

C868 – Software Capstone Project Summary

Task 2 – Section C



Capstone Proposal Project Name: Football Roster Program

Student Name: Casey White - 1303102

Table of Contents**Contents**

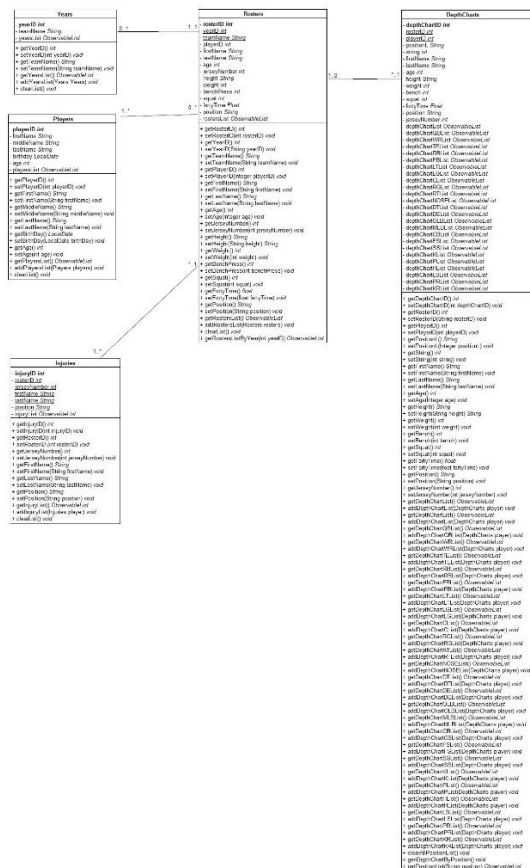
<i>Table of Contents.....</i>	<i>2</i>
<i>Task 2 Part C – C868 Software Development Capstone.....</i>	<i>4</i>
<i>Application Design and Testing.....</i>	<i>4</i>
Design Document	4
Class Design	4
UI Design.....	5
Unit Test Plan.....	6
Introduction	6
Purpose	6
Overview	6
Test Plan.....	6
Items	6
Features.....	6
Deliverables	6
Tasks	7
Needs	7
Pass/Fail Criteria.....	7
Specifications.....	8
Procedures.....	9
Results.....	9
C4. Source Code.....	9

<i>User Guide.....</i>	<i>10</i>
Introduction	10
Prerequisites.....	10
<i>How to get Started.....</i>	<i>10</i>
<i>Players</i>	<i>10</i>
<i>Create a New Player.....</i>	<i>10</i>
<i>Modify a Player</i>	<i>10</i>
<i>Delete a Player</i>	<i>11</i>
Add a Year	11
Add or Remove from Injury List	12
Add or Remove Depth Chart.....	12
Export Depth Chart.....	12
Export or Import Data	12
Default User or Change Password	13

Application Design and Testing

Class Design

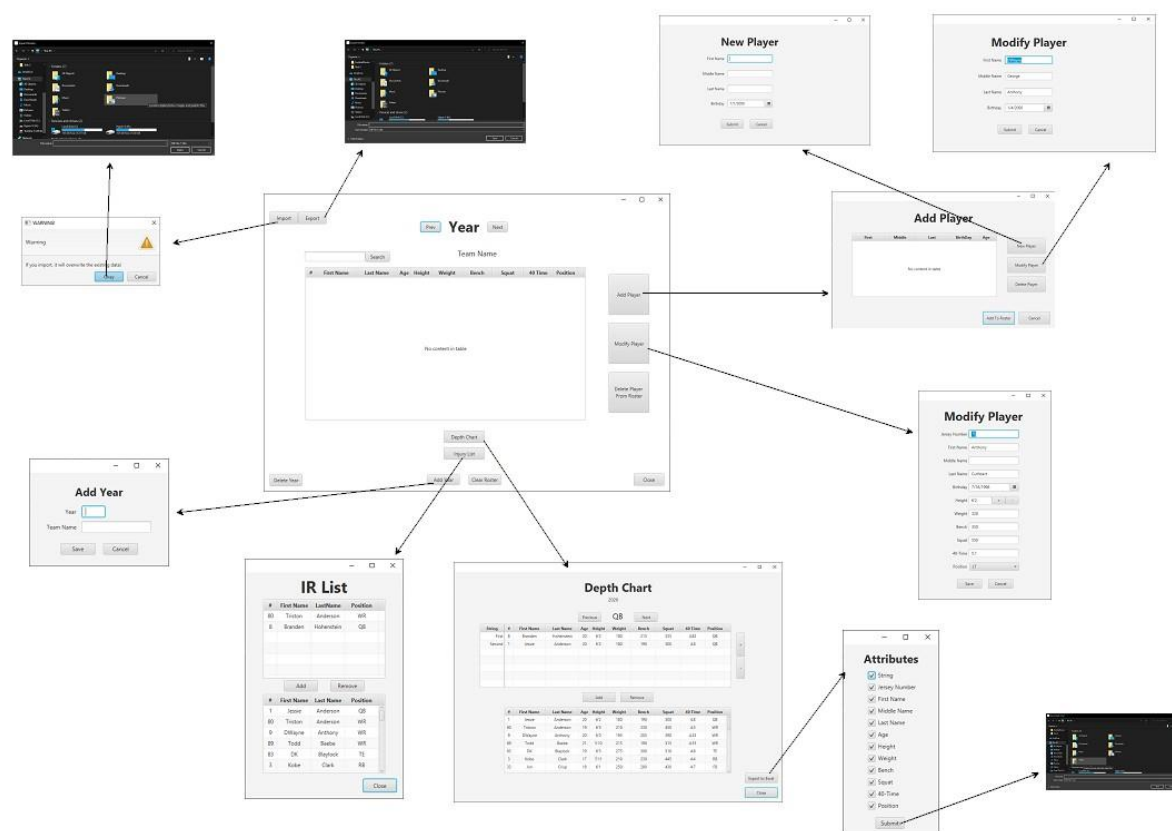
Rosters is tied to Players and Years and includes the players attributes. Players includes personal information for the player. DepthCharts has lists by position and string of the player for that position. Injuries is simply referencing Rosters as a list of current injured players.



UI Design

Below is the design diagram showing the user interface design and flow of the program.

The middle large screen is the home screen which is the first screen that opens when the program is ran. The approach was a highly functional home screen so there weren't many navigational layers so you can quickly do what you need to do.



Unit Test Plan

Introduction

Purpose

We created a single unit test for formatting height. The database stored height as an integer representing inches. The presentation shows it in a formatted way of 5'11, as an example. The unit test will test possibly inputs to validate outputs are shown currently and any crazy behaviors are fixed.

Overview

The height unit test prints out the results of each input to be tested. For example, if the stored height is 71, it should print 5'11. Which this is five feet and eleven inches. This will provide an efficient way to make sure that height is being formatted correctly to its corresponding stored data which is in inches.

Test Plan

Items

Unit test named getHeightFormattedTest().

It has five different types of inputs.

Features

The test transforms any numerical input and transforms it into readable feet and inches.

Deliverables

The test will simply print out in console the results and whether it passed or failed. This can only be ran with the source code in IntelliJ

Tasks

Open IntelliJ and load the source files. Right click and click run on `getHeightFormattedTest`.

Needs

IntelliJ and JavaFX 14.0.1.

Pass/Fail Criteria

- Input 0 : Expected 0'0

If zero inches is put in as an integer, 0'0 is the expected output which is zero feet and zero inches. Anything else is a fail.

- Input 1 = Expected 0'1

If one is input as an integer, 0'1 is the expected output which is zero foot and one inch. Anything else is a fail.

- Input 12 = Expected 1'0

If twelve is input as an integer to see if it'll flip from 0'11, 1'0 is the expected output which is one foot and zero inches. Anything else is a fail.

- Input 71 = Expected 5'11

If a typical height such as seventy-one is input as an integer, 5'11 is the expected output which is five feet and eleven inches.

- Input -20 = Expected 0'0

If any negative input as an integer is typed, 0'0 is the expected output which is zero foot and zero inches.

All these items should be also green check marked.

Specifications

Below is a screenshot of `getHeightFormattedTest()` found within the source project.

```
@Test
public void testHeightFormattedZERO() {
    int inches = 0;
    String heightFeet = String.valueOf(abs(inches) / 12);
    String heightInches = String.valueOf(abs(inches) % 12);
    assertEquals( expected: "0'0", (heightFeet + "'" + heightInches));
}

@Test
public void testHeightFormattedONE() {
    int inches = 1;
    String heightFeet = String.valueOf(abs(inches) / 12);
    String heightInches = String.valueOf(abs(inches) % 12);
    assertEquals( expected: "0'1", (heightFeet + "'" + heightInches));
}

@Test
public void testHeightFormattedTWELVE() {
    int inches = 12;
    String heightFeet = String.valueOf(abs(inches) / 12);
    String heightInches = String.valueOf(abs(inches) % 12);
    assertEquals( expected: "1'0", (heightFeet + "'" + heightInches));
}

@Test
public void testHeightFormattedREALISTIC() {
    int inches = 71;
    String heightFeet = String.valueOf(abs(inches) / 12);
    String heightInches = String.valueOf(abs(inches) % 12);
    assertEquals( expected: "5'11", (heightFeet + "'" + heightInches));
}

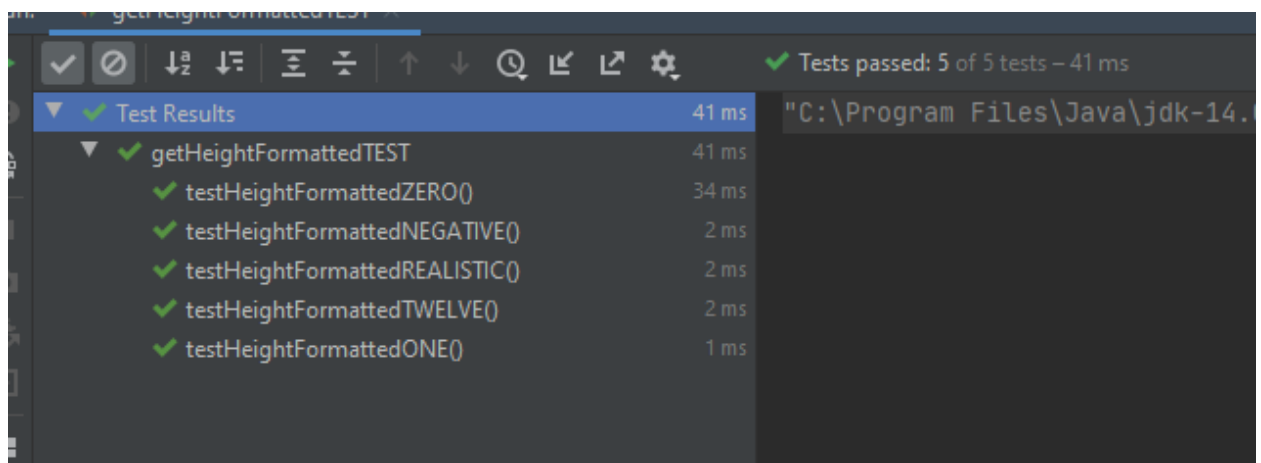
@Test
public void testHeightFormattedNEGATIVE() {
    int inches = -20;
    String heightFeet = String.valueOf(abs(inches) / 12);
    String heightInches = String.valueOf(abs(inches) % 12);
    assertEquals( expected: "1'8", (heightFeet + "'" + heightInches));
}
```


Procedures

We right clicked the function we wanted to test and had IntelliJ develop the test some what automatically. We then manually created each different test for potential input specific to the talked about function.

Results

Below is a screen shot of the simple unit test used for formatted height. The class pertains to the formatted height function with each method with potential inputs. The screenshot shows five tests passed for this specific unit test.



C4. Source Code

The source code can be found in the root of the zip file. It is also a zipped file named "SOURCE FILES – 1303102 Casey White – FootballRoster".

User Guide

Introduction

This guide goes over prerequisites, how to get started, adding a year, adding a player, modifying a player, deleting a player, adding to the injury list, removing from the injury list, exporting the depth chart, exporting the data, and importing the data. Everything is relatively self-explanatory and easy to follow.

Prerequisites

- Windows 10 Operating System
- Java SDK 14.0.1 or java runtime that supports this as a minimum

How to get Started

1. *Extract the 'FootballRoster Executable' to the location desired. This program is a standalone.*
2. *Go to where it is extracted and simply double click the icon to launch the program.*

Players

Create a New Player

1. *Launch the program*
2. *Go to Add Player*
3. *Click New Player*
4. *Fill in the fields and click submit*

Modify a Player

This can be done two ways. Either through the roster where you can also edit the attributes or from the add player view.

Through the Roster:

1. Launch the program
2. Select a player to modify
3. Click Modify Player.

Through the Add Player View:

1. Launch the program
2. Click Add Player
3. Select a player to modify
4. Click modify player

Delete a Player

You can either delete a player from the roster or from the entire database.

To delete from the roster:

1. Launch the program
2. Select a player
3. Select Delete player from roster

To delete from the database:

1. Launch the program
2. Select Add Player
3. Select a player
4. Select Delete

Add a Year

1. Launch the program
2. Click add a year

3. Fill in both fields and click submit

Add or Remove from Injury List

1. Launch the program
2. Click Injury List
3. Add or remove players from the list by selecting the player and add or removing

Add or Remove Depth Chart

1. Launch the program
2. Select Depth Chart
3. Select a player to add or remove from the DepthChart
4. You can change their string by click the player and clicking the plus or minus button

Export Depth Chart

1. Launch the program
2. Select Depth Chart
3. Click Export to Excel
4. Select the attributes to be included
5. Click submit
6. Navigate to where you want to save it and name it
7. Then click save

Export or Import Data

Import:

1. Launch the program
2. Select Import
3. Locate the file to import

4. Click it
5. Select open

It will overwrite existing data.

Export

1. Launch the program
2. Select Export
3. Navigate to where you want to save it
4. Name it and click save

Default User or Change Password

- Default user = Username: Admin Password: Password1
- After logging in, the password can be changed by clicking change password in the top right of the home screen.