P466444

*<PRogramming III Projects>*

Product Design Specification

VERSION HISTORY

[Provide information on how the development and distribution of the **Product Design Specification**, up to the final point of approval, was controlled and tracked. Use the table below to provide the version number, the author implementing the version, the date of the version, the name of the person approving the version, the date that particular version was approved, and a brief description of the reason for creating the revised version.]

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Version #** | **Implemented**  **By** | **Revision**  **Date** | **Approved**  **By** | **Approval**  **Date** | **Reason** |
| 1.0 | *Umakant* | *7/2/2020* | *Umakant* | *7/2/2020* | Initial Design Definition draft |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

TABLE OF CONTENTS

Table of Contents

Type chapter title (level 1) 1

Type chapter title (level 2) 2

Type chapter title (level 3) 3

Type chapter title (level 1) 4

Type chapter title (level 2) 5

Type chapter title (level 3) 6

[1 Introduction 6](#_Toc26448710)

[1.1 Purpose of The Product Design Specification Document 6](#_Toc26448711)

[1.2 Document conventions 6](#_Toc26448712)

[1.2.1 Database MYSQL 6](#_Toc26448713)

[1.2.2 Windows visual studio Winforms- Microsoft's .NET Framework 4.7.2 6](#_Toc26448714)

[1.2.3 C# 6](#_Toc26448715)

[1.3 PROJECT SCOPE 6](#_Toc26448716)

[2 General Overview and Design Guidelines/Approach 7](#_Toc26448717)

[2.1 PRODUCT PERSPECTIVE 7](#_Toc26448718)

[2.1.1 User Information: 7](#_Toc26448719)

[2.1.2 Hashing: 7](#_Toc26448720)

[2.1.3 User custom file storage 7](#_Toc26448721)

[2.2 USER CLASS and CHARACTERISTICS 7](#_Toc26448722)

[2.2.1 Register 7](#_Toc26448723)

[2.2.2 Login 7](#_Toc26448724)

[2.2.3 Mediaplayer page 7](#_Toc26448725)

[3 Architecture Design 8](#_Toc26448726)

[3.1 Hardware Architecture 8](#_Toc26448727)

[3.1.1 Minimum system requirement 8](#_Toc26448728)

[3.1.2 RECOMMENDED 8](#_Toc26448729)

[3.1.3 Installation requirements? 9](#_Toc26448730)

[3.2 Software Architecture 9](#_Toc26448731)

[3.2.1 Operating system 9](#_Toc26448732)

[3.3 Security Architecture 9](#_Toc26448733)

[3.4 Communication Architecture 10](#_Toc26448734)

[4 System Design 11](#_Toc26448735)

[4.1 Database Design 11](#_Toc26448736)

[4.2 Application Program Interfaces 11](#_Toc26448738)

[4.3 User Interface Design 12](#_Toc26448740)

[4.3.1 Login 12](#_Toc26448741)

[4.3.2 Media player 12](#_Toc26448743)

[5 Product Design Specification Approval 13](#_Toc26448744)

# Introduction

## Purpose of The Product Design Specification Document

This document includes all the function and specifications with their explanations to solve related problems. It will explain the purpose and feature of the system, the interface of the system, what does the system do and constraints under which it must operate.

This document is intend for both users and developers to understand how it works and how the system should go.

## Document conventions

This document uses the following conventions

### Database MYSQL

MySQL is an Oracle-backed open source relational database management system (RDBMS) based on Structured Query Language (SQL).

MySQL enables data to be stored and accessed across multiple storage engines, including InnoDB, CSV, and NDB. MySQL is also capable of replicating data and partitioning tables for better performance and durability. MySQL users aren't required to learn new commands; they can access their data using standard SQL commands.

### Windows visual studio Winforms- Microsoft's .NET Framework 4.7.2

Microsoft's .NET Framework 4.7.2 provides a programming guidelines that can be used to develop a wide range of applications–––from web to mobile to Windows-based applications. The .NET framework can work with several programming languages such as C#, VB.NET, C++ and F#.

### C#

C# (C-Sharp) is a programming language developed by Microsoft that runs on the .NET Framework. C# is used to develop web apps, desktop apps, mobile apps, games and much more.

## PROJECT SCOPE

The purpose of this project and the document is to design and build an online Media player application that allows user to register and login, save file to the database from client side and display file to the list box so the next time when the user logged in to the Media application they gain access to the existed Media file again.

The system will be composed into two different side. Server side and client’s side. The server side component will be using MySQL database which manages that allows admin to delete user and manage their file storage. The Client side component will be a graphical interface which has all the core functions and features for a Media player. The application will guide user to add all the Media files from local path, and supports user with a quick search function that allows user to find the Media they wanted with in a second.

# General Overview and Design Guidelines/Approach

This section describes the principles and strategies to be used as guidelines when designing and implementing the system.

- Must contain dynamic data structures

- Must contain hashing techniques

- Must contain sorting algorithm

- Must contain searching technique

- Must contain 3rd party library

- Must have a GUI - Must adhere to coding standards

## PRODUCT PERSPECTIVE

Media player application database stores the following information.

### User Information:

This includes username and salted password.

### Hashing:

This include salt and password that has been salted. Using salt we can generate the correct salted password for system to validate and verify.

### User custom file storage

User stores Media direct file path inside this column

## USER CLASS and CHARACTERISTICS

Media player Login application functions and features:

### Register

Enter username and correct password to register;

Method to generate random salt number and hash with password

### Login

Able to check username from MYSQL database and retrieve correct password with the salt value stored in database, with a method to validate hashed password.

### Media player page

For the first time register and login, user may needs to add Media file to the application.

Once user adds Media file to the application, user can see the name of the Media and they gain access by click selected output box and press play button.

#### Functions & features

#### Functional requirements

* User log in page
* Hash password and validation password functions
* Play next song &Play previous song
* Sorting list
* Search song
* Save and retrieve information from Database

#### Non-functional requirements

* User Interface design
* Option for Mp4 file
* Display user name at bottom panel of the Media player
* Tool tips

# Architecture Design

This section outlines the system and hardware architecture design of the system that is being built.

Once the user enters the correct username and password, user information will be stored in the separate class which allows user information passed to all forms and application. Once the user information is correctly stored in the class, the login form will be closed and shows the second form: Media interface. The user name is displayed at the top of the navigation bar which addresses that the user is correctly logged in.

[Describe the system architecture, how the application interacts with other applications. Not necessarily how the application itself works but, how the appropriate data is correctly passed between applications.]

## Hardware Architecture

The hardware requirement is listed below:

### Minimum system requirement

**OS:** Windows Vista, Windows 7 SP1, Windows 8/8.1, Windows 10 (32/64bit versions)

**Processor:** 2 GHz processor

**Memory:** 2 Gigabytes of RAM

**Graphics**: NVidia GeForce GTS460

### RECOMMENDED

**OS**: Windows 7 SP1, Windows 8/8.1, Windows 10 (32/64bit versions)

**Processor**: Intel Core 2 Quad Q6600 @ 2.4 GHz or AMD Athlon II X4 620 @ 2.6 GHz

**Memory**: 4 GB RAM

**Graphics:** NVidia GeForce GTX 660

**Sound** **Card**: DirectX Compatible Sound Card with latest drivers

### Installation requirements?

This is a simple application that should be able to run in most computers and laptops. Consider it does not have any specific hardware detecting methods while installing it. I may write a notification document (such as Text file saying the minimum system requirements to download this application and if your Pc is under the minimum requirements it will slow your pc run time).

## Software Architecture

Following are the software used for the Media player application.

### Operating system

For Microsoft Windows:

Windows Vista (2006) Windows 7 (2009) Windows 8 (2012) Windows 8.1 (2013) Windows 10 (2015)

Not supporting MACOS: Need to download and install Window Media player to be able to launch this application.

#### Database MySQL:

To save the user information and records, I have chosen SQL database.

#### .NET Framework

To implement the project we have chosen .NET Framework 4.7.2 for its more interactive support.

## Security Architecture

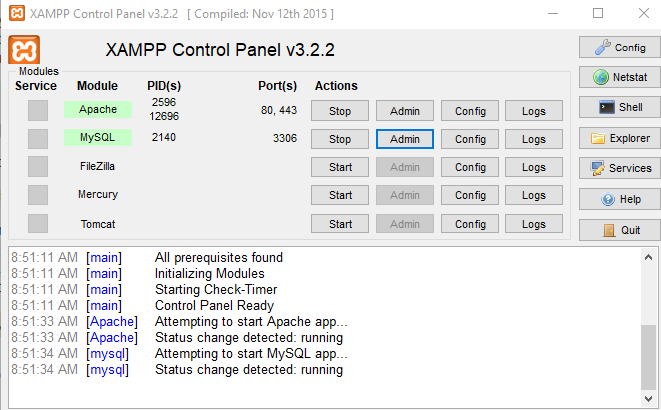
Security systems need database storage just like many other applications. To control the security of the users we hashes the password before they saved into the database. And to ensure that there is no threats can know the basic inquiries of users, we have physical system security procedures in place to delete all customer personal information we collect.

If there is extensive damage to a wide portion of the database due to catastrophic failure, such as a disk crash, the recovery method restores a past copy of the database that was backed up to archival storage (typically tape) and reconstructs a more current state by reapplying or redoing the operations of committed transactions from the backed up log, up to the time of failure.

## Communication Architecture

The application requires Transmission Control Protocol (TCP)/IP connection method to access the database.

When using MYSQL you need to launch xampp control panel first and start Apache before mysql database.



## USEr Architecture

1. Front-end software: Winform application - .NET Framework 4.7.2
2. Back-end software: MYSQL

# System Design

## Database Design

## 

## Application Program Interfaces

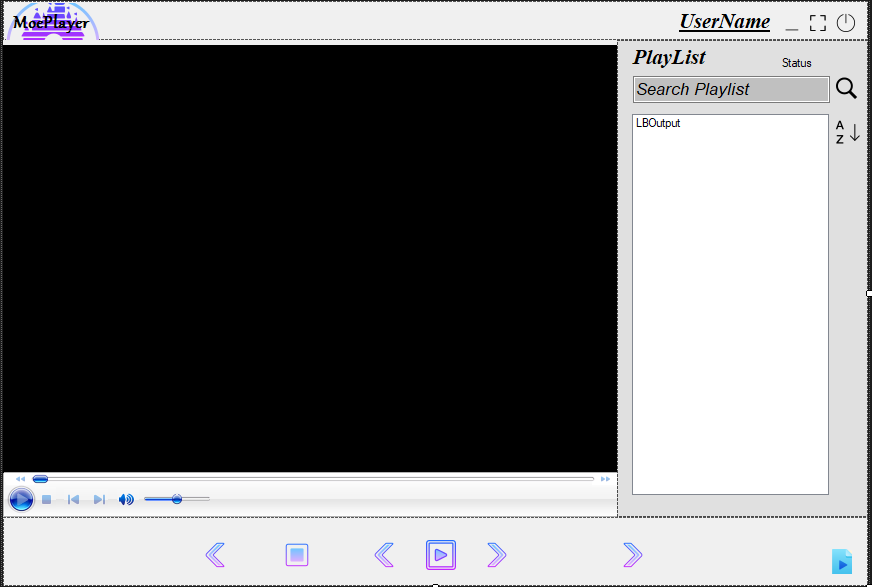
## 

## User Interface Design

### Login

## 

### Media player



# Product Design Specification Approval

The undersigned acknowledge they have reviewed the AT3 *Project* **Product Design Specification** document and agree with the approach it presents. Any changes to this Requirements Definition will be coordinated with and approved by the undersigned or their designated representatives.

[List the individuals whose signatures are required. Examples of such individuals are Business Steward, Technical Steward, and Project Manager. Add additional signature lines as necessary.]

|  |  |  |  |
| --- | --- | --- | --- |
| Signature: |  | Date: |  |
| Print Name: |  |  |  |
| Title: |  |  |  |
| Role: |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Signature: |  | Date: |  |
| Print Name: |  |  |  |
| Title: |  |  |  |
| Role: |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Signature: |  | Date: |  |
| Print Name: |  |  |  |
| Title: |  |  |  |
| Role: |  |  |  |

Appendix A: References

[Insert the name, version number, description, and physical location of any documents referenced in this document. Add rows to the table as necessary.]

The following table summarizes the documents referenced in this document.

|  |  |  |
| --- | --- | --- |
| **Document Name and Version** | **Description** | **Location** |
| *1.0* | *Software Requirements Specification document with example* | <https://krazytech.com/projects/sample-software-requirements-specificationsrs-report-airline-database> |