Low Level Design

Personal Audio Book Desktop App

Author: Shraddha Giradkar

Contents

1. Introduction

1.1. What is Low-Level design document?

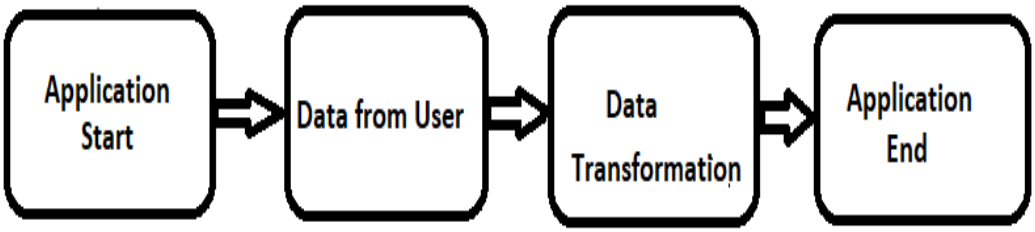
The goal of LLD or a low-level design document (LLDD) is to give the internal logical design of the actual program code for Personal Audiobook Desktop App System. LLD describes the class diagrams with the methods and relations between classes and program specs. It describes the modules so that the programmer can directly code the program from the document. The LLD phase is the stage where the actual software components are designed.

During the detailed phase the logical and functional design is done and the design of application structure is developed during the high-level design phase.

1.2. Scope

Low-level design document (LLD) is a component-level design process that follows a step-by-step refinement process. This process can be used for designing data structures, required software architecture, source code and ultimately, performance algorithms. Overall, the data organization may be defined during requirement analysis and then refined during data design work

2.Architecture:



3.Architecture Description:

3.11. Data from User:

Here we will collect data from user such as books that user wants to read in pdf formats. Listening to audiobooks can help bridge the gap between decoding words and assigning meaning. Receiving information both visually and audibly reinforces word recognition, improves fluency, builds vocabulary and supports the development of comprehension skills. So that user gets book in audio format.

3.12. Data Transformation:

In the Transformation Process, we will convert our original data which is in pdf format to audible Format. And after that user can listen to the book.

4.Logging:

We should be able to log every activity done by the user.

- The System identifies at what step logging required
- The System should be able to log each and every system flow.
- Developers can choose logging methods. You can choose database logging/ File logging as well.
- System should not be hung even after using so many loggings. Logging just because we can easily debug issues so logging is mandatory to do.

5.Deployment:

Deployment We will be deploying the model to AWS.

6. Unit Test Cases

Test Case Description	Pre-Requisite	Expected Result
Verify whether the Application URL is accessible to the user	1. Application URL should be defined	Application URL should be accessible to the user
Verify whether the Application loads completely for the user when the URL is accessed	1. Application URL is accessible 2. Application is deployed	The Application should load completely for the user when the URL is accessed
Verify whether user is able to give input fields	1. Application is accessible 2. User is giving input to the application	User should be able to see input
Verify whether user gets Submit button to submit the inputs	1. Application is accessible 2. User is giving inputs in to the application	User should get Submit button to submit the inputs