enter these values without the quotation marks.**
2. Click login button this will navigate you to the Main screen.

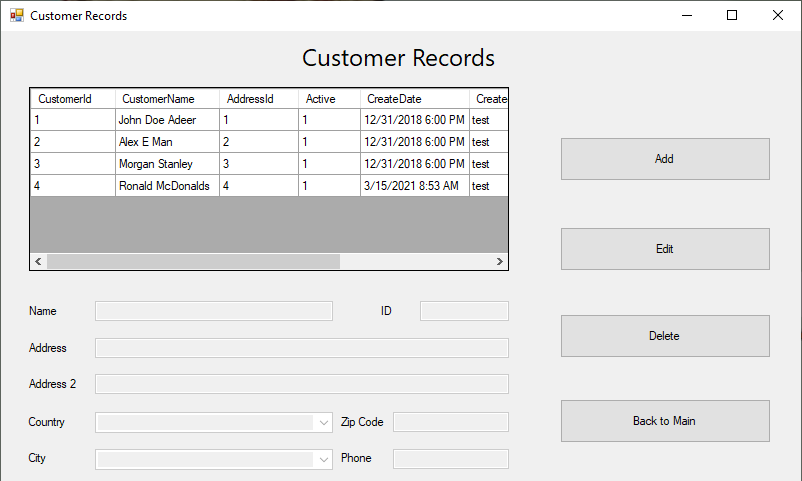
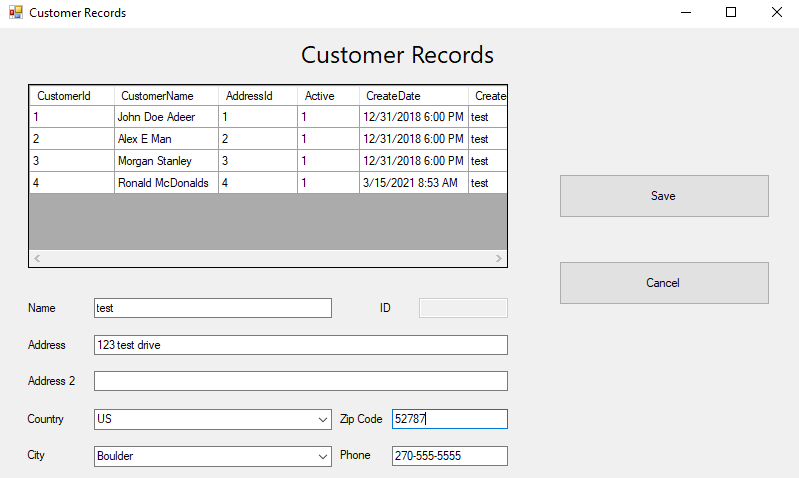
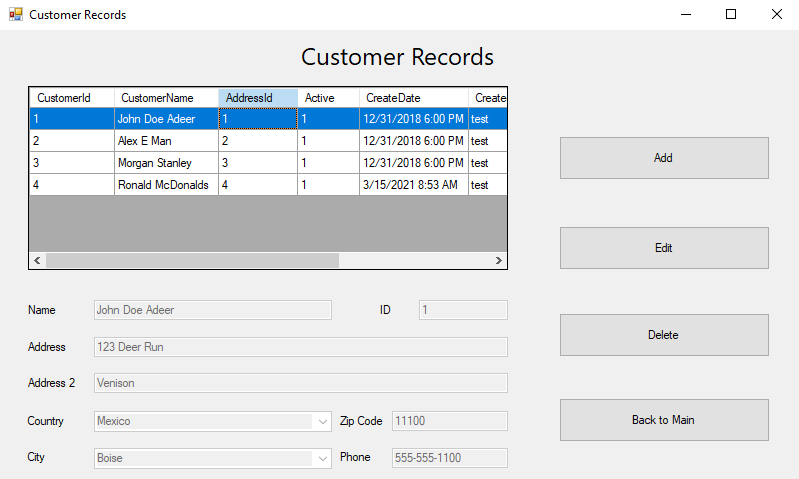
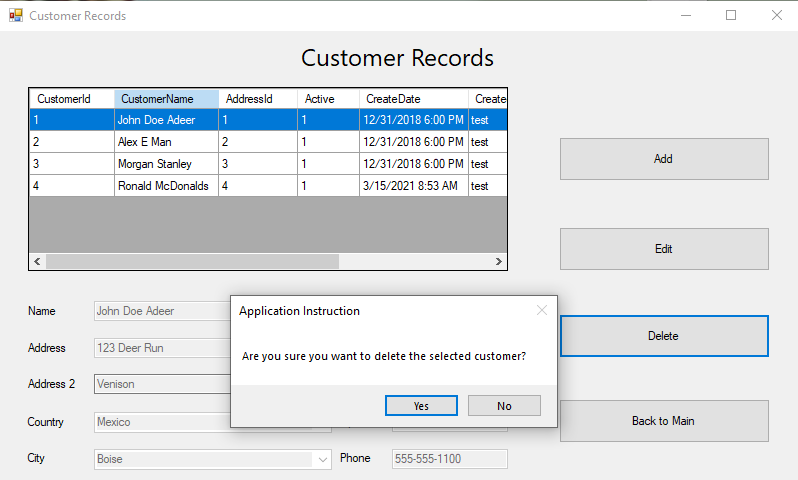


## *Navigation-Main Menu*

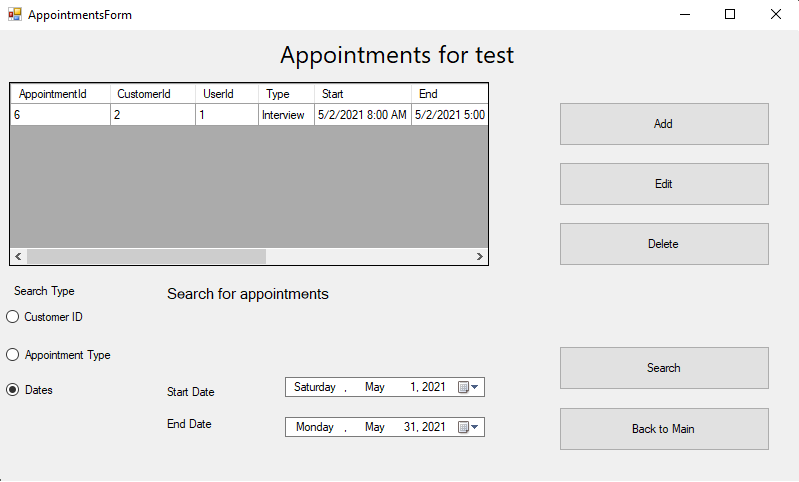
1. You can press exit button from the bottom middle of the page to exit the application. You can also hi the x at the top right.
2. You can press the customers button to open the customer records screen.
3. You can press the appointments button to open the appointments screen
4. You can press the reports button to open the reports screen.



## *Customers records screen*

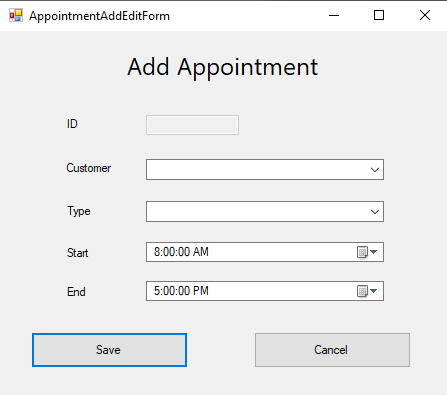
1. Click the Back to Main button to return to the Main Screen
2. Click the Add button to highlight the options below.
3. Click the save button will all fields completed. Address 2 field is option and the ID field will be generated by the system and is unable to be updated manually.
4. Click the cancel button if you do not want to make a new customer record.
5. When you have a customer, record selected by click on it click the edit button to edit the record. It will highlight the options below like when adding a customer so you can change the data. 
6. To delete customer when you have a record selected click the delete button. It will bring up a confirmation dialog. Hit yes to delete the customer and hit no if you did not want to delete that customer.

## *Appointments Screen*

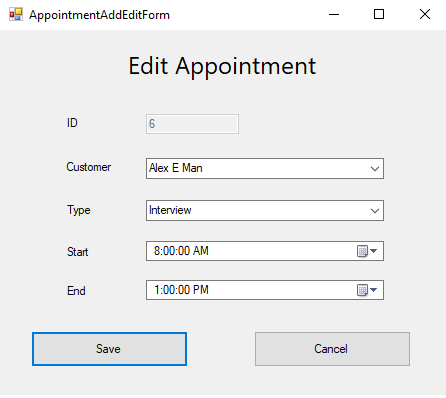
1. Click the Back to Main button to return to the Main Screen
2. As you can see this page has similar options to the customer records page but also includes a search button. The Add and Edit button function similarly to customer records page but opens the add/edit appointment dialog. Look further down in the user guide for information on that screen.
   1. Don’t forget to edit an appointment you must have it selected.
3. When the forum opens it defaults to a search of dates for the start and end of this month as shown in step 1 image.
4. Under search types you have three options Customer ID, Appointment Type, and Dates. Each radio button brings up different boxes each with different data you can put in. Once you put in valid data hit the search option and, in the list, box above will list the appointments that match criteria.

## *Add/Edit Appointment screen*

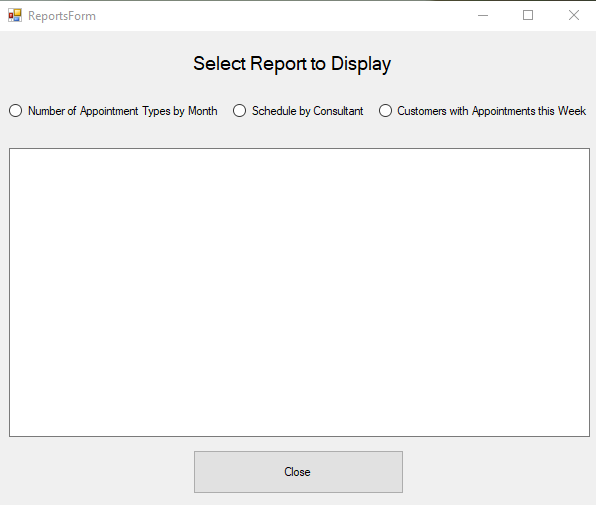
1. Click the cancel button to return to the appointments screen.
2. Enter a customer, type, start, and end date then click the save button to save a new appointment.



1. The Edit appointment dialog has similar functionally but auto populates with the appointment’s current information.



## *Reports Screen*

1. Click the close button will return you back to the main screen.
2. To generate a report, click on the radio button of the report you want to be generated. The report will have a title at the top, follow by a datetime stamp, and lastly will generate a report with multiple rows and columns of information inside the txt box.