

Software Design Description Document

Video and Audio Sharing Portal Project

Version 1.0

Prepared by

Ali Gökalp Peker

Deniz Koçak

Middle East Technical University

Computer Engineering Department

1.Introduction.....	3
1.1.Document Overview	3
1.2. Scope.....	3
1.3.References.....	3
1.4.Definitions and Acronyms.....	3
2.Decomposition Description	4
2.1.Module Decomposition	4
3.Interface Description	6
3.1.Module Interface.....	6
3.2.Data Interface	8
4.Detailed Design.....	8
4.1.Class Diagram	8
4.2.Package Dependency	9
4.3.Sequence Diagram	10
5.Development Process.....	14
Appendix A	14

1.Introduction

1.1.Document Overview

This document describes design elements for multimedia software project.Document includes modules of software and dependency relations between them.Also SDD mentions about interface descriptions and detailed design of software.And used standards will be described.

1.2. Scope

This contains complete description of the online Multimedia Management System. The basic architecture for this system contains client-server and also multi-tier paradigms.

Users can register new accounts, modify existing ones, upload, play, share, rate and comment on media files using right their web browsers.

1.3.References

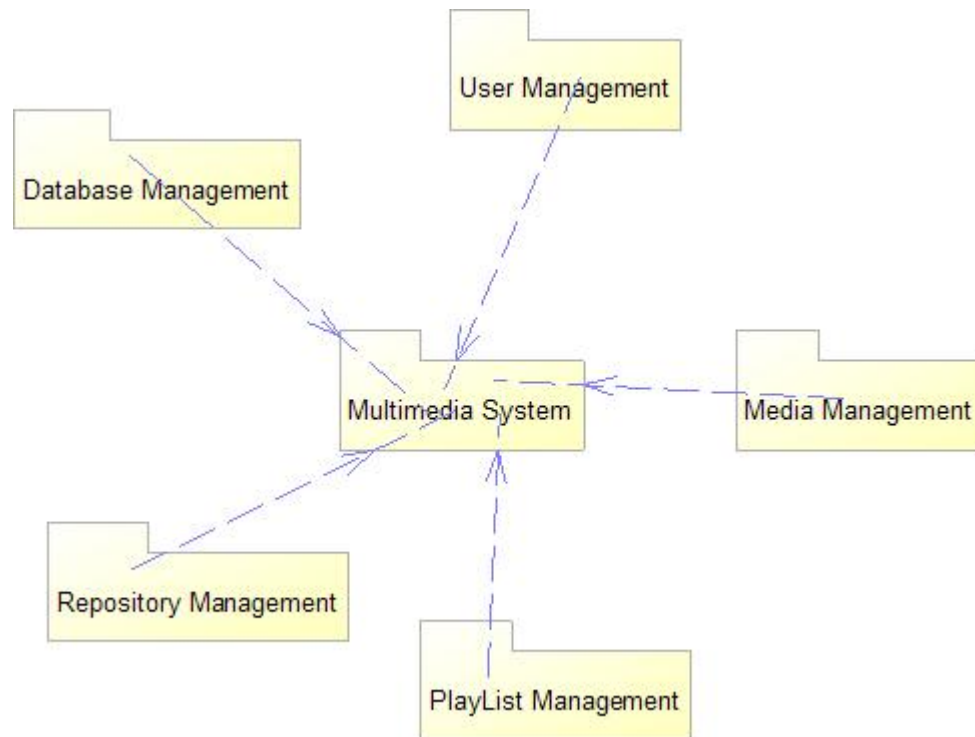
[IEEE-SA]IEEE Std 1016-1998 , IEEE recommended Practice for Software Design Descriptions by IEEE-SA Computer Society (IEEE-SA Standards Boards, 1998)

1.4.Definitions and Acronyms

SDD	Software Design Description
IEEE	Institute of Electrical and Electronic Engineers
ORM	Object Relational Mapping
DBMS	Database Management System

2.Decomposition Description

2.1.Module Decomposition



Decomposition diagram represents the functional decomposition of the system. Module details explained below.

Name:	• User Management
Type:	• Module
Purpose:	• This entity is responsible for the management of users. Adding, deleting a user or modifying user credentials are basic tasks of this module.
Functions:	• Registration of new users. This module is able to check user name and e-mail to prevent conflicting users on the same system. • Checking user login info is valid

Name:	• Database Management
Type:	• Module

Purpose:	This entity is responsible for the initialization of the database and communication between the other modules and system database. It is a bridge between database and the rest of the system.
Functions:	- Initialize database with proper parameters. This module saves updated information about database like host name, user name, password. It isolates the database information from the rest of the system. - Execute the queries coming from other modules, processes and return the results to the client. This modules is able to check the validity of the given query and modify it if necessary.

Name:	Media Management
Type:	Module
Purpose:	This entity is responsible for the management of media files stored on the repository.
Functions:	- Playing of media files. - Commenting and rating existing media files in the repository.

Name:	Play List Management
Type:	Module
Purpose:	This entity is responsible for the management of playlist for each user.
Functions:	- Creation of new playlists. This module is able to check the name of new playlist to prevent name collisions and warn user in case of a collision. - Removal of existing playlists. - Adding and removing media files to/ from the playlists.

Name:	Repository Management
Type:	Module
Purpose:	This entity is responsible for retrieving media, uploading a media to repository and searching the repository with

	respect to the given file, album or artist name.
Functions:	• Retrieves media with given media id • Uploads media with given media id • Perform searches with respect to the keyword entered by the user and return the results. This entity is able to check if the given keyword is valid for a database query.

3.Interface Description

3.1.Module Interface

Name:	Repository Management	
Function:	Management of media files in the repository	
Interfaces:	Name	retrieveMedia
	Parameters	media : Media
	Returns	Media
	Pre-condition	None
	Post-condition	None
	Name	searchMedia
	Parameters	mediaName : String albumName : String artistName : String
	Returns	mediaList
	Pre-condition	None
	Post-condition	None

Name:	Database Management	
Function:	Management of DBMS and communication between system modules/processes and DBMS.	
Interfaces:	Name	connect
	Parameters	hostname : String username : String password : String port : Integer

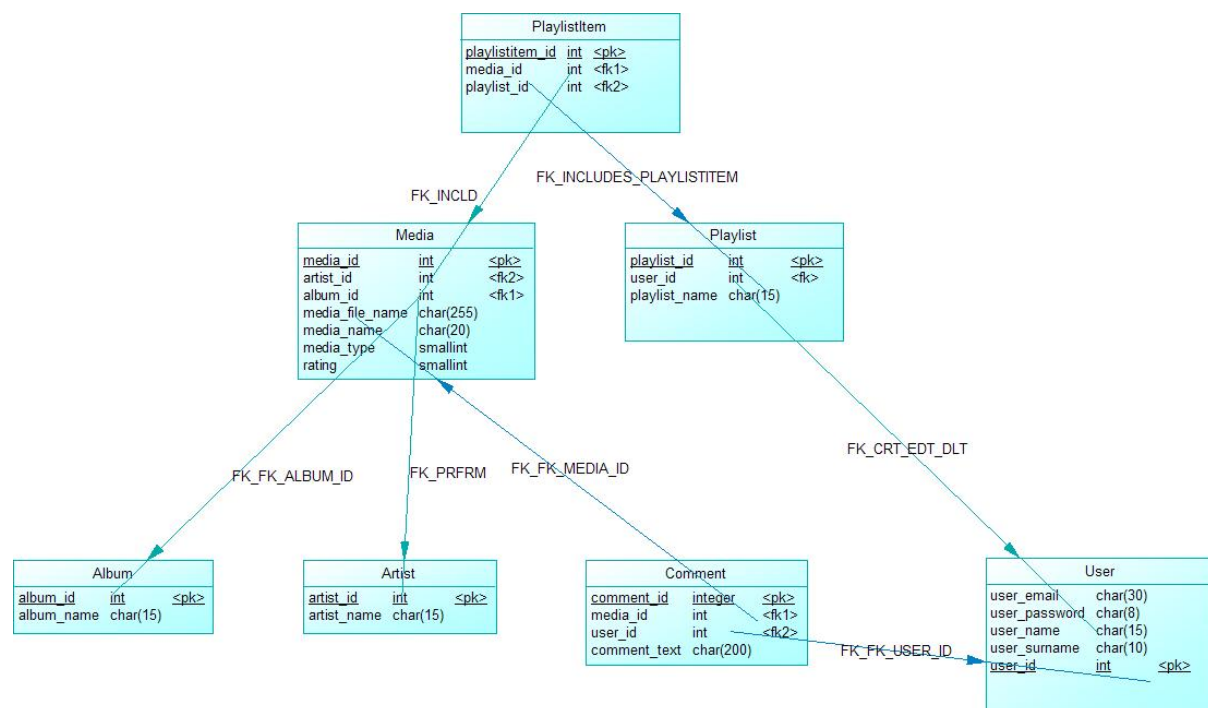
	Returns	Boolean
	Pre-condition	None
	Post-condition	Database connection shall be completed when return TRUE
	Name	disconnect
	Parameters	None
	Returns	Boolean
	Pre-condition	None
	Post-condition	Database connection shall be completed when return TRUE
	Name	executeQuery
	Parameters	query : String
	Returns	Boolean
	Pre-condition	connect must be called and return TRUE
	Post-condition	Given query shall send to the DBMS

Name:	Media Management	
Function:	Performs the actions on a media file like playing, rating commenting a media.	
Interfaces:	Name	playMedia
	Parameters	media : Media
	Returns	Boolean
	Pre-condition	None
	Post-condition	None
	Name	rateMedia
	Parameters	media : Media
	Returns	Boolean
	Pre-condition	None

	Post-condition	None
	Name	commentMedia
	Parameters	media : Media comment : String user : User
	Returns	Boolean
	Pre-condition	None
	Post-condition	None

3.2.Data Interface

As a primary concept, software uses database management system for data management. Therefore related data model is prepared so physical data design can be mentioned as data interface. Database creation script can be found in Appendix A.



4.Detailed Design

4.1.Class Diagram

Main architecture diagram is displayed below. Data Access Objects are generated by ORM so its diagram is identical to Data Interface diagram.

User Management
+ signInNewUser (User user) : boolean
+ checkUserLogin (User user) : boolean

Repository Management
- repositoryPath : String
+ retrieveMedia (Media media) : boolean
+ uploadMedia (Media media) : boolean
+ searchMedia (String mediaName, String albumName, String artistName) : MediaList

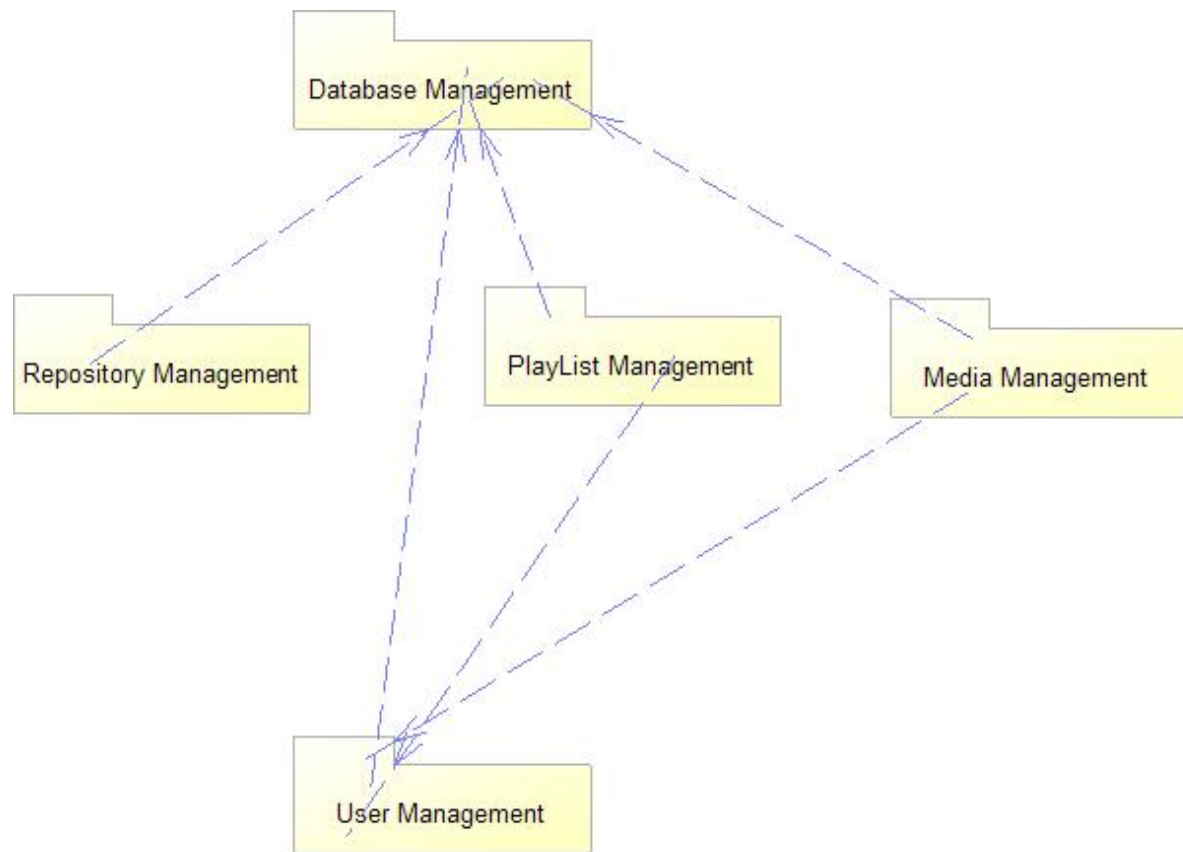
Media Management
+ playMedia (boolean media) : boolean
+ rateMedia (Media media, int rating, User user) : boolean
+ commentMedia (Media media, String comment, User user) : boolean

PlayList Management
+ createPlaylist () : void
+ deletePlayList (int playListId) : boolean
+ addMediaToPlaylist (int playListId, Media media) : boolean
+ removeMedia (int mediaID) : boolean

Database Management
+ connect (String host, int portNo, String username, String password) : boolean
+ disconnect () : boolean
+ executeQuery (String query) : boolean

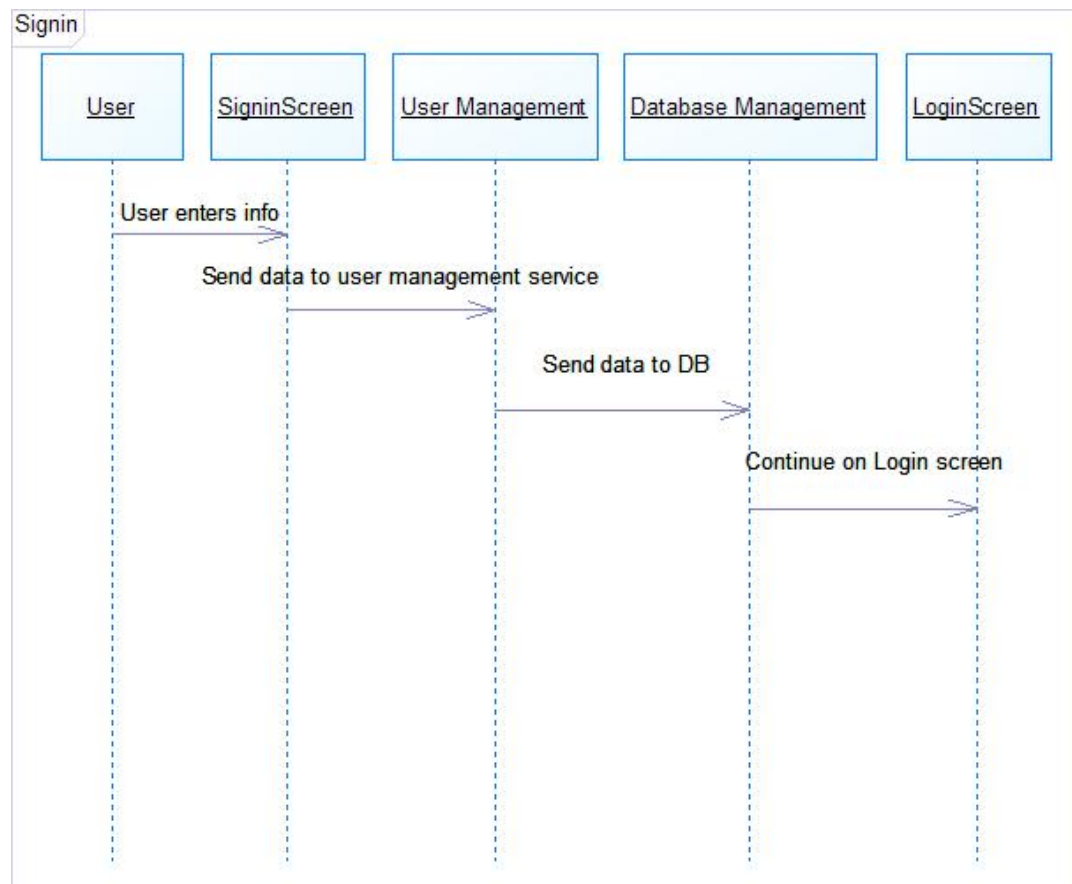
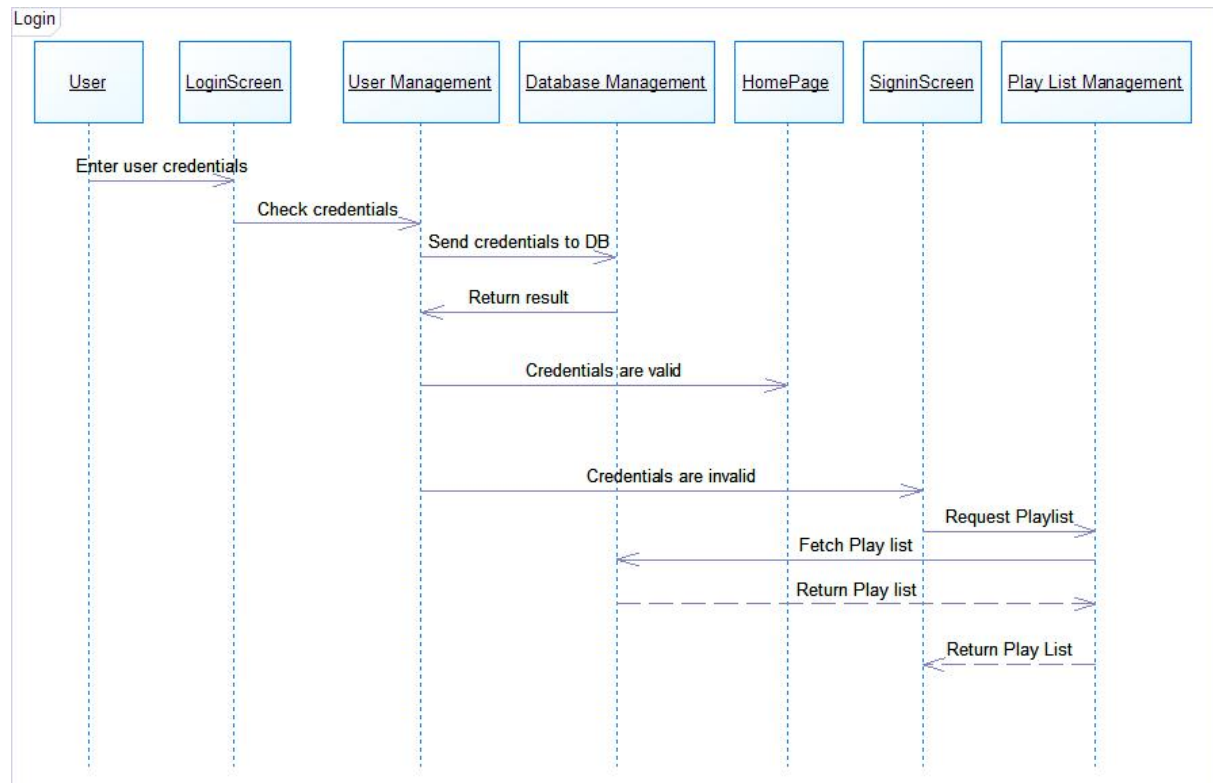
4.2.Package Dependency

Package dependency is displayed below as a link for each dependency pair.Each module is regarded as packages.

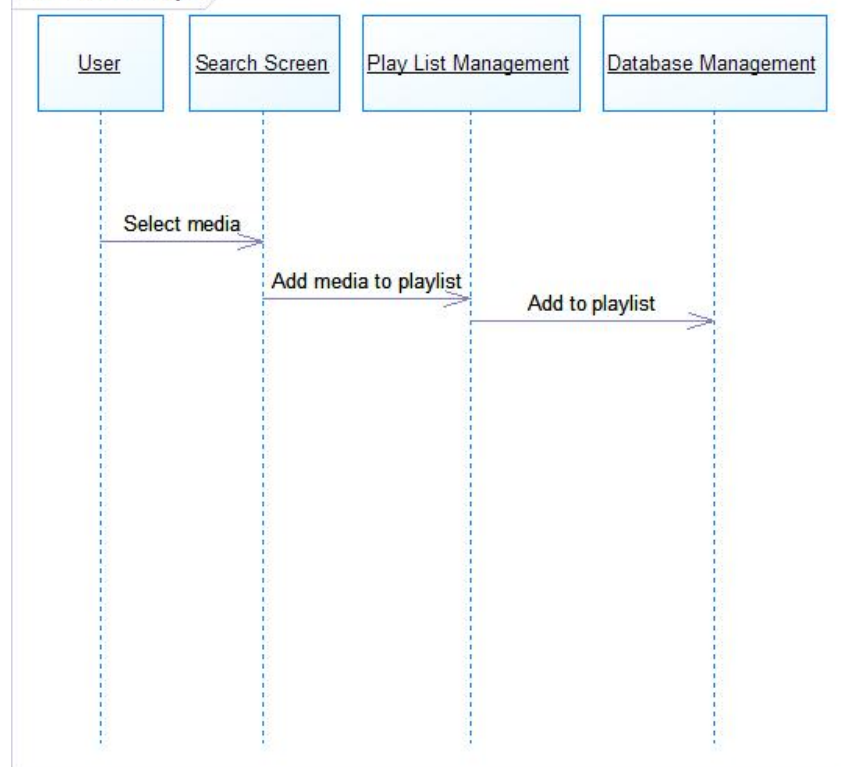


4.3.Sequence Diagram

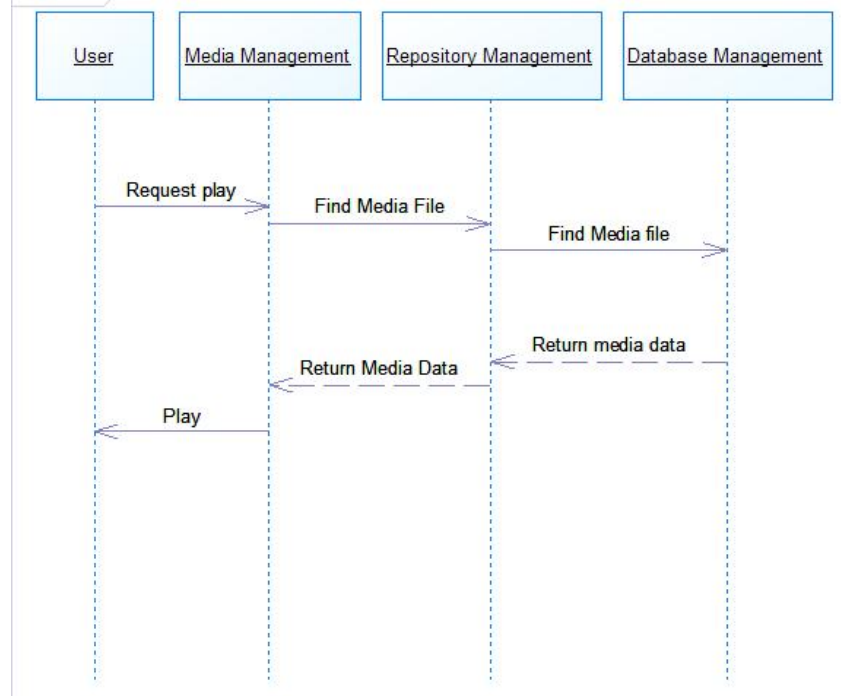
Each use-case mentioned in Multimedia System SRS document is explained as sequence diagrams.



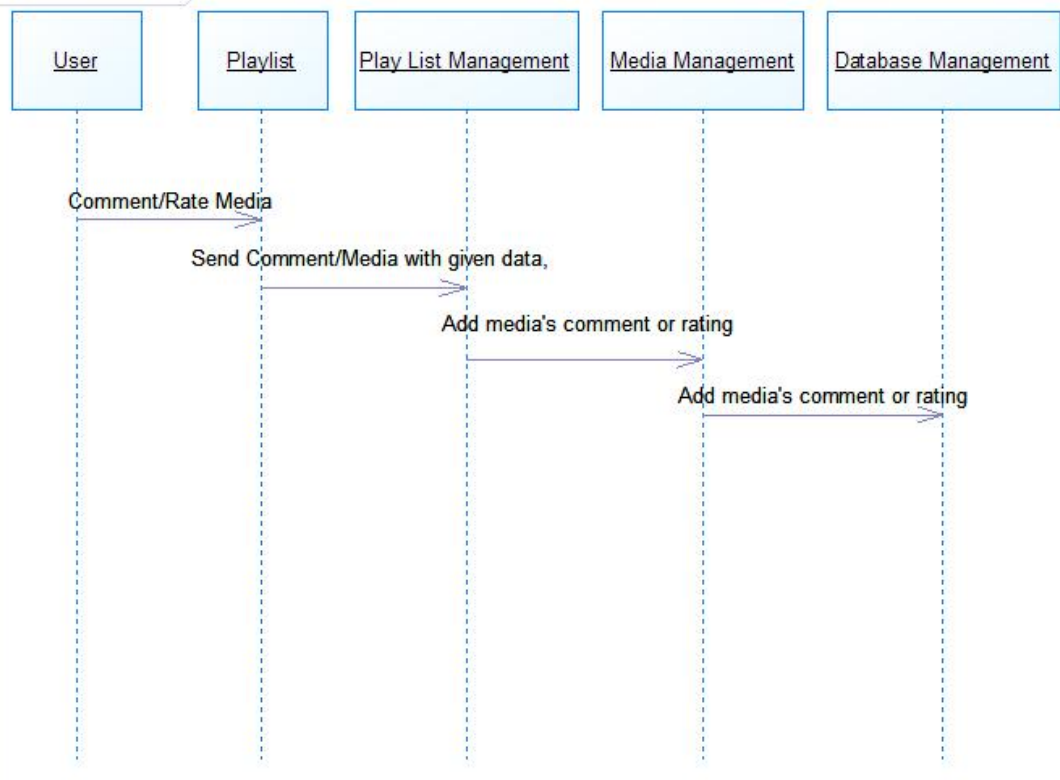
Add Media To Playlist

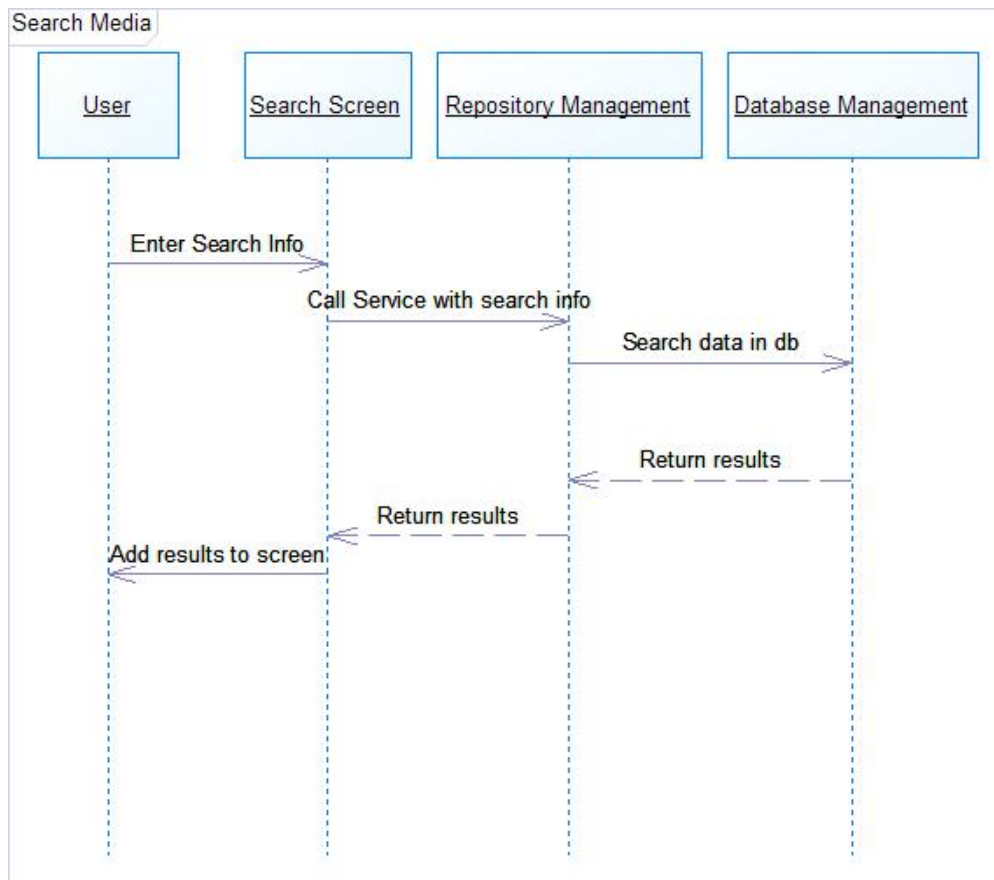


Play Media



Rate/Comment Media





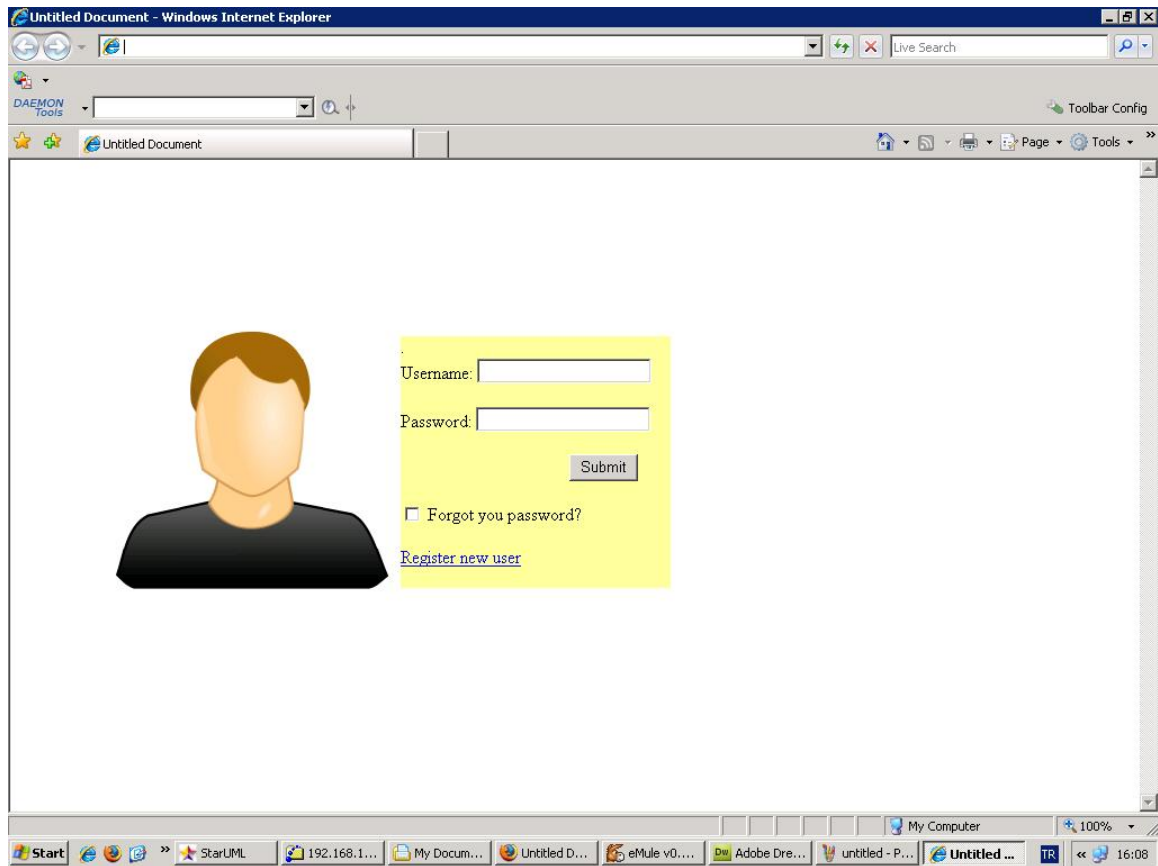
5.Development Process

Development process will be used as primary development process. However disciplines from other processes will be also used in development cycle

- Sprint methodology from [XP]
- Use standards for coding[XP]
- Simplicity[XP]

Appendix A

Login Screen



Search Screen

