# **Project Document**

# Fiddle Player Practice App

## Version 1.0

## [1/10/2019]

## CPSC 4910 SP19

## Prepared by:

Team 2

*Jakob Beckleheimer, Dylan Brownell, Tanner Groll, Derek Pendleton, and Phil Snider*

<Fiddle Player Practice App>

SECTION 1 - INTRODUCTION #

1.1 PURPOSE #

1.2 SCOPE #

1.3 TERMINOLOGY #

1.4 REFERENCES #

SECTION 2 – MANAGEMENT #

2.1 GENERAL DESCRIPTION #

2.2 SOFTWARE VALIDATION #

2.3 CONFIGURATION MANAGEMENT PLAN #

2.4 PROJECT SCHEDULE #

SECTION 3 - REQUIREMENTS #

3.1 FUNCTIONAL REQUIREMENTS #

3.1.1 Functional Capabilities #

3.1.1 System Inputs #

3.1.2 System Outputs #

3.2 NONFUNCTIONAL REQUIREMENTS #

3.2.1 Performance Considerations #

3.2.2 User Interface #

3.2.3 Language #

3.2.8 Operating Environment #

SECTION 4 – DESIGN #

4.1 Configuration Design #

4.2 Technical Design #

4.3 SYSTEMS ARCHITECTURE #

SECTION 5 - IMPLEMENTATION STRATEGY #

SECTION 6 - USER DOCUMENTATION #

6.1 INSTALLATION GUIDE #

6.2 USER'S GUIDE #

SECTION 7 - TEST AND VALIDATION #

SECTION 1 - INTRODUCTION

The Fiddle Player App is an application intended to aid fiddle or violin players learn how to play and practice their instrument. The app shall assist with tuning, intonation, scales, tempo, etc. (Stakeholders: Russ Hare)

* 1. PURPOSE

The purpose of this project is to work as a team to produce a well-functioning app that assists the user who wishes to play a fiddle or violin in various ways.

1.2 SCOPE

The app will be programmed in C# in order to easily make it multi-platformed. The team is expected to work in a Scrum based environment, with a team of 5 members. The app must be able to help the user with tuning, intonation, scales, tempo, etc. Professor William Hare will act as the Enterprise Project Manager for the project. The app will be developed using the best possible engineering processes and practices in mind. Reuse of code will be used.

1.3 TERMINOLOGY

This is where you would list any and all acronyms and terms the average reader or customer might need. This will probably be added to as the project progresses.

1.4 REFERENCES

List any sources of information

SECTION 2 – MANAGEMENT

* Jakob Beckleheimer – Application Architect
* Dylan Brownell – QA Manager
* Tanner Groll – SA Manager
* Derek Pendleton – Development Manager
* Phil Snider – Project Manager

2.1 GENERAL DESCRIPTION

A functioning “Fiddle Player” app.

2.2 SOFTWARE VALIDATION

2.3 CONFIGURATION MANAGEMENT PLAN

How are you going to handle configuration management? It needs to be thought about early and upfront.

2.4 PROJECT SCHEDULE

This should be a part of each and every report, not only iterations. You need to decide by Iteration I how you are going to track your project.

SECTION 3 - REQUIREMENTS

CPSC 4900

3.1 BUSINESS REQUIREMENT SPECIFICATION (BRS)

(User Requirements)

3.2 FUNCTIONAL REQUIREMENT SPECIFICATION (FRS)

(System Requirements)

3.2.1 Functional Capabilities

3.2.2 System Inputs

3.2.3 System Outputs

3.3 NONFUNCTIONAL REQUIREMENTS

3.3.1 Performance Considerations

3.3.2 User Interface

3.3.3 Language

3.3.8 Operating Environment

SECTION 4 – DESIGN

CPSC 4900 Iteration I and maybe Iteration II but no later than II. You should include UML diagrams in this section. You may need to go back and review add to and delete from as the semester progresses.

4.1 SYSTEMS ARCHITECTURE

4.2 Configuration Design

4.3 Technical Design

SECTION 5 - IMPLEMENTATION STRATEGY

CPSC 4900 No later than Iteration II. You may need to go back and review add to and delete from as the semester progresses. I expect a detailed explanation on how to you plan to implement your project. How do you plan to meet the requirements?

SECTION 6 - USER DOCUMENTATION

This section is a necessity for each project and can be completed last, prior to delivery.

6.1 INSTALLATION GUIDE

6.2 USER'S GUIDE

6.3 Administrator Guide (Possibly)

SECTION 7 - TEST AND VALIDATION

I want to see a test plan no later than iteration II. As you decide how you are going to implement your project you should define this. It may change somewhat as the project progresses but thought on the front end needs to be given. The earlier the better. **Include a test plan, test cases, and validation of the software**. You may need to go back and review add to and delete from as the semester progresses.