



CSE 3105/CSE 3137

OBJECT ORIENTED ANALYSIS AND DESIGN

FALL 2020

COURSE PROJECT: *Media Browser Application*

Requirements Analysis Document

Group 30

İbrahim Görkem Kavas-170316034

Canan Urhan-180316008

Uğur Akçelik-180316050

Arda Fidan-180316059

Osman Alican-190316078

Muhammed Turan Körsulu-130316032

1 December 2020

Table of Contents

1	Introduction	1
2	Current System	1
3	Proposed System	1
3.1	Overview	1
3.2	Functional Requirements.....	1
3.3	Nonfunctional Requirements.....	1
3.4	System Models.....	2
3.4.1	Scenarios	2
3.4.2	Use Case Model.....	2
3.4.3	Object Model	3
3.4.4	Dynamic Models.....	3
3.4.5	User Interface Mock-ups.....	4
4	Glossary.....	8

1 Introduction

Media Browser is freeware used for playing video files, playing audio files and displaying images in a number of formats. Media Browser based on JavaFX and Media Browser uses JavaFX 2.2 Media library. Media Browser supports all operating systems which can run Java VM. Media Browser can create/manage playlist for audio and video files.

The formats currently supported are the following (docs.oracle.com) :

- Audio: MP3; AIFF containing uncompressed PCM; WAV containing uncompressed PCM; MPEG-4 multimedia container with Advanced Audio Coding (AAC) audio
- Video: FLV containing VP6 video and MP3 audio; MPEG-4 multimedia container with H.264/AVC (Advanced Video Coding) video compression
- Image: BMP, GIF, JPEG, PNG

2 Current System

The current system of is that it is a media browser with an infrastructure that can play music and display images. We will determine the functionality and additional features of system ourselves and arrange this according to the expectations of the users. Our additional features include creating playlists, music genre categorization etc. it will be.

3 Proposed System

3.1 Overview

3.2 Functional Requirements

- The system must allow the user to play music.
- When the user hit the stop button, it should be stopped.
- The system must allow the user to display images.
- The system must allow the user to create a playlist.
- When a user creates an account, the server shall send a welcome message.
- The system must allow the user to play video files.

3.3 Nonfunctional Requirements

- The program must be compatible with Windows 10 operating system.
- The program must be written in Java Programmin Language.
- The program must have a White theme.

- The visibility must be improved.

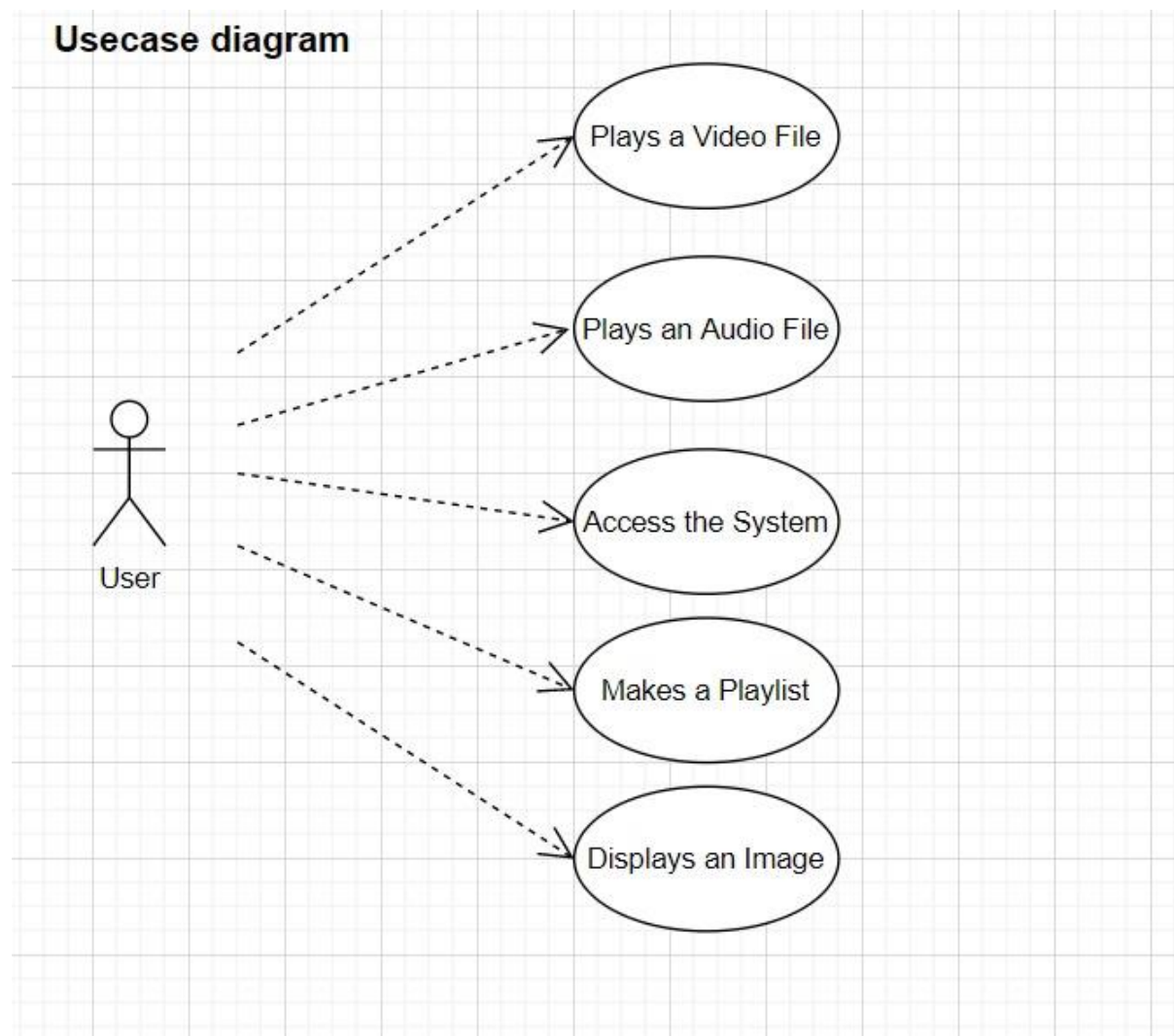
3.4 System Models

3.4.1 Scenarios

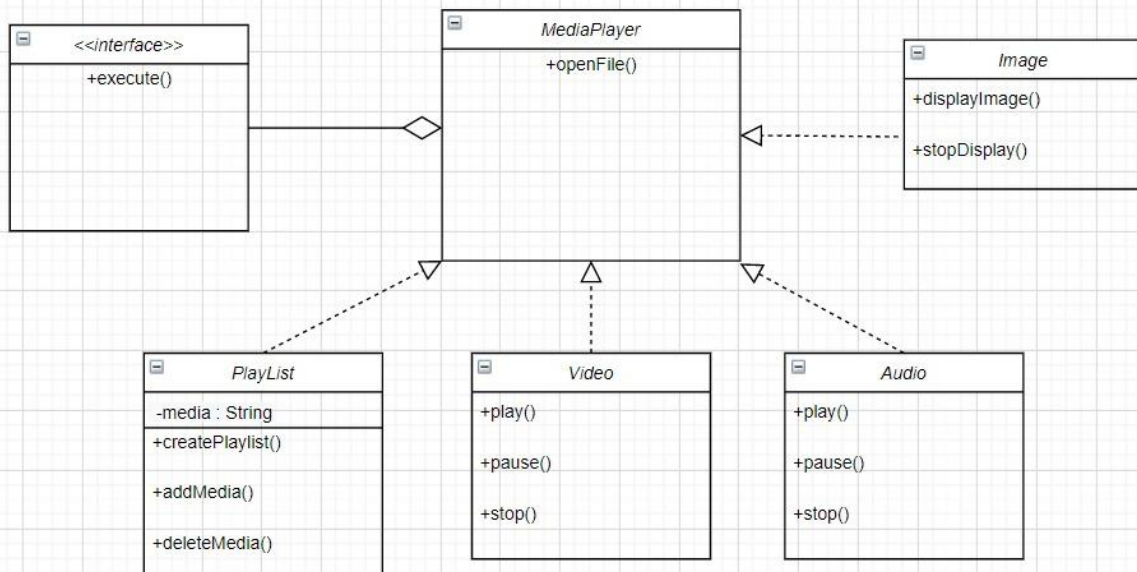
User open the computer and run the application then play the song after a while user does not like that song and user changes the song after that user decides to create a playlist that has user's favorite songs. After that user plays a video file and display images and when user is done with this application, user close the application.

3.4.2 Use Case Model

Usecase diagram



Class diagram



3.4.3 Object Model

<Object model section documents in detail all the objects we identified, their attributes, and, operations. As each object is described with textual definitions, relationships among objects are illustrated with class diagrams.>

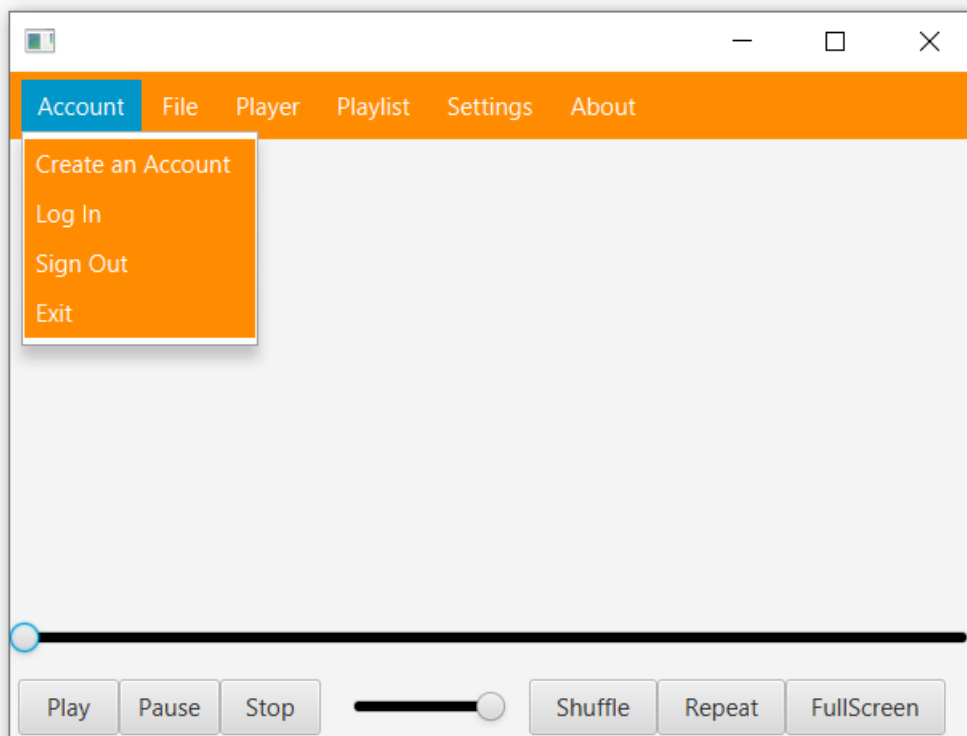
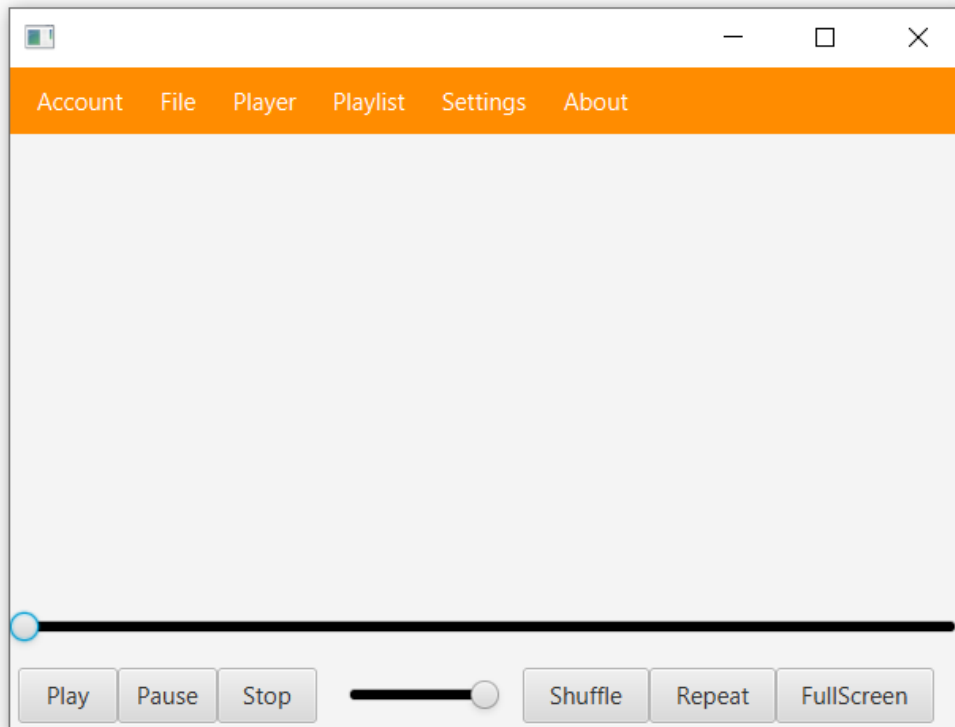
Step 5 activity

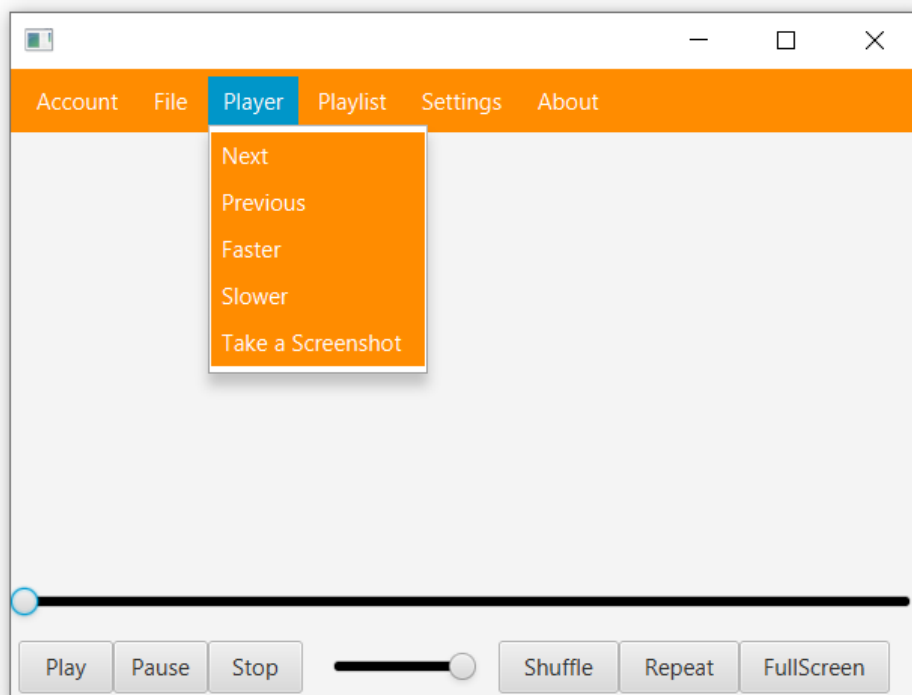
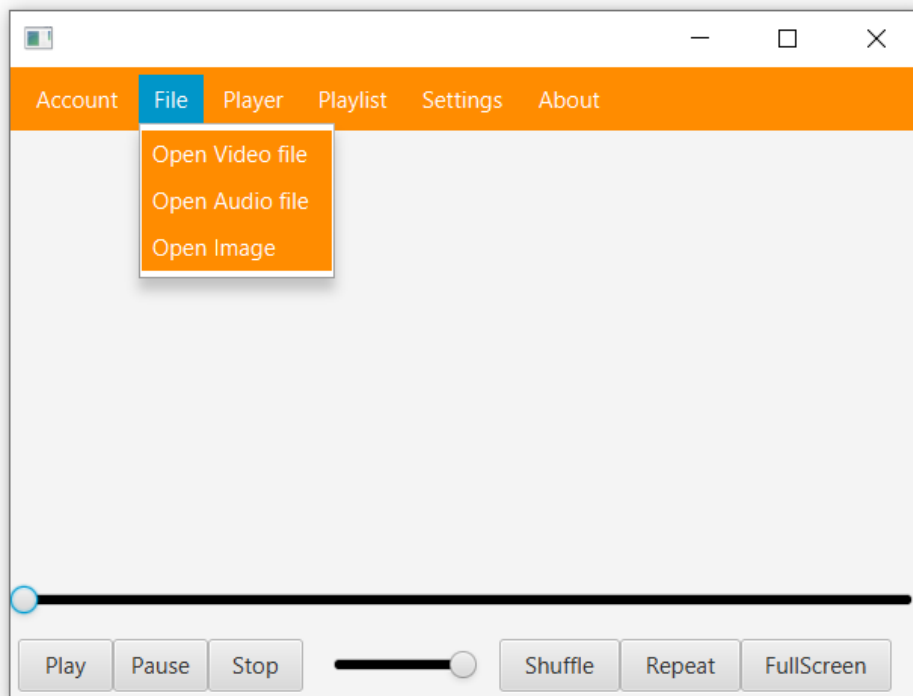
3.4.4 Dynamic Models

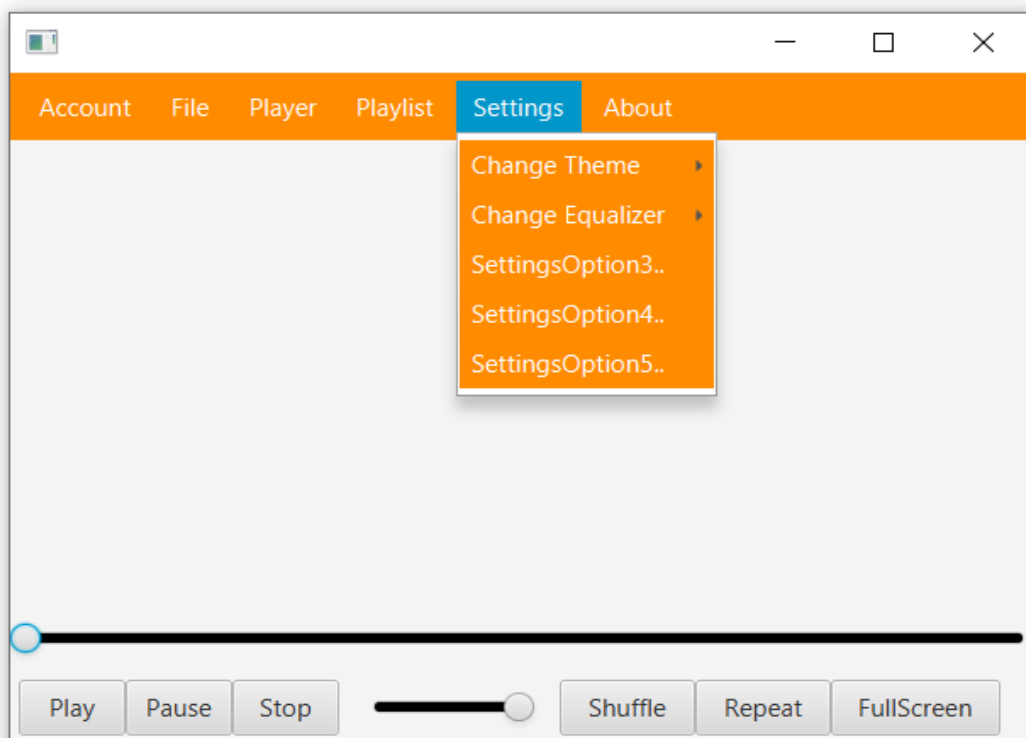
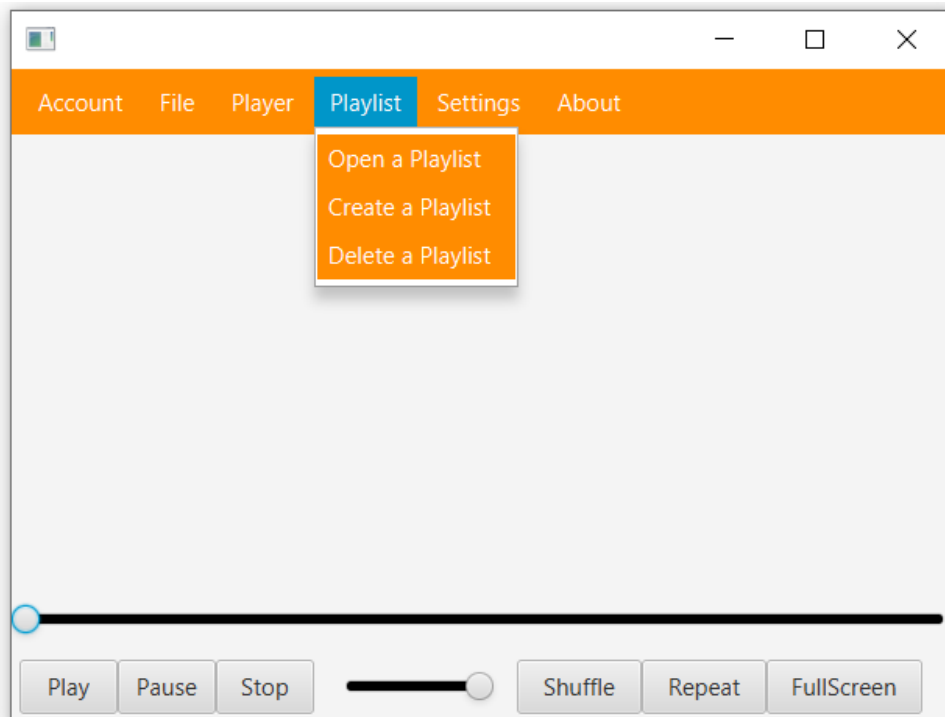
<Dynamic models section documents the behavior of the object model in terms of state machine diagrams and sequence diagrams. Although this information is redundant with the use case model, dynamic models enable us to represent more precisely complex behaviors, including use cases involving many actors.>

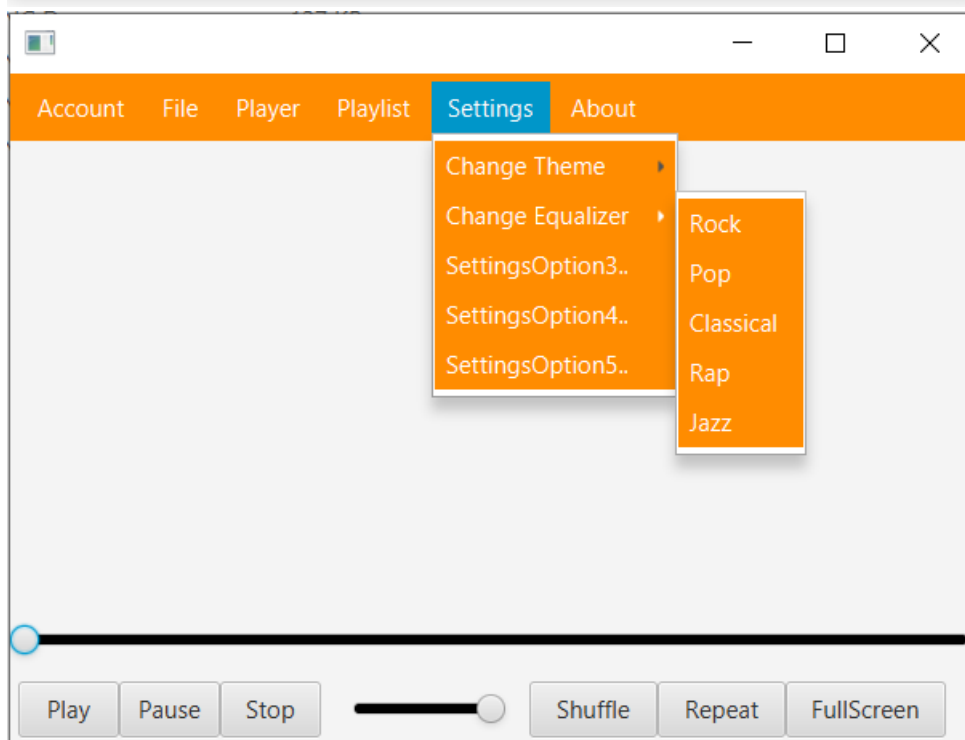
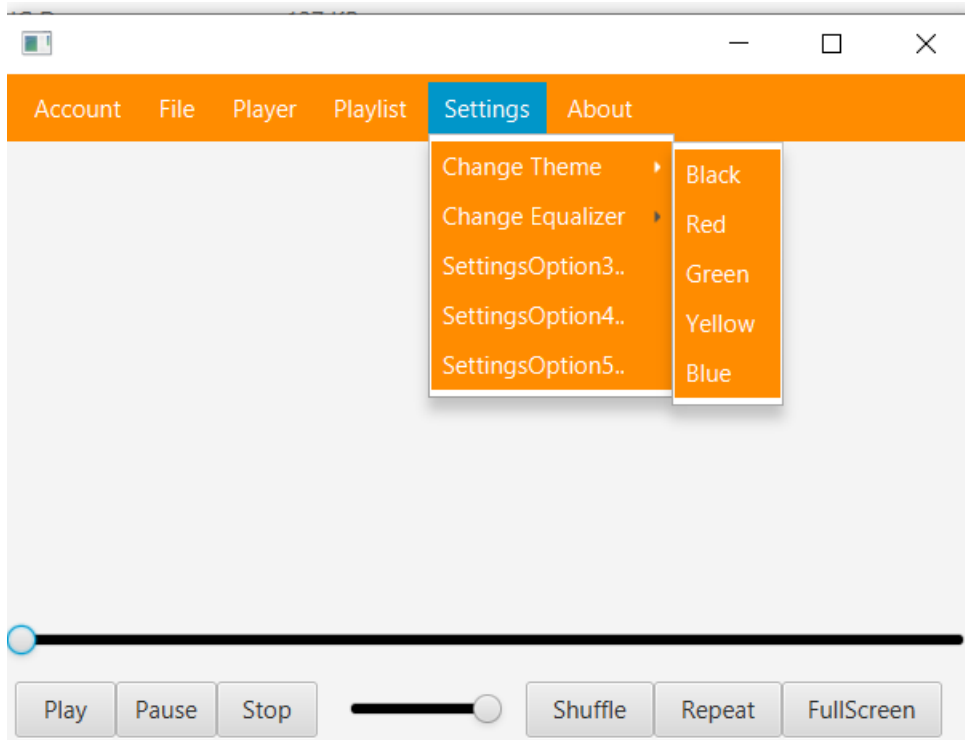
Step 5 activity

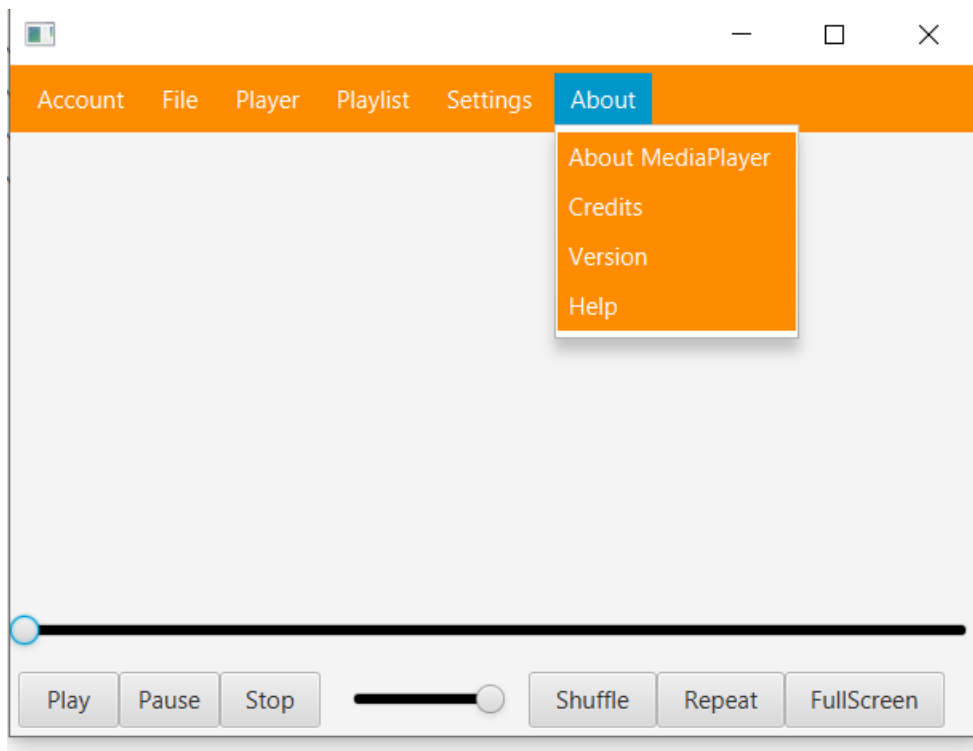
3.4.5 User Interface Mock-ups











4 Glossary

<We also maintain a glossary of important terms, to ensure consistency in the specification and to ensure that we use the client's terms. We explain the application domain concepts that need to be defined precisely, as these terms could have a different interpretation in other contexts.>

Step 3, 4 and 5 activity