**Large Applications Practicum**

**Group-3**

Design Document

Vishnu Priya Jindal (B16041)

Ananya Shukla (B17007)

Aman Saxena (B17034)

Aaditya Arora (B17071)

***October 2019***

**Evoice:** The **Text-to-Speech** Synthesizer

Design Document

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author(s)** | **Description** |
| v1.0 | 10/16/19 | Vishnu Priya Jindal, Aaditya Arora, Aman Saxena, Ananya Shukla | Initial version |

**Table of Contents**

[**Introduction**](#_nyz2cen74qqk) **2**

[Design Overview](#_qn5vgyhe6g4) 3

[Intended Audience](#_e1ue4f7m2vx4) 3

[References](#_op5zze80on31) 3

[**Detailed Design**](#_kr3gavoqr8m1) **3**

[Architecture](#_ga9zwr5p7a95) 4

[Components](#_evq1kepobnwe) 4

[Interfaces](#_tdu2c8gyq3iv) 4

[Algorithms and Data Structures](#_1sprssh8m43x) 4

[External Data](#_xxnoppst2877) 4

[Databases](#_8xsv74hi370) 4

[Files](#_lqzg10iayoor) 5

[Performance](#_frzu471go0ce) 5

# Introduction

Evoice is a text-to-speech synthesizer for English and other languages, developed in order to help the visually impaired people using computer generated voice which can read the text to the user.

## Design Overview

*Describe the overall design philosophy. Eg, that the product is based on an OSS platform, or that it uses an RDBMS to efficiently handle large amounts of data.*

*Evoice* is based on Festival TTS which is a C++ library that offers text to speech through several APIs. *Evoice* has created a Python2 based wrapper that calls the functions from Festival. The GUI is built in Python2 and reads text either from the input field or from a file.

## Intended Audience

This document is intended for software designers and students who have an interest in the field of software development. This may also be read by the users of this product.

## References

[1] <http://espeak.sourceforge.net/>

[2] <http://www.cstr.ed.ac.uk/projects/festival/>

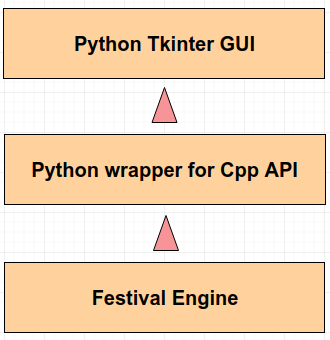
[3] <https://en.wikipedia.org/wiki/Speech_synthesis>

# Detailed Design

The software is written in Python2 and uses the C++ APIs of Festival.

## Architecture

The product is built on the Festival TTS that provides C++ APIs to read the text in a computer generated voice. It creates Python wrappers to call the C++ functions provided by Festival TTS.



#### Components

**Festival** is used as-is, offering APIs for text to speech conversion.

**Main.py** contains the wrappers in Python2 that are used to call the C++ functions of Festival.

**GUI** built using Tkinter library that provides the text field for input, slider for the rate and an option to choose a text file.

#### Interfaces

The Backend of the application is written in Python and is using Festival Cpp API. The Frontend of the application is also written in Python and is connected to the backend by importing the functions defined in the backend.

## Algorithms and Data Structures

There are no significant algorithms or data structures developed for this product.

## External Data

#### Files

*Evoice* provides the users with an option to read the text files apart from the manual method.

## Performance

The performance of the product can be tested by running files of different sizes.

## 