**Taylor Mcwilliam**

**EM 1**

**2/6/21**

**Software Engineering**

**Software Requirements Specification**

**(SRS) Document**

**< Nirvana VST reverb component>**

**<2/5/21>**

**<V.1>**

**<By: Taylor Mcwilliam>**

|  |
| --- |
| Revisions |

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Primary Author(s) | Description of Version | Date Completed |
| 1 | Taylor Mcwilliam | Beta v.1 | 2/6/21 |

|  |
| --- |
| Review & Approval |

Requirements Document Approval History

|  |  |  |  |
| --- | --- | --- | --- |
| Approving Party | Version Approved | Signature | Date |
| Bineet Sharma |  |  |  |
|  |  |  |  |

Requirements Document Review History

|  |  |  |  |
| --- | --- | --- | --- |
| Reviewer | Version Reviewed | Signature | Date |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |
| --- |
| Table of Contents |

1. Introduction 3

1.1 Purpose 3

1.2 Document Conventions…………………………………………………………………….3

1.3 Intended Audience…………………………………………………………………………3

1.4 Scope……………………………………………………………………………………….3

1.5 References………………………………………………………………………………….3

2. General Description 3

2.1 Product Perspective………………………………………………………………………...3

2.2 Product Features 3

2.3 User Class Characteristics 3

2.4 Operating Environment 3

2.5 Constraints 3

2.6 Assumptions and Dependencies 3

3. System Requirements 3

4. External Interface Requirements 4

4.1 User Interfaces 4

4.2 Hardware Interfaces 4

4.3 Communications Interfaces 4

4.4 Software Interfaces 4

5. Non Functional Requirements 4

**1. Introduction**

* + **Purpose:** Nirvana is a digital audio component otherwise called VST, virtual studio technology, it’s purpose is to be a low latency, small sized reverb plug-in that is low priced yet high quality, to be used in any D.A.W. (Digital Audio Workstation).
  + **Document conventions**:D.A.W. (Digital Audio Workstation),VST, virtual studio technology, Reverb: the act of digital creation of reverberation of digitally recorded sound. GUI, Graphical User Interface.VST2
  + Available for Windows and Mac
  + VST2 (virtual studio technology) is the most widely used audio plugin format and has been in use for many years. On Windows, it is the standard format used by hosts, and you can be pretty certain that your Windows DAW will support it. The VST format is also designed for Mac though it is slightly less widespread. On Windows the VST2 format is in the form of a .dll file, and on Mac is the .vst file. For a list of DAWs that support VST2 please see the wikipedia page. A few examples of popular DAWs that do not support VST2 are Logic, Garage Band and Pro Tools.
  + VST3
  + Available for Windows and Mac
  + VST3 (virtual studio technology 3) is the most recent format designed by Steinberg. It is an entirely new format – different from the VST2 format. It is designed with a few features that are useful particularly for feature-heavy plugins. Despite being a new and slightly more powerful format to VST2 it is not widely supported yet.
  + AU
  + Available on Mac only
  + AU (audio unit) is a format designed by Apple. It is roughly comparable to the VST2 format though is not compatible with it. The AU format comes in the .component file format. AU is only for Mac and is then only format supported by by Apple’s DAWs Logic and Garage Band.
  + AAX
  + Available on Windows and Mac
  + AAX (avid audio extension) is the new format for use in Pro Tools. Pro Tools is very well established host used throughout the music production, mixing and mastering industry. This type of format is unique to Pro Tools, but is still worth mentioning due to Pro Tools’s prevalence in those areas of audio production.
  + Standalone
  + Some developers also offer a standalone version of their products. As the name suggests this is not actually a plugin. It is just a version of audio plugin that can be launched like you would a normal desktop application. It does not require a DAW in order to work. This is a convenient way for users to use the plugin and is sometimes helpful for live performances.
  + 32 bit plugins versus 64 bit plugins
  + 32 bit or 64 bit are types of computer architecture, with 64 bit being much more common in modern computers (the wider addressing width of 64 bit architectures is generally advantageous in modern computing). When a plugin is described as one of these, it simply means it is built for that particular architecture. The complications come from the fact that some DAWs that run on 64 bit systems can host both 32 bit and 64 bit plugins. This is not always the case and therefore it is important to ensure that you have a host that is capable of hosting the plugin, otherwise it will not work. Finding the architecture of your DAW and operating system is found in different ways, but a quick google search will quickly point you to the right place to look.

* + **Intended audience**: all stakeholders of the project, developers, project managers, and testers for better clarity.
  + **Scope:** The benefits of this project include low cost in both production, development and memory space, with the advent of mobile virtual studios on smartphones the marketplace for consumer and professionals continues to grow and create large potential revenue possibilities .
  + **References:** None at the moment, see JUCE IDE or Waves.com to get a better picture of both the C++ platform/IDE for GUI and Audio Plug-in creation as well as VST software available on market respectively .

**2. General Description**

**2.1 Product perspective**: The product was birthed out of perspective that a light-weight and affordable Reverb that can create rich, immersive sounds and work efficiently in conditions where host software program is balancing large CPU loads and will benefit greatly from low-latency, low cost(memory) plugins that will help prevent CPU overload and crashing.

**2.2 Product features**: features include a GUI and Audio Plugin that works with all D.A.W. formats: VST, AU...

* + User class and characteristics: categories of users include: consumer and professional audio artists: music producers, recording, mixing and mastering engineers at every level, the code will be open source so users also will be programmers and non-programmers alike.
  + Operating environment: will operate in select older versions and all newer versions (as of 2/7/21) of; Windows, Mac, Linux and Ubuntu.
  + Constraints: The only constraints in development is ensuring low-latency and quick read write speeds as is crucial especially when creating digital instruments for musical production where timing is key and errors are audible.
  + Assumptions and dependencies: The main assumption is that JUCE IDE will be able to compile code in such a manner that it produces error-proof code in every plug-in format and is usable on all listed operating systems.

**3. System Requirements**

**3.1 Functional requirements**

Operating requirements are as follows:

**Mac**

**CPU**

Intel Core i5 / i7 / i9 / Xeon

[See notes regarding Apple M1 (ARM-based) processors](https://www.waves.com/support/waves-with-apple-m1-processors)

**Memory**

8 GB RAM

8 GB free disk space on the system drive

**Operating System**

10.13.6, 10.14.6, 10.15.7, 11.0.1 (Intel only)

**Screen Resolution**

Minimum: 1024x768

Recommended: 1280x1024 / 1600x1024

USB displays are not supported as the primary display.

V12 plugins require a graphics card that supports ‘Metal’. [Learn more](https://support.apple.com/en-us/HT205073).

**Windows**

**CPU**

Intel Core i5 / i7 / i9 / Xeon / AMD Quad-Core

**Memory**

8 GB RAM

8 GB free disk space on the system drive

**Operating System**

Windows 10 64 bit

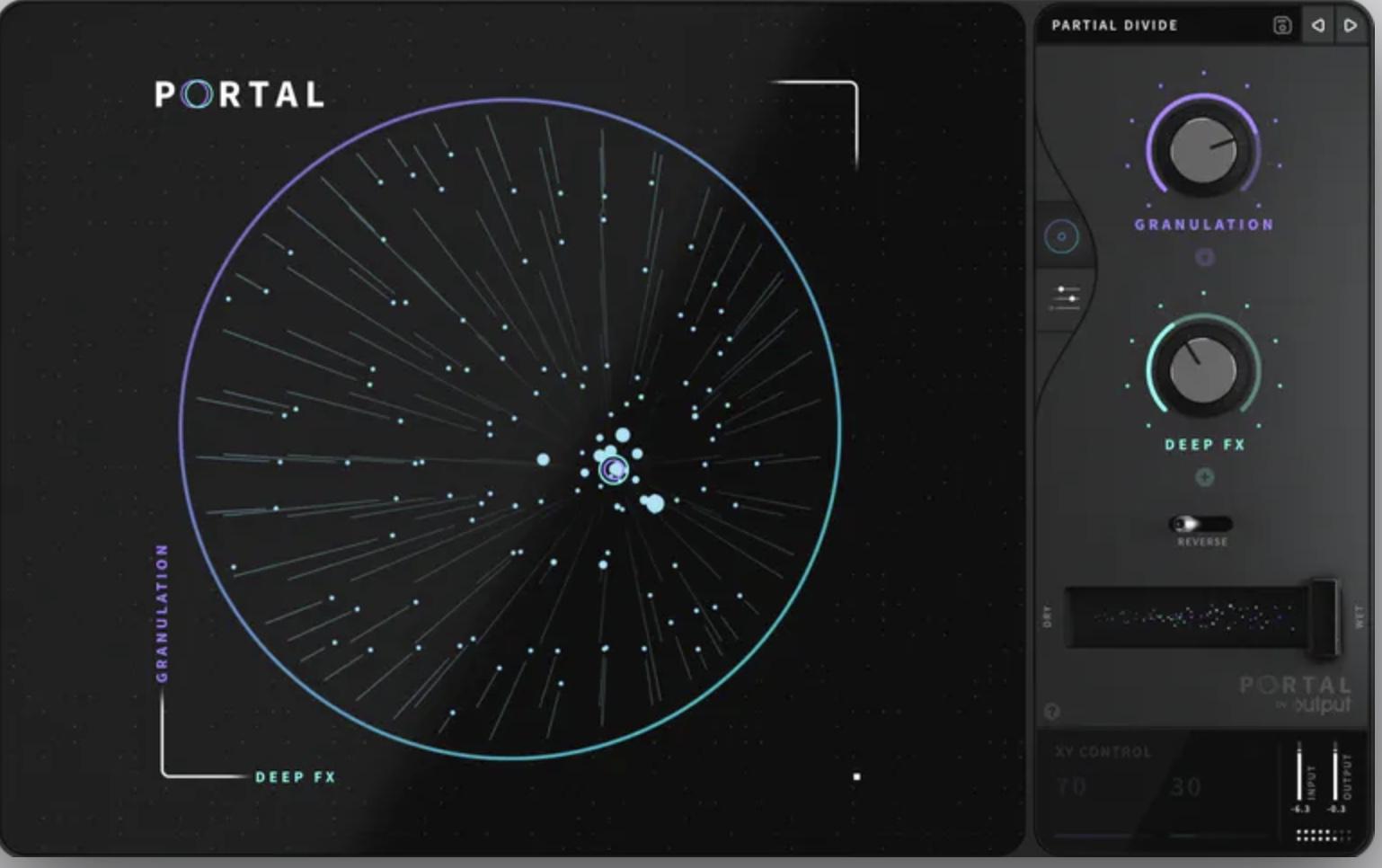
**Screen Resolution**

Minimum: 1024x768

Recommended: 1280x1024 / 1600x1024

**4.External Interface Requirements**

4.1 User Interfaces



4.2 Hardware Interfaces

None.

4.4 Software Interfaces

D.A.W. , operating system..

5. Non-Functional Requirements

5.1 Performance requirements

Low-latency, low-cost.

5.2 Safety requirements

None at the moment.

5.3 Security requirements

Open Soucet

5.4 Software quality attributes

Low maintenance .

5.5 Other requirements

Updates will be available in future.