**C868 – Software Capstone Project Summary**

**Task 2 – Section C**

|  |  |
| --- | --- |
| **Capstone Proposal Project Name:** | http://www.idevnews.com/views/images/uploads/general/wgu_logo.png  Java Scheduling Desktop Application |
| **Student Name:** | Brandon Rider |

Contents

[Task 2 Part C – C868 Software Development Capstone 4](#_Toc90667388)

[Application Design and Testing 4](#_Toc90667389)

[Design Document 4](#_Toc90667390)

[Class Design 4](#_Toc90667391)

[UI Design 5](#_Toc90667392)

[Unit Test Plan 7](#_Toc90667393)

[Introduction 7](#_Toc90667394)

[Purpose 7](#_Toc90667395)

[Overview 7](#_Toc90667396)

[Test Plan 7](#_Toc90667397)

[Items 7](#_Toc90667398)

[Features 7](#_Toc90667399)

[Deliverables 8](#_Toc90667400)

[Tasks 8](#_Toc90667401)

[Needs 8](#_Toc90667402)

[Pass/Fail Criteria 8](#_Toc90667403)

[Specifications 9](#_Toc90667404)

[Procedures 9](#_Toc90667405)

[Results 10](#_Toc90667406)

[Source Code 10](#_Toc90667407)

[User Guide 11](#_Toc90667408)

[Introduction 11](#_Toc90667409)

[Installation and Using the Application 11](#_Toc90667410)

[*Login* 12](#_Toc90667411)

[*Customers* 12](#_Toc90667412)

[*Update an Existing Customer* 12](#_Toc90667413)

[*Create a New Customer* 14](#_Toc90667414)

[*Delete an Existing Customer* 16](#_Toc90667415)

[*Appointments* 16](#_Toc90667416)

[*Update an Existing Appointment* 16](#_Toc90667417)

[*Create a New Appointment* 18](#_Toc90667418)

[*Delete an Existing Appointment* 19](#_Toc90667419)

[*Reports* 20](#_Toc90667420)

Task 2 Part C – C868 Software Development Capstone

# Application Design and Testing

# Design Document

## Class Design

This Entity Relationship Diagram shows how the database will be designed. It includes the relationships between each class, the fields contained in each class, and the methods available for use in each class. The design is built to improve efficiency and performance when the application acts on data from the database.Diagram

Description automatically generated

*Figure 1: Class Design*

## UI Design

The user interface was designed for ease of use and performance. I considered user experiences with previous applications and applied design standards to this application. The below designs are low fidelity showing the login page, appointments detail page, and custom reports detail page. I used these low-fidelity designs to code the outline of the user interfaces in XML using the scene builder tool.

Graphical user interface

Description automatically generated with medium confidence

*Figure 2: Low Fidelity – Login View*

Table

Description automatically generated

*Figure 3: Low Fidelity – Appointment List View*

*Graphical user interface, application

Description automatically generated*

*Figure 4: Low Fidelity – Reports View*

# Unit Test Plan

## Introduction

### Purpose

Unit testing was performed within this application by a manual method of testing. This allowed the developer to check the logic of specific units of code while using the interface like how the user will eventually use the application. Remediation testing was useful when my unit test failed and was easily retested by the same unit test once the specific code was revised.

### Overview

I created unit tests that ran concurrently with the program which created a system output notifying me of data and functionality. I chose to use custom code instead of a testing framework so I could keep the application performance at high efficiency. The tests were written to be event-driven to evaluate the specific functionality associated with the event.

## Test Plan

### Items

The tests are written into the functionality of the application and only manual click-through of the application is needed to evaluate the tests.

### Features

User Login feature including initialization, file output, and database connection functionalities.

Customer features include adding, deleting, and updating functionalities.

Appointment features include adding, deleting, and updating functionalities.

### Deliverables

The test results are written on the console window and can easily be printed in a text file.

They include the name of the functionality, test description, and a pass or fail rating for each specified test.

### Tasks

1. Verify that the testing code is live and not commented out.
2. Build and run the application in IntelliJ IDEA.
3. Open the console window in IntelliJ IDEA.
4. Navigate through each unit testing specified feature.
5. Confirm in the console window that testing results are written after each feature is used.
6. Print console log with test results to a pdf that can be provided to the client.

### Needs

Tests can be performed on any computer that meets the technical requirements to run the program. All that is needed is to run the application and then the tester navigates through the application. Each feature they navigate to will automatically run that features specified tests. There are no packages or libraries needed to perform the tests.

### Pass/Fail Criteria

Each test had an expected outcome. Once the test was executed if the specified outcome was the actual results then it was a successful test. If the expected results did not match the actual results, then the test result was a failure. Remediation included debugging the specific functionality that failed a test and retesting once satisfied with the code revisions. Failed tests during the development phase were recorded on a text document so tasks could be created for code improvements.

## Specifications

Below is a snippet of the testing code used for the database connection functionality of the User Login feature.

Graphical user interface, text, application

Description automatically generated

## Procedures

1. Determine which features and functionality would most benefit from unit testing.
2. Determine whether manual or automatic unit testing would be appropriate.
3. Design and write the test cases in pseudo-code to verify the logic.
4. Write the unit tests in code and implement them into the application code base.
5. Execute each unit test and verify results.
6. Evaluate the logic for any failed tests and revise tests as needed.
7. If the unit tests were not faulty, debug and revise the specific code for that functionality.
8. Retest unit tests for failed features.
9. Repeat steps 5-8 as necessary.

## Results

Below is a snippet of testing results for the User Login feature which includes the testing of the initialization, file output, and database connection functionality. Results show which parts of the specified functionalities passed or failed their written tests. Passes tests are in green and state “Pass” while failed tests are in red and state “Fail”.

Graphical user interface, text, application, email

Description automatically generated

# Source Code

The source code is included in the project package as a compressed file named “Scheduling-App-main”. Installation instructions (steps 1-5) in the provided User Guide give access to the entirety of the source code line by line.

# User Guide

## Introduction

These instructions will help you easily install, access, and operate the provided software application. Follow the below steps to learn and understand the full functionality and capabilities of the software. Each module after login has a back button that conveniently reverts to the menu page if needed.

## Installation and Using the Application

The following steps will install the application:

1. Navigate to the LabClient VM and log in.
2. Extract the compressed source code onto the Desktop.
3. Run the IntelliJ IDEA application
4. Click “File” from the top menu and then click “Open”.
5. Graphical user interface, text, application

   Description automatically generated
6. Navigate to the Desktop folder and click on the “Scheduling-App-main” folder so it is highlighted.
7. Graphical user interface, text

   Description automatically generated
8. Once the application loads in IntelliJ IDEA navigate to the “Run” menu item on the top of the page and click “Run”.
9. Graphical user interface, text, application, chat or text message

   Description automatically generated
10. Once compiled you can begin navigating the software application starting with the login page.

## *Login*

1. *Initial login will use the ‘test’ account credentials.*

*Graphical user interface, text, application

Description automatically generated*

1. *Enter ‘test’ into the User ID input box and ‘test’ into the Password input box.*
2. *Click the “Submit” button.*

## *Customers*

### *Update an Existing Customer*

1. *Once logged in, click on the existing Customer record you would like to update which will highlight the record.*

*Graphical user interface, application, table

Description automatically generated*

1. *Click on the “Update Existing” button.*

**

1. *Edit the desired information in the form by clicking in the desired input box and typing the new information.*

*Graphical user interface, text, application

Description automatically generated*

1. *To edit the Country or Division fields click on the drop-down arrow and then select the desired information.*

Graphical user interface

Description automatically generated with medium confidence

1. *Once the desired changes have been made click the “Update Existing Customer” button.*
2. **

### *Create a New Customer*

1. *Once logged in, click on the “Create New” button at the bottom of the Customer List page.*

*Graphical user interface, application, table

Description automatically generated*

1. *Add desired information into the input boxes by clicking on each box and adding text when highlighted. Note: Customer ID cannot be edited.*

*Graphical user interface, application

Description automatically generated*

1. *Add the Country and Division information by clicking on the drop-down arrows and selecting desired information.*

*A picture containing graphical user interface

Description automatically generated*

1. *Click the “Save New Customer” button to save the new customer record.*

**

*Graphical user interface, text, application

Description automatically generated*

1. *To edit the Country or Division fields click on the drop-down arrow and then select the desired information.*

Graphical user interface

Description automatically generated with medium confidence

1. *Once the desired changes have been made click the “Update Existing Customer” button.*
2. **

### *Delete an Existing Customer*

1. *Once logged in, click on the existing Customer record you would like to delete which will highlight the record.*

*Graphical user interface, application, table

Description automatically generated*

1. *Click on the “Delete Selected” button.*

**

## *Appointments*

### *Update an Existing Appointment*

1. *Once logged in and on the Customer List page click on the “Appointments” button.*

**

1. *Once logged in, click on the existing Appointment record you would like to update which will highlight the record.*

*Table

Description automatically generated*

1. *Click on the “Update Existing” button.*

**

1. *Edit the desired information in the form by clicking in the desired input box and typing the new information. Note: Appointment ID is not editable at this point.*

*Graphical user interface

Description automatically generated*

1. *To edit the Customer ID, User ID or Contact Name fields click on the drop-down arrow and then select the desired information.*

*Graphical user interface, text, application, chat or text message

Description automatically generated*

1. *Once the desired changes have been made click the “Update Existing Appointment” button.*
2. **

### *Create a New Appointment*

1. *Once logged in and on the Customer List page click on the “Appointments” button.*

**

1. *Add desired information into the input boxes by clicking on each box and adding text when highlighted.   
   Notes: Appointment ID cannot be edited and Format Example: 2020-05-29 22:05*

*Graphical user interface

Description automatically generated*

1. *Add the Customer ID, User ID, and Contact Name information by clicking on the drop-down arrows and selecting desired information.*

*Graphical user interface, text, application, chat or text message

Description automatically generated*

1. *Click the “Save New Appointment” button to save the new customer record.*

**

### *Delete an Existing Appointment*

1. *Once logged in, click on the existing Appointment record you would like to delete which will highlight the record.*

*Table

Description automatically generated*

1. *Click on the “Delete Selected” button.*

**

## *Reports*

1. *To see reports when currently viewing the Customer List or Appointment List, click on the "Reports" button at the bottom of the page.*

**

1. *The default Reports view provides 3 custom reports to choose from, click on the "Report: Total Appointments", "Report: Contact Schedule", or "Report: Total Customers Per Country" button depending on which report you desire to view.*

**

1. *The clicked report will be highlighted and a table with the specified report will appear below the buttons. To view the other reports, click either of the previously unclicked report buttons.*

*Table

Description automatically generated*