Project Report

# Introduction

In this report I plan to cover my experiences of developing software using object orientated programming in C# (.net).

# Solution

-met the requirements

-most the additional

-little to no duplication

-testing the main functionality (not the UX or database)

-would need to me made async in live environment

-UX is a bit cluttered

# Approach Taken

-Waterfall?

-Created interfaces that met the design doc

-allowed me to see if the classes I made had all the required fields/methods

-Used EF to produce code first from database models

-added data annotations for validating

-as using WPF MVVM could bind errors to view

-implemented classes based on interfaces

-created Tests

-fixed issues identified by tests.

-created repositories to hit the database

-created mock repositories for testing

-this would allow more testing going forward

-always have same data for tests

-created viewmodel to hold the data for UX

-this contented the files required to bind to the view

-created view

-made it scale with screen size

-bound viewmodel to view

-ran through view to test for bugs

-fixed bugs as necessary

-binding to type not property name

-not displaying error messages

-forgetting <DataGrid.Columns> so entire datagrid crashed UX when bound

-added advanced features

-repeated

Liked:

-Focused on what needed done first

-interfaces meant I knew I had every required method and field

-was able to use wpf mvvm instead of winforms

Disliked:

-wpf mvvm made some parts of the task harder (own fault)

-wish I has modelled out the application in UML or similar as occasionally I had to redo something as I had misunderstood the doc.

# Object Oriented Programming in C#

Like:

-Nice language (easy to understand)

-Has powerful features (LINQ, lambda)

-Is now multi-platform (core)

Dislike:

-Intermediate language - slower to run then native

-Some of the keywords are a little odd (e.g. abstract)

# Visual Studio Development Environment

Like:

-Familiar with

-Code Snippets (ctor, prop, etc)

-Easy to use

-Lots of information online

-Dark theme

Dislike:

-Costly for advanced features

-Sometimes hard to find the option you want (buried in 4 menus)

# Went Well

-Work done early

-Language I know well

-Similar to applications I have built in the past so knew where to start.

# Went Poorly

-Practitioner comments as mvvm does not handle dialogs well (had to work around)

-Misreading design doc (wasted time and effort) allowed more the \*2 of the value

-Should have done TDD so bug were identified quicker

# Conclusion

-Overall liked the project

-Did not learn much (background knowledge)