English Listening App Documentation

# General Overview

* Description of the scope of the project (from srs maybe?)
* Description of a general functionality?

# Database

## Overview

For this project, English Listening App (ELA), we needed a database to store all of the information that related to the countries, grades, topics and lessons. Each of these categories have relations to all of the others. We needed to be as atomic as we could for both good database design and planning for the future, more countries. The database has single entities and relationships that help keep the database organized as well as having it in a form to optimize the ability to supply data to the end-user efficiently. Since the application requires MP3’s, the database needed the ability to supply that functionality.

## Design

For starters, we chose a MySql database since we were all familiar with it and it is a very reliable and popular database. Taking in to consideration of both good database design and the needs of our project, we came up with an atomic and robust database scheme. This allowed for easy adding of new atomic items as well as the relationships between them. Our atomic objects are: countries, grades, and topics. Each of these are a single column table that represent a single entity relating to their title. Next we have our relationship tables: country\_grade\_relationship and country\_grade\_topic\_relation. These both help with the organization of our database as well as easy to query for the end-user. The last table is the lessons table, the lowest level table. This holds the filepath to the mp3 file.

## Implementation

After the design was complete and checked by Professor Dan Li at Eastern Washington University, we got to implementing the database on a localhost, XAMPP’s phpMyAdmin. The countries table has a single column labeled ‘cid’……should we just post images??

\*\*\*describe the primary/foreign key relations

\*\*\*describe the cascading constraints on lessons table so that if a c\_g\_t is edited or deleted so is the lesson.

# End-User

# Admin

## Overview

The admin’s role for this application is to keep the database up to date for the end-user. This involves adding and deleting sections and lessons in the database.

## Design

The first design decision was to make the admin portion a web-based project. This would make it easier on the client to take files from their computer and upload them as well as have a larger interface and better user experience though a website, most likely used through a PC. The thought behind the design of the admin website was to make it as intuitive as possible as well as to disallow as many mistakes as possible. For example, certain fields would be auto-populated in order to easily keep the database in good form without asking for any more user input as necessary. This also means that there is less typing, just simple click and choose. The only required typing would be for countries and topics, or certain parts of the lessons. Since this is very private data, we implemented a secure login page through asp.net, which the whole site is built from.

## Implementation

\*\*\*login page implementation

The second page solely handles adding to the database. There is a section for every piece to the database, excluding grades: country, topic, country\_grade, country\_grade\_topic, and lesson. The country and topic require a typed input. As for country\_grade and country\_grade\_topic, these have autofilling dropboxes so that it is both easy for the admin to be able to select and add to the database as well as keep the database in good state by only allowing entries to be added that have not been added yet and not allow any entries to be added that cannot be added due to key restrictions. When / if an action is not allowed or something goes unexpectedly, a notification is prompted to the screen for the admin to see.

The third page deals with the deleting and editing of the database……