**C868 – Software Capstone Project Summary**

**Task 2 – Section C**

|  |  |
| --- | --- |
| **Capstone Proposal Project Name:** | http://www.idevnews.com/views/images/uploads/general/wgu_logo.png  Fittr – A Golf Club Fitting Tracker |
| **Student Name:** | Elliott Moos |

# Table of Contents

[Table of Contents 2](#_Toc127810675)

[Application Design and Testing 5](#_Toc127810676)

[Design Document 5](#_Toc127810677)

[Class Design 5](#_Toc127810678)

[UI Design 7](#_Toc127810679)

[Test Plan 8](#_Toc127810680)

[Introduction 8](#_Toc127810681)

[Purpose 8](#_Toc127810682)

[Overview 9](#_Toc127810683)

[Test Plan 9](#_Toc127810684)

[Items 9](#_Toc127810685)

[Functions/Features 10](#_Toc127810686)

[Deliverables/Outcomes 10](#_Toc127810687)

[Tasks 10](#_Toc127810688)

[Needs 11](#_Toc127810689)

[Pass/Fail Criteria 11](#_Toc127810690)

[Specifications 11](#_Toc127810691)

[Procedures 12](#_Toc127810692)

[Results 12](#_Toc127810693)

[Source Code 13](#_Toc127810694)

[Link to Live Version 13](#_Toc127810695)

[User Guide 13](#_Toc127810696)

[Set Up and Run Application for Maintenance Purposes 13](#_Toc127810697)

[Prerequisites 13](#_Toc127810698)

[Installation 14](#_Toc127810699)

[Guide For Admin User 14](#_Toc127810700)

[Login with an Admin Account 14](#_Toc127810701)

[Viewing and Managing Fitters 15](#_Toc127810702)

[Viewing Fitters 15](#_Toc127810703)

[Creating a Fitter 17](#_Toc127810704)

[Deleting a Fitter 18](#_Toc127810705)

[Viewing and Managing Customers 19](#_Toc127810706)

[Viewing Customers 19](#_Toc127810707)

[Creating a New Customer 20](#_Toc127810708)

[Updating Customer Information 20](#_Toc127810709)

[Deleting a Customer 22](#_Toc127810710)

[Viewing and Managing Stores 22](#_Toc127810711)

[Viewing Stores 22](#_Toc127810712)

[Creating a New Store 23](#_Toc127810713)

[Updating Store Information 23](#_Toc127810714)

[Deleting a Store 24](#_Toc127810715)

[Viewing and Managing Fittings 25](#_Toc127810716)

[Viewing the Fitting Calendar 25](#_Toc127810717)

[Scheduling new Fittings 25](#_Toc127810718)

[Updating a Fitting 26](#_Toc127810719)

[Deleting a Fitting 27](#_Toc127810720)

[Viewing Fitting Reports 27](#_Toc127810721)

[Fitting Search 28](#_Toc127810722)

[Fittings Today 28](#_Toc127810723)

[Fittings by Fitter 29](#_Toc127810724)

# Application Design and Testing

Design Document

## Class Design

The class diagrams below display the class design for the Fittr application. FastAPI isn’t an

opinionated web framework, but this project loosely bases its architecture on the MVC architecture pattern, so the database models map directly to the model classes. The project also utilizes the repository pattern to abstract database interactions for each model into a repository that manages its own database session.

There are five main model classes: Address, Store, Customer, Fitter, and Fitting. To simplify the class diagram, the underlying Object Relational Mapper library (SQLModel) models are omitted. There is a one-to-many relationship between Fitter and Fitting, as well as Customer and Fitting. A one-to-many relationship exists between Address and Customer, Store, and Fitter. A one-to-many relationship exists between Store and Fitter, as a Fitter can only have one “home” Store.

Each repository serves as a collection of abstract database interactions for each model and is created via a FastAPI dependency for each request that requires it. The repository manages a database session for the life of the request and closes it at the end to avoid session bloat. Methods are denoted with a “()”.

Diagram, schematic

Description automatically generated

Table

Description automatically generated

## UI Design

Below are the low and high-fidelity versions of the detail page for a particular Fitter. The top navigation menu displays the main navigation options as well as a link to the current Fitter page via the “username”. There is also a “Logout” button in the top right corner of the page. The Fitter details are displayed in the left section of the page with a button to update or delete a Fitter displayed if the current Fitter is a Lead fitter only. There is a table of Fittings associated with the Fitter to the right of the Fitter details section along with a “+” button to the right of the Fittings table title that will navigate the current Fitter to the Fittings Calendar so they can add a new Fitting there.

A picture containing table

Description automatically generated

*Low-fidelity version of the Fitter detail page*

Graphical user interface, table

Description automatically generated

*High-fidelity version of Fitter detail page*

Test Plan

## Introduction

### Purpose

The Fittr web application is intended to facilitate the management of multiple resources in one place. This requires a role-based access control system to limit certain actions to management and the owners of Tyson’s West. Role-based access is critical to maintaining the integrity of the data including the ability to create and delete Fitter resources as well as Store resources. If a non-Lead Fitter had access to these actions, they could wreak havoc on the system.

### Overview

Given the role-based access control requirements, special focus was given to testing the functions that support those requirements. In FastAPI, server routes are defined by functions that define the logic which needs to be executed for a given URI and HTTP request method. FastAPI supports automated code execution for route definitions in the form of dependency injection. Each page that is involved with creating or updating Stores or Fitters utilizes a custom dependency that checks the current session token cookie for the Fitter that is logged in and compares it to the minimum role required to complete the request for the page. In the cases being tested, that custom dependency should send a “403 Forbidden” response to the client if the current Fitter does not have the “Lead” role.

The unit tests are written using the pytest testing framework, as well as the FastAPI “TestClient” plugin that allows testers to interact with a temporary version of the server in a way that makes testing easier. The tests are executed inside of a Docker container to minimize the number of packages/software the test must have installed on their computer. The tests will write the results to a results file and export them to the testers system in the same directory the tests were run in.

## Test Plan

### Items

To test that a non-Lead Fitter cannot access the Create Fitter page, we need three model instances: One Lead Fitter, one non-Lead Fitter, and one Store. Both Fitters will be associated with the same store. We can make a request to the TestClient server setting the current Fitter to a non-Lead fitter and proving that a “403 Forbidden” response is sent back. We can also make a request to the TestClient server setting the current Fitter to a Lead Fitter and proving that a “200 OK” response is sent back, indicating that the Lead fitter can access that page.

To test each case: Lead Fitter can access the Store/Fitter create/delete pages, and a non-Lead Fitter cannot, we can use the same strategy as described above for each respective server route.

### Functions/Features

To set up each TestClient, we need to initialize an empty test database and create the Fitter and Store records. We will use a “FitterRespository” and a “StoreRepository” and pass it the session associated with the test database connection to create each Fitter and Store record before any of the tests run by marking those functions with “@pytest.fixture”. This is how you define useful bits of code that you can use in any of the unit tests pytest runs.

To test the TestClient response, we will need to call the appropriate route with the appropriate HTTP request method. The following methods of the FastAPI class will be exercised:

* url\_path\_for will be called with the global name of the route and any necessary path parameters

The output of the url\_path\_for call will be passed to the appropriate HTTP request method on the TestClient instance (i.e., TestClient.get()). We can then inspect the response that is sent back and assert the value we expect it to be.

### Deliverables/Outcomes

When the tests are run, pytest produces structured console output. Each test is given a descriptive name that indicates the exact behavior under scrutiny, e.g., “test\_non\_lead\_fitter\_cannot\_access\_create\_fitter\_page”. The other parts of the output sentence reflect the context of the test.

### Tasks

To execute this unit test the following steps are required:

1. Write the code to be tested.
2. Create a conftest.py file in the root “tests” directory of the project.
3. Use the conftest file to configure pytest and any related fixtures/database setup.
4. Write the test code using pytest fixtures and plain python functions that begin with “test\_”.
5. Run the test suite by executing the “run\_tests.sh” shell script in the root of the project.
6. Examine the output to determine pass or fail.

### Needs

The test setup assumes that the tester is running on a Unix/Linux or MacOS machine. The following software packages are required to run the tests with the provided run\_tests.sh script:

* docker/docker-compose (Docker Desktop)
* bash

If the tester does not have a Unix/Linux or MacOS machine, they can run the steps in the run\_tests.sh script manually.

### Pass/Fail Criteria

The criteria for a successful role-based access request are that the server responds with a “200 OK” message and the appropriate HTML for the specified minimum Role or a “403 Forbidden” message when a Fitter without the specified minimum Role attempts to make the requests. A failing test would be the result of the server responding with a “200 OK” message for an unauthorized Fitter or a “403 Forbidden” message for a Fitter with the correct Role.

## Specifications

This is a screenshot of the test code described above. It is an excerpt from the test\_fitter\_create.py file. The test file is located at tests/server/routes/test\_fitter\_create.py in the root directory of the project. There are four more test files the tests/server/routes directory, but they all do something very similar to this code.

Text

Description automatically generated

## Procedures

As seen in the code excerpt above, there are two tests that test the role-based access control of the create Fitter page route handlers.

The first test uses a TestClient configured with a current Fitter role of “Expert”, which is non-Lead, and attempts to send a GET request to the create Fitter page route. The test attempts to assert that the response’s status code is indeed “403”, indicating that the request was denied because the client was authenticated, but unauthorized to make the request.

The second test uses a TestClient configured with a current Fitter role of “Lead”, which is Lead, and attempts to send a GET request to the create Fitter page route. The test attempts to assert that the response’s status code is indeed “200”, indicating that the request was accepted because the client was authenticated with the correct minimum role of “Lead”.

## Results

The following screenshot displays the successful test runs for all the tests in the test suite. Each test is marked as “PASSED”. As such, the expectations expressed in the test could be confirmed.

Text

Description automatically generated

Source Code

A compressed version of the source code can be found in the file fitter.zip

Link to Live Version

<https://capstone.elliottmoos.com/>

# User Guide

## Set Up and Run Application for Maintenance Purposes

Several steps are needed to setup and run this FastAPI web application in a local environment for the purposes of changing the code base or fixing bugs. This is a summary of the steps required to set up the application on Unix-based systems. If you want to develop the application on a Windows machine, please execute the steps in the provided convenience scripts manually.

### Prerequisites

The following system programs need to be installed:

* Docker Desktop. [Instructions here](https://www.docker.com/products/docker-desktop/).
* Bash (optional to run the convenience scripts)

### Installation

In the root of the project, run the run\_local.sh convenience script.

Navigate to <http://localhost:8080/> in your browser to view the site. You can also use the values in the provided .env file to create a new server in the pgAdmin console at [http://localhost:8081](http://localhost:8081/) and interact with the database tables there.

## Guide For Admin User

The rest of the User Guide will provide a walk-through of the Fittr fitting tracking application to demonstrate its functionality in detail.

## Login with an Admin Account

The application will have data pre-seeded for convenience. There are two Lead Fitters that exist in the system on startup. One is “lead1” and the other is “lead2”. Both have the same password of “password”.

To log in with a Lead Fitter account enter the respective Lead Fitter username and password in the login page. This will redirect you to the home page if login is successful.

Graphical user interface, text, application

Description automatically generated

On the login page, enter “lead1” as the Username and “password” (all lower case) as the password. Then click the Login button. You will be redirected to the home page. All menu items in the main navigation bar should be accessible to all Fitters. (Note to evaluators: This site is hosted in Digital Ocean’s NYC1 region. If you reside outside of this region, the site may take a little longer to initially load.)

## Viewing and Managing Fitters

### Viewing Fitters

To view all Fitters, click on the “Fitters” link in the top navigation bar. This will take you to a page that lists all fitters in a tabular format. The data in the table is paginated. Pagination links can be found under the table.

Graphical user interface, application, table, Teams

Description automatically generated

Graphical user interface

Description automatically generated with medium confidenceThe buttons on each row are links to either view or delete the given fitter.

The “Profile” button will redirect you to a page with profile details and all fittings for a given fitter. The Fitter page will have the fitter’s details, including the address, home store, and a bio.

Graphical user interface

Description automatically generated

If the current Fitter is a Lead Fitter or the Fitter being viewed, an “Update” button will be displayed underneath the Fitter details which redirects to a form with the existing Fitter details pre-populated. Some fields may be disabled if the current Fitter is not a Lead Fitter. For example, non-Lead Fitters cannot change their home store or role.

Graphical user interface, text, application

Description automatically generated

### Creating a Fitter

Graphical user interface

Description automatically generated with medium confidenceIf the current Fitter is a Lead Fitter, a “Create” button will be displayed underneath the Fitters table which will redirect to the “Create Fitter” page with a form including all the possible fields for a Fitter.

The form to create a new Fitter is straightforward. Required fields are marked with an asterisk. The Home Store and Role can be chosen from a pre-defined list. The street 2 address field is optional.

A picture containing website

Description automatically generated

### Deleting a Fitter

Deleting a Fitter is also straightforward. If the current Fitter is a Lead Fitter, a “Delete” button is displayed in both the Fitters table row and the Fitter profile page details. Clicking either button will redirect you back to the Fitters page.

|  |  |
| --- | --- |
|  |  |

## Viewing and Managing Customers

### Viewing Customers

To view all customers, click the “Customers” button in the top navigation section of any Fittr page. This will take you to a page that lists all customers in a tabular format. The data in the table is paginated. Pagination links can be found under the table. There is a search bar for first name and last name that supports first name, both, and last name searches. Clicking the “Search” button after a search has been executed will clear the search filter and display all customers once again.

Graphical user interface, application

Description automatically generated

### Creating a New Customer

To create a new customer, click the “Create” button displayed underneath the customers table. This will redirect you to a “Create Customer” page that contains a form with all of the fields necessary to create a new Customer.

A picture containing text

Description automatically generated

The form is straightforward. All required fields are marked with an asterisk. Once you are done filling out the form, click the “Create” button at the bottom of the form and you will be redirected to the Customers page where the new Customer will be displayed.

### Updating Customer Information

To update a Customer, you need to navigate to the Customer profile page. You can do this by clicking the “Profile” button in the last column of the customer table for the Customer whose profile you want to update.

Graphical user interface

Description automatically generated with medium confidence Graphical user interface, text, application

Description automatically generated

You can then click the “Update” button displayed under the Customer profile and this will redirect you to a form with pre-populated form values based on the Customer whose profile you are updating.

The form is straightforward. Required fields are marked with an asterisk.  
Graphical user interface, application

Description automatically generated

Once you are done updating the information for the Customer, click “Update” button. This will redirect you back to the Customer profile for the Customer you just updated.

### Deleting a Customer

Deleting a Customer is straightforward. Much like a Fitter, you can click the red “Delete” button present in both the Customer table row column and the Customer profile to delete the specified Customer. This will redirect you back to the Customers page.

## Viewing and Managing Stores

### Viewing Stores

To view all Stores, click the “Stores” button in the top navigation section of any Fittr page. This will take you to a page that lists all Stores in a tabular format. The data in the table is paginated. Pagination links can be found under the table.

Graphical user interface, application, website

Description automatically generated

### Creating a New Store

To create a new Store, you must be a Lead Fitter. Click the “Create” button displayed underneath the Stores table. This will redirect you to a “Create Store” page that contains a form with all the fields necessary to create a new Store.

Graphical user interface, text, application

Description automatically generated

The form is straightforward. All required fields are marked with an asterisk. Once you are done filling out the form, click the “Create” button at the bottom of the form and you will be redirected to the Stores page where the new Store will be displayed.

### Updating Store Information

To update a Store, you need to be a Lead Fitter. Navigate to the Store profile page for the store you want to update. You can do this by clicking the “Profile” button in the last column of the Store table for the Store whose profile you want to update.

Graphical user interface, text, application

Description automatically generated Graphical user interface, text, application

Description automatically generated

You can then click the “Update” button displayed under the Store profile and this will redirect you to a form with pre-populated form values based on the Store whose profile you are updating.

The form is straightforward. Required fields are marked with an asterisk.  
Graphical user interface, text, application, email

Description automatically generated

Once you are done updating the information for the Customer, click “Update” button. This will redirect you back to the Customer profile for the Customer you just updated.

### Deleting a Store

Deleting a Store is straightforward. You must be a Lead Fitter. Much like a Customer, you can click the red “Delete” button present in both the Store table row column and the Store profile to delete the specified Store. This will redirect you back to the Stores page.

## Viewing and Managing Fittings

### Viewing the Fitting Calendar

To view the fitting calendar, click on the “Fittr” or “Fittings” links in the top navigation bar.

Graphical user interface, table

Description automatically generated with medium confidence

You can navigate between work weeks by clicking the “Previous” or “Next” buttons on the top left and right corner of the page, respectively. Each fitting is represented by a block with a colored bar attached to the left side of it representing which fitter is associated with the fitting. The block contains the fitting description as well.

### Scheduling new Fittings

To schedule a new fitting, click on an open time block on the fitting calendar, enter a description, choose the customer, store, and fitter, and click the “OK” button.

Table

Description automatically generated

Graphical user interface, application

Description automatically generated

Updating a Fitting

The fitting calendar supports drag-and-drop rescheduling of fittings within the work week.

To reschedule a fitting, click and hold a fitting block and drag it to a new time block.

The calendar will not allow a fitter to reschedule a fitting to conflict with current fittings that would double book a customer or a fitter. The only fitters who can reschedule fittings are Lead fitters and the fitter associated with the fitting being rescheduled.

Table

Description automatically generated

Update the fitting details by clicking on the fitting block, updating the information, and pressing the “OK” button. Again, only Lead fitters and associated fitters may update fitting details.

Graphical user interface, application

Description automatically generated

### Deleting a Fitting

Deleting a fitting can only be done by a Lead fitter, or a fitter who is associated with the fitting being deleted. To delete a fitting, simply click the grey “x” in the upper right corner of the fitting block on the fittings calendar.

Graphical user interface

Description automatically generated with low confidence

## Viewing Fitting Reports

There are three reports generated for all fitters:

1. A tabular report to show fittings within a specified date range or start/end date.
2. A bar graph report on the number of fittings based on time of day for the current day.
3. A bar graph report on the number of fittings for each fitter.

### Fitting Search

This report can be viewed by selecting “Reports” > “Fitting Search” from the dropdown in the top navigation bar. The report is driven by fitter input and defaults to display all fittings in the database in a paginated table. The table shows the fitting start/end date and time, description, customer profile, fitter profile, and store profile. You can change the range by using the date pickers on the top of the page.

Graphical user interface, application, table

Description automatically generated

### Fittings Today

The report can be viewed by selecting “Reports” > “Fittings Today” from the dropdown in the top navigation bar. The report shows a blue bar chart with the number of fittings grouped by fitting block. This information can be used to make fitting schedule decisions and distribute fittings more evenly throughout the day.

Table

Description automatically generated with low confidence

### Fittings by Fitter

The report can be viewed by selecting “Reports” > “Fittings by Fitter” from the dropdown in the top navigation bar. The report shows a blue bar chart with the number of fittings grouped by fitter. This information can be used to make fitting schedule decisions and determining fitter performance.

Chart

Description automatically generated