Design Document

**Digital Board Marker**

Offline Player Description

This module is designed and developed by   
Komal Shehzadi (2016-CS-178)  
  
Supervised by  
Samyan Qayum Wahla

Date: Aug 19, 2019

Contents

[1. Document Purpose 6](#_Toc17230239)

[2. Requirements Addressed 6](#_Toc17230240)

[3. Detailed Design 6](#_Toc17230241)

[3.1. Lecture Player 6](#_Toc17230242)

[3.1.1. Short Description 6](#_Toc17230243)

[3.1.2. Component Diagram 6](#_Toc17230244)

[3.2. Lecture Playlist 9](#_Toc17230245)

[3.2.1. Short Description 9](#_Toc17230246)

[3.2.2. Component Diagram 9](#_Toc17230247)

[3.3. Token Authentication 9](#_Toc17230248)

[3.3.1. Short Description 9](#_Toc17230249)

[3.3.2. Component Diagram 9](#_Toc17230250)

[4. Rules and Assumptions 12](#_Toc17230251)

[5. Module Workflow Description 12](#_Toc17230252)

[5.1. User General Flow 12](#_Toc17230253)

[5.1.1. User General Flow Diagram 12](#_Toc17230254)

[5.2. Application General Flow 13](#_Toc17230255)

[5.2.1. Application General Flow Diagram 14](#_Toc17230256)

[6. Tools and languages used 15](#_Toc17230257)

[6.1. Visual Studio Windows Form Application 15](#_Toc17230258)

[6.2. SQLite 15](#_Toc17230259)

[6.3. C# 15](#_Toc17230260)

[6.4. SQL 15](#_Toc17230261)

[7. Local Database Diagram 15](#_Toc17230262)

[8. References 15](#_Toc17230263)

Table of Figures

[Figure 1 – lecture player details 7](#_Toc17230274)

[Figure 2 – Player Screen 7](#_Toc17230275)

[Figure 3 – Start Over Button 7](#_Toc17230276)

[Figure 4 – Play Now Button 8](#_Toc17230277)

[Figure 5 – Pause Button 8](#_Toc17230278)

[Figure 6 – Elapsed Time Label 8](#_Toc17230279)

[Figure 7 – Total Time Label 8](#_Toc17230280)

[Figure 8 – Go-Back Button 8](#_Toc17230281)

[Figure 9-Exit Button 9](#_Toc17230282)

[Figure 10 – Name of Lecture Label 9](#_Toc17230283)

[Figure 11- Authentication Form 10](#_Toc17230284)

[Figure 12- username input 10](#_Toc17230285)

[Figure 13- password input 10](#_Toc17230286)

[Figure 14- Login Button 11](#_Toc17230287)

[Figure 15- User General Work Flow 12](#_Toc17230288)

[Figure 16- Application General Work Flow 14](#_Toc17230289)

Table of Tables

[Table 1 - Revision History 5](#_Toc17230384)

[Table 2 - Requirements Addressed 6](#_Toc17230385)

[Table 3 – Lecture player details 9](#_Toc17230386)

[Table 4- Authentication Form Details 11](#_Toc17230387)

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Author** | **Date** | **Version** | **Notes** |
| Komal Shehzadi | August 16, 2019 | v1.0 | Document of first stable version of DBM Offline Player |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table - Revision History

# Document Purpose

This document addresses all functionalities, Structure, Workflow and detailed overview of DBM Offline player desktop application. It describes requirements, design diagrams and structure diagrams, unit testing, programming tools and languages used in designing and developing the offline player.

# Requirements Addressed

|  |  |  |
| --- | --- | --- |
| **#** | **Requirement** | **Priority** |
|  |  |  |
| **1** | Authenticate user with website | HIGH |
| **2** | Fetch lectures data and download link in app | HIGH |
| **3** | Show refreshed lecture playlist | MEDIUM |
| **4** | Play video and audio synchronized with stopwatch | HIGH |
| **5** | About page. | LOW |
| **6** | Contact us page. | LOW |

Table - Requirements Addressed

# Detailed Design

Offline player has three major sub-modules lecture playlist, token authentication, lecture player.

## Lecture Player

### Short Description

The objective of offline player sub-module Lecture player is to downloaded lectures play offline in the application. The challenge is to move lecture video to a specific position. This application is developed in visual studio as windows form application.

### Component Diagram

Diagram of lecture player page is shown:

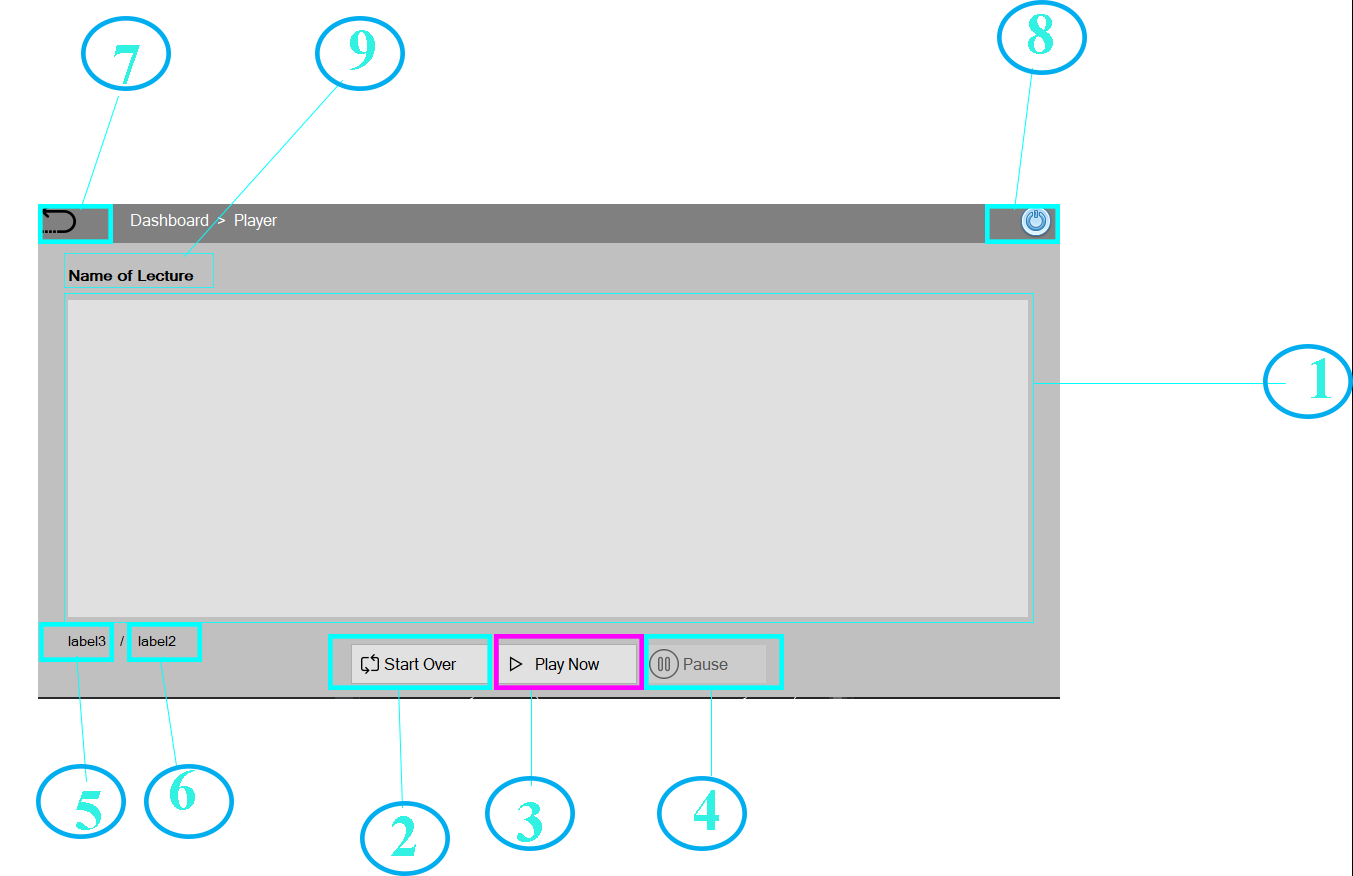


Figure – lecture player details

|  |  |  |
| --- | --- | --- |
| **Component Number** | **Description** | **Diagram** |
|  |  |  |
| **1** | **Name:** Player Screen | Figure – Player Screen |
| **Detail:** Player screen is a place where animated video drawn by data in json file will be played. |
| **2** | **Name:** Start Over Button | Figure – Start Over Button |
| **Detail:** This button is responsible for suspending all the threads except main thread and play the video, audio and timer from the start. |
| **3** | **Name:** Play Now Button | Figure – Play Now Button |
| **Detail:** This play now button starts three new threads of audio, video and timer start it is responsible to play video at start. |
| **4** | **Name:** Pause Button | Figure – Pause Button |
| **Detail:** It pauses the audio, video and timer at a specific time when they have been played. It suspends all threads which are responsible to play video for a time being. |
| **5** | **Name:** Elapsed Time Label | Figure – Elapsed Time Label |
| **Detail:** This label is responsible to show the elapsed time to the user so that he can keep track of video. |
| **6** | **Name:** Total Time Label | Figure – Total Time Label |
| **Detail:** This label shows total time of video at which it will be finished. |
| **7** | **Name:** Go-Back Button | Figure – Go-Back Button |
| **Detail:** This button takes the user to the previous screen of lecture playlist where he or she can select any other lecture to download and play. |
| **8** | **Name:** Exit Button | Figure -Exit Button |
| **Details:** As it is quite obvious by the name it exits the application. |
| **9** | **Name:** Name of Lecture Label | Figure – Name of Lecture Label |
| **Detail:** This label shows the name of lecture currently playing on the player selected by the user on previous screen. |

Table – Lecture player details

## Lecture Playlist

### Short Description

The objective of offline player sub-module Lecture playlist is to show both downloaded and not downloaded lectures record in the application. The challenge is when we don’t have any internet connectivity manage all the downloaded lectures correctly. This application is developed in visual studio as windows form application.

### Component Diagram

## Token Authentication

### Short Description

The objective of this sub module is to secure application. Token based authentication with web API will be done and records will be stored in local database. The challenge is we have to keep track of the token expiry on which we are providing privileges to the user. This sub module includes visual studio, and SQLite tools and windows form application technology with C# and SQL Languages.

### Component Diagram

The authentication page with details is as follows:

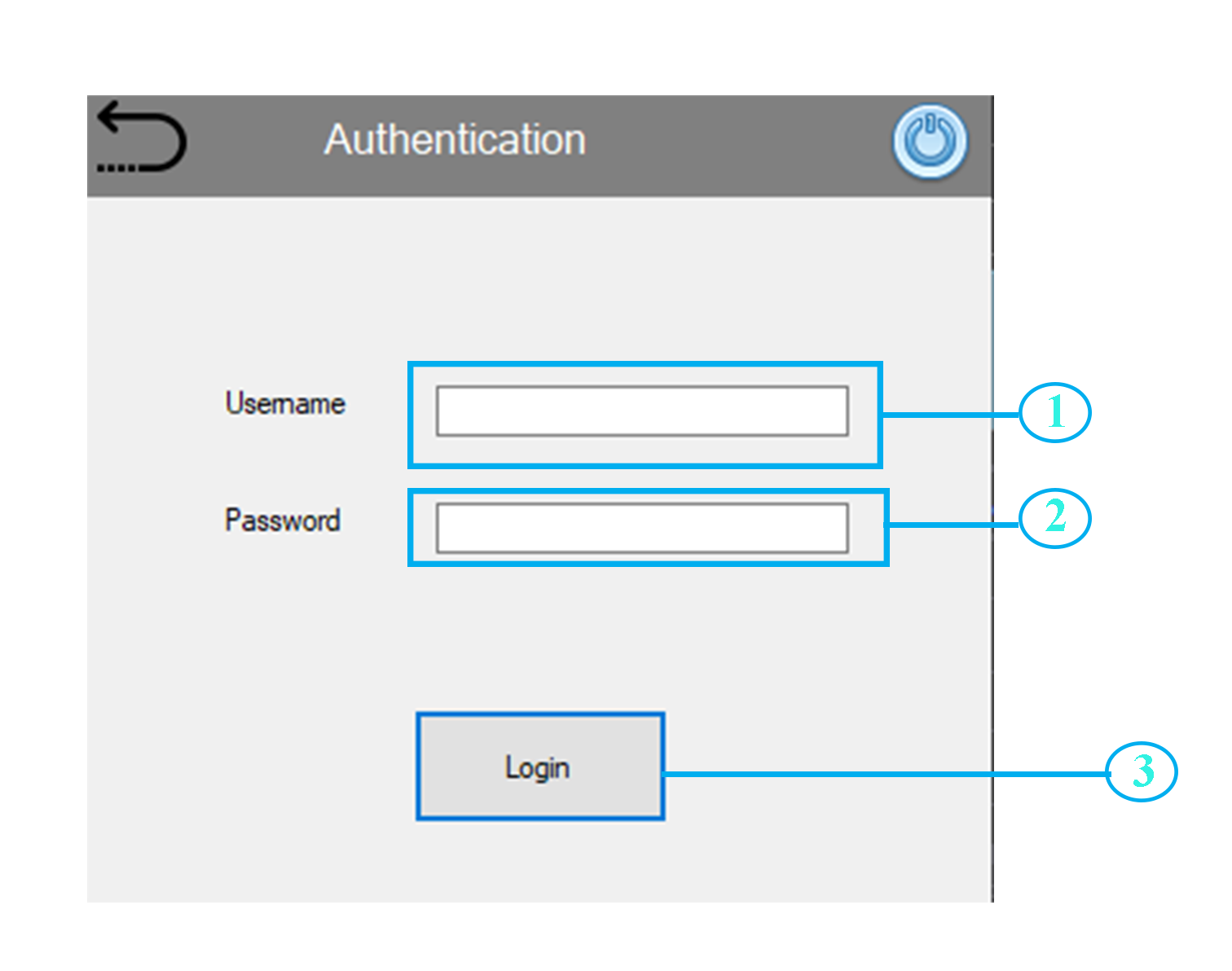


Figure - Authentication Form

|  |  |  |
| --- | --- | --- |
| **Component Number** | **Description** | **Diagram** |
|  |  |  |
| **1** | **Name:** Username Input | Figure - username input |
| **Detail:** username input takes valid username from user at the time of authentication |
| **2** | **Name:** Password Input | Figure - password input |
| **Detail:** password input takes a password corresponding to given username. |
| **3** | **Name:** Login Button | Figure - Login Button |
| **Detail:** Login button contacts with WEB API for token info for given username and password. |

Table - Authentication Form Details

# Rules and Assumptions

Following are rules and cases of assumptions that are assumed to be true while normal working

* Suitable Operating System must be installed on System such windows XP and higher.
* Suitable Version of .net framework should be installed on system otherwise we have to install app with manifest files.

# Module Workflow Description

## User General Flow

* User starts the application.
* For first time application needs authentication from user to save token information in local database.
* Once user is authenticated, he can view lecture playlist via internet connectivity.
* User can download and play lectures.

### User General Flow Diagram

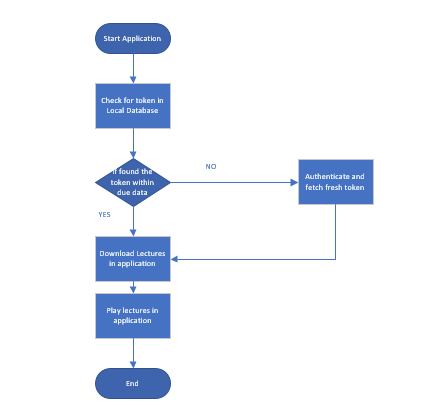


Figure - User General Work Flow

## Application General Flow

* Application at start checks in local database for user token and its expiry date if it is expired app asks for authentication again if not it proceeds.
* Application checks for internet connectivity if found it fetches new lecture records from website if not it shows previously fetched and downloaded lectures. Here downloaded lectures can be played.

### Application General Flow Diagram

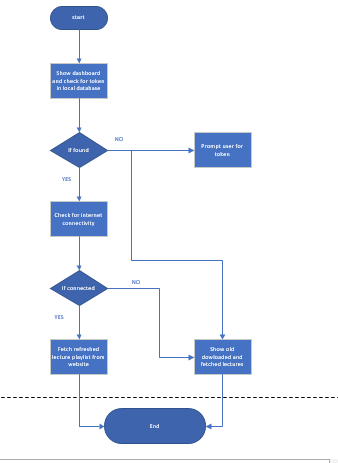


Figure - Application General Work Flow

# Tools and languages used

List all software that are used to develop and needed to operate the developed module are detailed below.

## Visual Studio Windows Form Application

Application is developed in visual studio as a windows form application.

## SQLite

Local database used in application is SQLite because it is light weighted.

## C#

Language for backend development is C#.

## SQL

SQL is used to manipulate and retrieve data from local database.

# Local Database Diagram



# References

List all documents that were used in defining the design documented in this document.