Pyo-Synth

A digital Synthesizer for mac os X machines

[1 Introduction](#_ck3fc0gq5d6x)

[1.1 Purpose](#_e2mzwt6lrgzu)

[1.2 Document Conventions](#_d6dedn17jmma)

[1.3 Intended Audience](#_v4yz7t423rvz)

[1.4 Project Scope](#_e9fol4k5qr9l)

[2 Overall Description](#_ylrijt2yw328)

[2.2 Features](#_n8pn84ye2blx)

[2.1.1 Create sound profiles](#_k071x16x3u7g)

[2.1.2 save and load work](#_qn441ygphkf8)

[2.1.3 save and load online](#_7aa8tzwpkhmp)

[2.3 Operating Environment](#_as5af5u8fbn7)

[2.4 Dependencies](#_keoo6ptc3s68)

[3 System Features](#_7mdlqbhoec0)

[3.1 Functional Requirements](#_h6uvxck55n31)

[4 External Interface Requirements](#_e7p4mcm52rvg)

[4.1 User Interfaces](#_mzizutifuk1a)

[4.2 Hardware Interfaces](#_3au8jikk4aul)

[4.3 Software Interfaces](#_gnvw2v87vhjd)

[5 Nonfunctional Requirements](#_61gihqtqetjw)

[5.1 Software Quality Attributes](#_k3s793qjphq6)

# 1 Introduction

## 1.1 Purpose

The purpose of this document is to build a digital synthesizer Application that takes midi input from external midi controllers and apply them to waves and filters in the pyo sound library.

## 1.2 Document Conventions

This section outlines the following conventions

## 1.3 Intended Audience

This is a synthesizer application intended for digital Musicians and Music hobbyists with an entry level understanding of synthesis. Introduces the concepts of Waveforms and filters to those with an elementary understanding of the concept.

## 1.4 Project Scope

The purpose of the Pyo-Synth is to give users an easy way to connect external midi controllers to their machine and create music.

# 2 Overall Description

## 2.1 Features

### 2.1.1 Create Pound Profiles

Users can can create synthesizer profiles to modify sound output from the pyo sound library.

### 2.1.2 Save and Load Work

Users can save their synthesizer profiles locally and load previously saved work

### 2.1.3 Save and Load Online

Users can create a profile which allows them to save their work online and download their previous work, or the work of others.

## 2.3 Operating Environment

* Database: sql
* Operating system: Mac os X
* Required libraries: MySQLdb, WxPython, Pyo-C library
* Platform: python 3.6

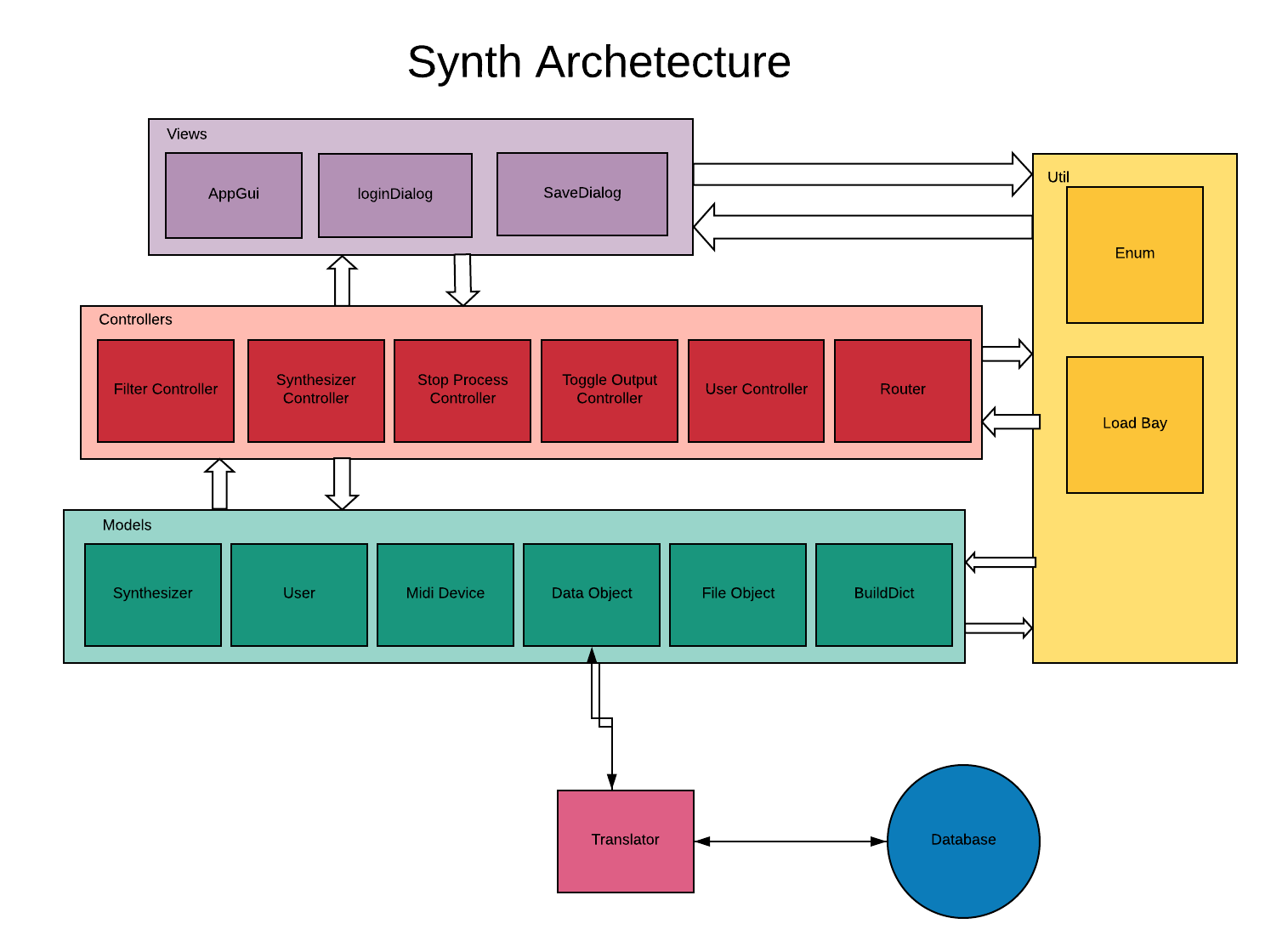
## 2.4 Dependencies

We need to assume that users who are going to be using this application have little to no working understanding of how synthesis or audio processing is going to work. So the application needs to be approachable and usable first and foremost.

Users then need to have enough different parameter options to experiment and play with different sounds and orders of filters and audio effects. There needs to be the ability to have enough variation between these effects as well.

Lastly, users need to be able to save whatever orders of waveforms and filters they’ve selected as a Preset, they then need to be able to save these presets locally or onto the cloud.

## 2.5 Diagram



# 3 System Features

## 3.1 Functional Requirements

Application must receive midi note data and output audio from the instrument according. Users need to be able to select between multiple different waveforms and audio effects in order to form some understanding of what the different options will sound like together. Users then need to be able to save the settings they have made in the synth as presets. Presets can either be saved locally as text files or on to the cloud so long as the users have an account.

# 4 External Interface Requirements

## 4.1 User Interfaces

* Front-end software: Python 3.6.7
* Back-end software: mySQL 5.7

## 4.2 Hardware Interfaces

* Mac OSX
* Midi input source (hardware or virtual)

## 4.3 Software Interfaces

We have chosen Mac OSX as our operating system because of its stability and “creative” user base.

For our programming language we decided to use python because of its readily usable music audio library Pyo. For the GUI we used WxPython as our library just to avoid future issues with Pyo’s native dependencies on WxPython.

To store users and user info we used MySQL 5.7 because it is what is used on Google cloud services. We opted with google cloud services its dependability, reliable uptime, and scalability for potentially larger amounts of users and user info.

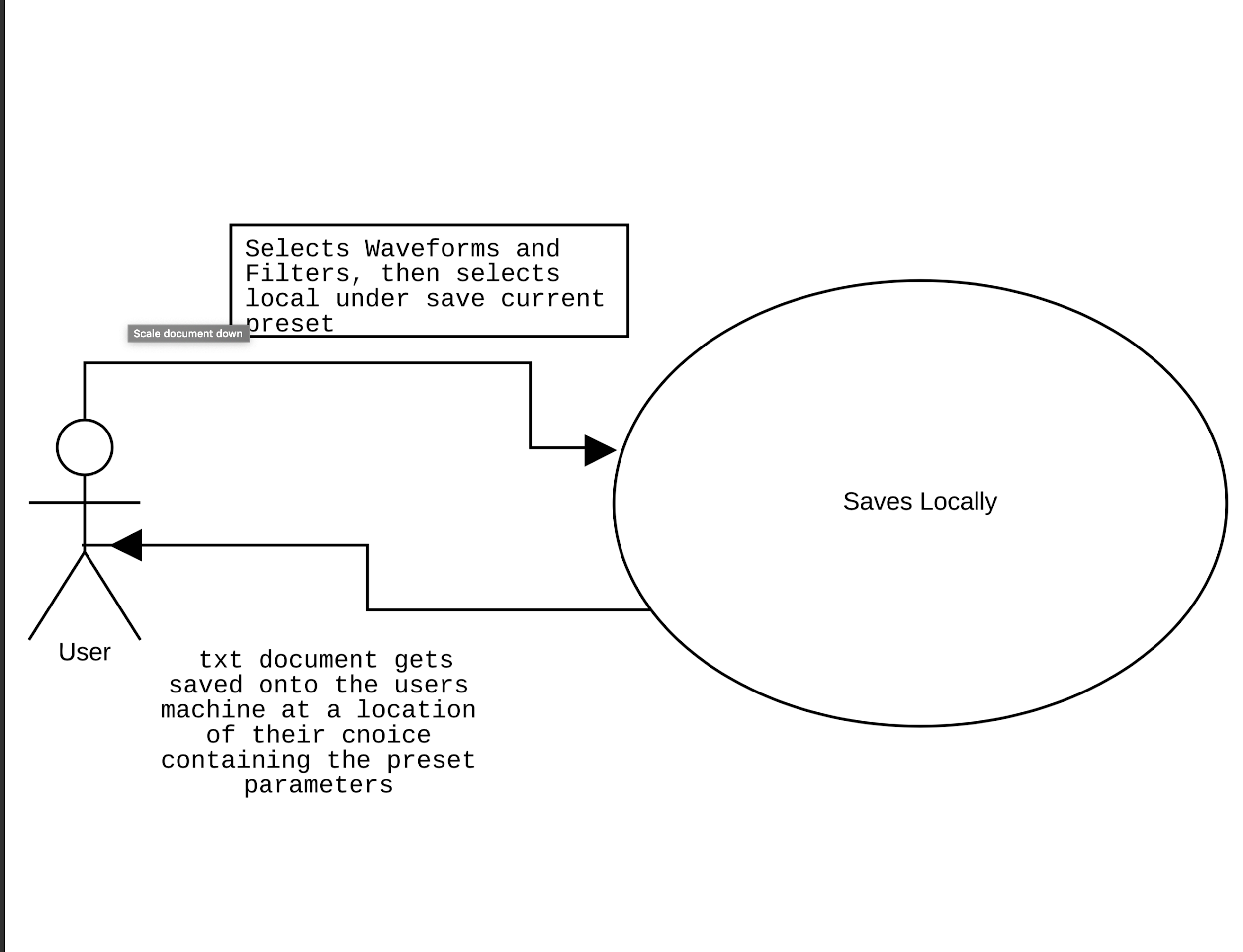
# 5 Nonfunctional Requirements

## 5.1 Software Quality Attributes

* **Accessibility:** Any person with some means of outputting midi data should readily be able to make sound and quickly be able to understand how they are making the sounds they are making, and then how they can get other sounds they may want.
* **Practicality:** Under the assumption that the user doesn’t have much or anything in the way of understanding how sound design or synthesis works, this application should give them some elementary insights into what settings make what sounds on other, more complicated synthesizers or sound generators
* **Maintainability:** The software needs to be able to be readily and easily updated to have new features such as new waveform generators and effects/filters.

# 6 Use Cases

## 6.1



## 6.2

