**C868 – Software Capstone Project Summary**

**Task 2 – Section C**

|  |  |
| --- | --- |
| **Capstone Proposal Project Name:** | http://www.idevnews.com/views/images/uploads/general/wgu_logo.png  Software Agency Scheduling Platform |
| **Student Name:** | Jamey Mathis |

**Table of Contents**

[Table of Contents 2](#_Toc576602919)

[Task 2 Part C – C868 Software Development Capstone 4](#_Toc467403781)

[Application Design and Testing 5](#_Toc676187946)

[Design Document 5](#_Toc2048158341)

[Class Design 5](#_Toc838222176)

[UI Design 5](#_Toc418511064)

[Unit Test Plan 6](#_Toc825999313)

[Introduction 6](#_Toc2105200894)

[Purpose 6](#_Toc1818634141)

[Overview 6](#_Toc28100427)

[Test Plan 6](#_Toc1415861512)

[Items 6](#_Toc1942214386)

[Features 6](#_Toc93620575)

[Deliverables 6](#_Toc126651531)

[Tasks 7](#_Toc1994590404)

[Needs 7](#_Toc215889833)

[Pass/Fail Criteria 7](#_Toc870849184)

[Specifications 7](#_Toc650512378)

[Procedures 8](#_Toc1863981540)

[Results 8](#_Toc1070483110)

[C4. Source Code 8](#_Toc1814249129)

[C5. Link to Live Version 9](#_Toc1226146482)

[User Guide 10](#_Toc1861484837)

[Introduction 10](#_Toc1053700842)

[Installation and Using the Application 10](#_Toc1982684711)

[Login and Signup (An example) 10](#_Toc251618402)

[Classes 11](#_Toc614110389)

[Create a New Class 11](#_Toc287900648)

[Reports 12](#_Toc918783738)

Task 2 Part C – C868 Software Development Capstone

# Application Design and Testing

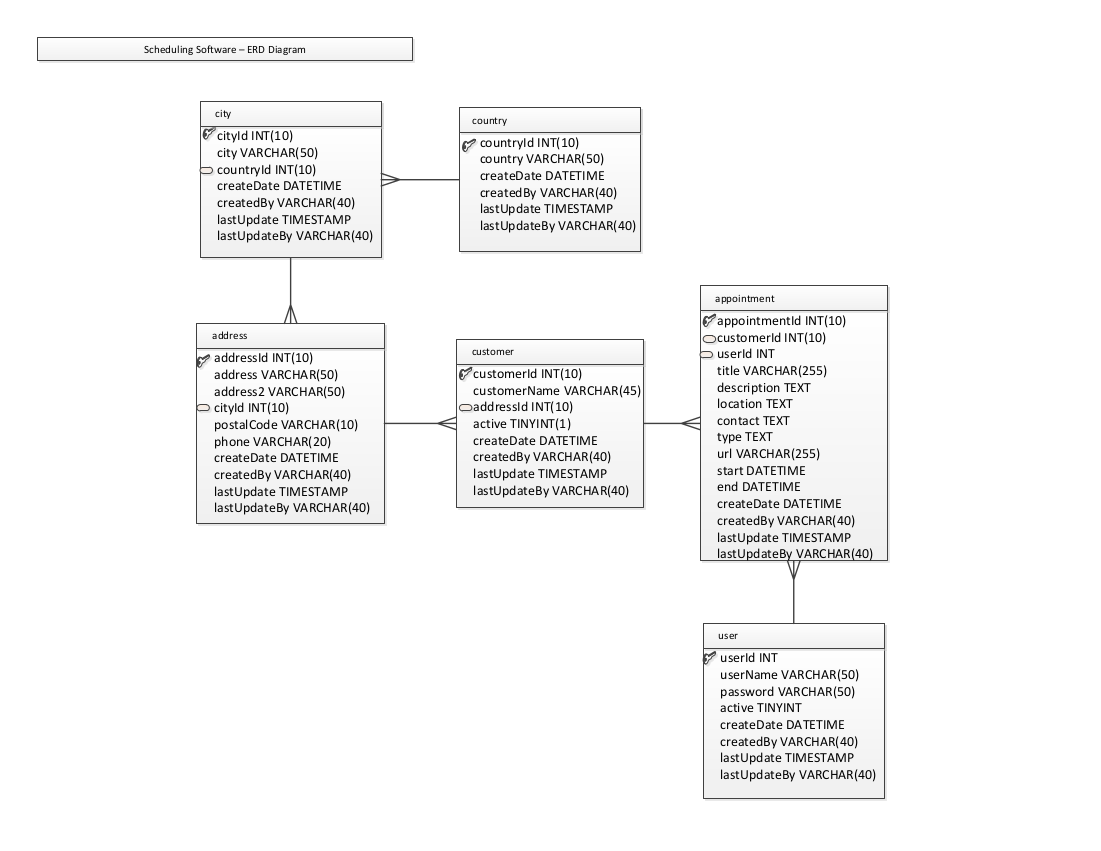
# Design Document

## Class Design

The proposed software consists of six data classes:

* + User
  + Customer
  + Appointment
  + Address
  + City
  + Country

This ERD was created based on requirements gathered from all stakeholders, and the data classes relate to each other as follows:



The address, city, and country classes are represented in the source inside of the Customer class for simplicity.

## UI Design

The proposed software consists of X UI “views” or “screens”. The UI is designed with user-friendliness and data integrity in mind to allow users of various knowledge levels to utilize the full suite of features offered by the application.

This low-fidelity wireframe was created based on requirements gathered from all stakeholders:

[IMAGE]

# Unit Test Plan

## Introduction

### Purpose

The primary method used for testing was manual testing via the UI to verify the application conformed to specifications and requirements. Individual UI “screens” were considered to be “units” and were integrated into the application after being considered to be passing. If a bug was discovered, it was documented and assigned to a member of the development team for review.

### Overview

The tests were performed by manipulating the user interface to execute the test case, in which the database and UI were simultaneously tested to perform as expected in the test case. All appointment times were verified to be saved in UTC in the database for easy inter-office scheduling and displayed in the application user’s local timezone.

The application’s error handling was also tested, to ensure that data integrity was maintained in the event of failure.

## Test Plan

### Items

The application should be running inside the work environment and all tests must pass. In case of failure, the defect is documented and sent to a developer. This process repeats until there are no further failures.

### Features

The following functionality is tested:

* + Login
  + Search
  + Weekly/monthly views
  + Create, view, update, and delete customers
  + Create, view, update, and delete appointments

### Deliverables

This test plan with all test cases and their results.

### Tasks

* + Perform the tasks described in the test cases
  + Document any defects and report to the development team
  + Repeat all test cases until there are no further failures

### Needs

The environment needs to have .NET Framework 4.7.2 running on Windows 8 or above with a working Internet connection to successfully perform tests.

### Pass/Fail Criteria

Each test case describes inputs and expected output; this is used to determine if the test results match what is expected as described by the test case.

## Specifications

Provide sample code that represents what testing code was used. Screenshots are acceptable.

## Procedures

Provide a detailed list of the steps you used to complete the testing process. Be sure to mention if iterations were/are part of the process used and when pass/fail results were provided.

## Results

Here you will describe and provide examples of the testing results. If you were using a testing package include a screenshot of the interface. Screenshot work best.

# C4. Source Code

The source code may be found in the submission, or on GitHub:

<https://github.com/mathiscode/wgu-capstone-c868>

# C5. Link to Live Version

A prebuilt release version may be found at:

<https://github.com/mathiscode/wgu-capstone-c868/releases>

# User Guide

*Note: This may be included as a separate document if you desire.*

## Introduction

Provide a description of the content you’re providing in the User Guide. This guide will include how to install, log into, sign up, and use all of the functions of the application. The steps need to be clearly defined and fully tested so the process works flawlessly for the evaluator.

## Installation and Using the Application

This procedural information should follow the basic rules of such technical references. While some procedures may provide for personal judgment yours should be clear and concise. Here are other rules to remember:

* Provide step-by-step sequences in the correct order.
* Follow the timing and sequencing of the actual operations.
* Provide visual stepping stones by using bullets or labeling steps.
* Strive to be concise. Avoid lengthy paragraphs but include enough detail so false assumptions are not made.
* Use common terms and jargon appropriate for the audience (someone with basic IT background).
* Explain why steps are completed or what they will yield as well as "How to" instructions.
* Test the instructions to ensure they match the actual product.
* Format the material for ease of reading and use graphic aids to clarify point/steps.
* Write in the present tense and the active voice.

## *Login and Signup (An example*)

1. *Click the "Log in" button in the top right corner of the app.*

**

1. *If you already have an account, log in with your account name and password. If you need an account, click on the link below that states “Need an account?”*
2. *If you need to create an account, choose a unique username and password. By default, the password requires at least 6 characters. This function could be changed to address new password requirements.*

## *Classes*

### *Create a New Class*

1. *Once logged in, click on the link at the top labeled “Classes”. This will enable you to create a new class of students.*

**

1. *Click on “+ Add Class”.*

**

1. *Enter a class name and its description. The class name must be unique.*
2. *Click “Add Class” to add the class, otherwise click “Cancel” or outside of the modal to cancel adding the class.*



## *Reports*

1. *To access the reporting feature, from the Schedule module, click on “Generate Report” near the top right of the page.*

**

1. *By default, all events are generated and displayed.*