



**CSE 3105/CSE 3137**

**OBJECT ORIENTED ANALYSIS AND DESIGN**

**FALL 2020**

**COURSE PROJECT: <Media Application Browser>**

***Requirements Analysis Document***

***Group 13***

*Muratcan Erek – 180316042*

*Emre Çakmak – 180316021*

*İsmet Dirilen – 180316031*

*Ali Osman Beker – 180315010*

*Mehmet Çağlar – 160316026*

*20 November 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 .....	2
3.4	System Models.....	2
3.4.1	Scenarios.....	3
3.4.2	Use Case Model.....	4
3.4.3	Object Model .....	7
3.4.4	Dynamic Models.....	7
3.4.5	User Interface Mock-ups .....	7
4	Glossary.....	7

## **1 Introduction**

The product is not ordinary and basic with hundreds of functions. People do not always feel the same in daily, sometimes they can be tired, cheerful and sometimes sad and this case, it pushes them to listen to music suitable for their mood. For this reason, our product includes mod lists. There is no accounting for tastes. People makes friends with people who are close to their interests' feelings and likes and we have developed and products that includes a sharing function to strengthen your love ties with our loved ones and friends. We all have our favorites or things we like more. For example, same of the songs can be very meaningful memorable and valuable to us. In order to easily access these songs, you can create a top list in our application and listen whenever you want.

## **2 Current System**

Our application will offer practical usage features to its users, for example, you will be able to listen to songs similar to the songs you listen to thanks to the smart song suggestion feature.

Unlike other types, it will increase the interaction between people by finding users who listen to similar song types and suggesting them to each other. These recommendations will not only increase interpersonal interaction, people will ask each other to listen to other users by sharing their favorite songs.

In our product, there are playlists created according to emotional modes, not only with them, but also playlists that you can listen to in situations such as sports, walking, in the office, according to your current mode. If you want, you can also create your own playlists and listen.

## **3 Proposed System**

### **3.1 Overview**

Project provides a practical way to explore different experience with the functions it has with these features , the system provides a musical culture.

### **3.2 Functional Requirements**

- Users can create their personal playlist.
- Users can create a playlist according to the desired mode.
- Users can see that most popular musics.
- User can share to listened musics.

### **3.3 Nonfunctional Requirements**

#### **Usability**

- There are 2 Type login operation,Top musics shouldn't play with default package On the other hands,it must be Access with Premium package.
- The system must ensures the continuity of the music after the music that the user is listening to is finished.
- The user must be able to download the playlists she has created with her favorite songs and the music lists that are already available.

#### **Reliability**

- Premium package must be return in users have any problem within 1 month.
- All information must kept in real-time backup for a possible hacking situation.
- Unless the user should not gives permission, the songs he listened to and the playlist he created are kept confidential as per the law on protection of personal information.

#### **Performance**

- It shouldn't be delay while add musics in the system(adding and removing data should be fast)
- When the system music database reaches the limit, the database must be able to divided into sub-parts so that the system is not affected by the slowdown.

#### **Supportability**

- Operation should be used to simple.
- nterface should be simple and easily accessible
- The system should be able to statc interface the toplist and modlist requested by other users.

### **3.4 System Models**

### 3.4.1 Scenarios

**Scenario Name :**Music by instant mode

**Participating,Actor,Instances:**Kane(User),Mike(Admin)

#### **FLOW OF EVENTS:**

1-Kane argue with his mother beacuse of he don't his homework and retreats own room.He start to search entertaining music by he lying to bed for fixed mood and then Kane start to listen to entartaining music mood of kane is fall when he hear to sad music.Therefore,send your complaint to customer service.

2-Mike takes the compalint into evaluation.After evaluation, it provides feedback by updating the complaint received from user

3-Mike creates to modlist function to the system according to Kane's request and he wants to uptade to system from kane

4-After Kane uptade the system. Now,it can access any musics according to his instant mode

**Scenario Name :**Playlist Category Of Favorite Music

**Participating,Actor,Instances:**Joe(User)

#### **FLOW OF EVENTS:**

1-Joe is bored to search one by one that between different categories his likes musics.

2-Therefore,He wants to collect to playlist what he calls own playlist to using playlist function.

3-Searches by category or manually using select song button in the playlist it creates.

4-He is shape to reorganization of playlist with adding and removing transactions.

5-Thus, Joe can listen to his favorite musics together with he calls own playlist.

**Scenario Name :**Follow of Agenda

**Participating,Actor,Instances:**Alice(User),Joe(Admin)

### FLOW OF EVENTS:

1)Alice spends her free time listening to music. Alice thinks she's relieved by music. one day, while listening to songs again, Alice realizes that she always listens to the same songs and is not aware of the agenda.

2)Thus, Alice prepares a complaint report to be aware of the agenda and newly released songs and sends this report to the Joe

3)Joe examines the complaint and request report sent by Alice

4)After the sent report is examined, it is processed according to Alice's wishes and desires.

### 3.4.2 Use Case Model

#### 1- USE CASE NAME: MODLIST

#### 2-PARTICIPATING ACTOR: CUSTOMER , ADMIN

#### 3-ENTRY CONDITION:

- Admin views report sheet.

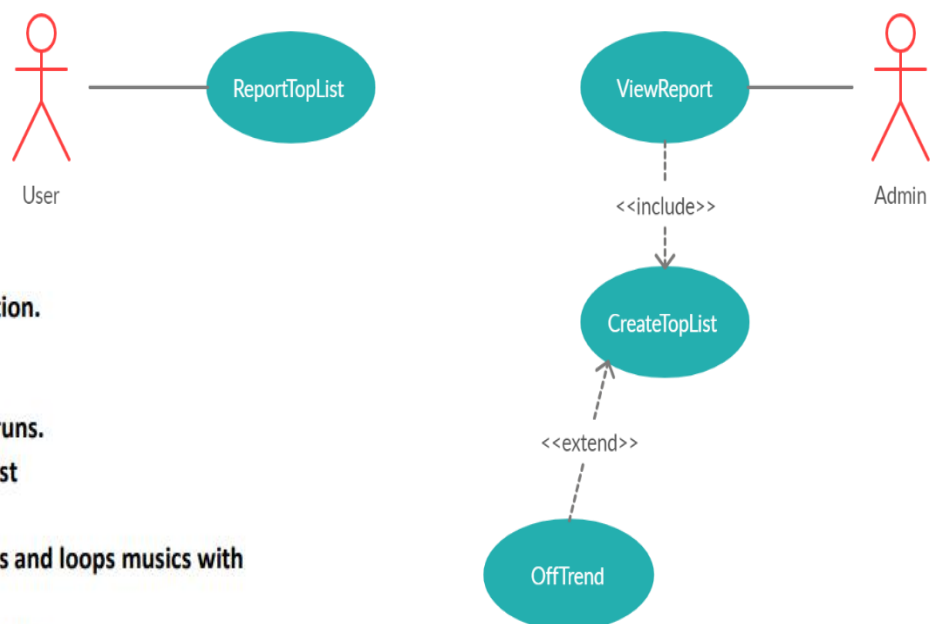
#### 4-EXIT CONDITION:

- Admin adds modlist function in application.

#### 5-FLOW OF EVENTS:

- Admin creates to modlist function and runs.
- Admin collect to selected all musics in list
- Admin names all musics by mode
- Admin creates list for same types musics and loops musics with the same mod type
- Admin adds modlist function in application.

#### 6-EXCEPTIONS:



### 1- USE CASE NAME: TOPLIST

### 2-PARTICIPATING ACTOR: CUSTOMER , ADMIN

### 3-ENTRY CONDITION:

- Admin views report sheet.

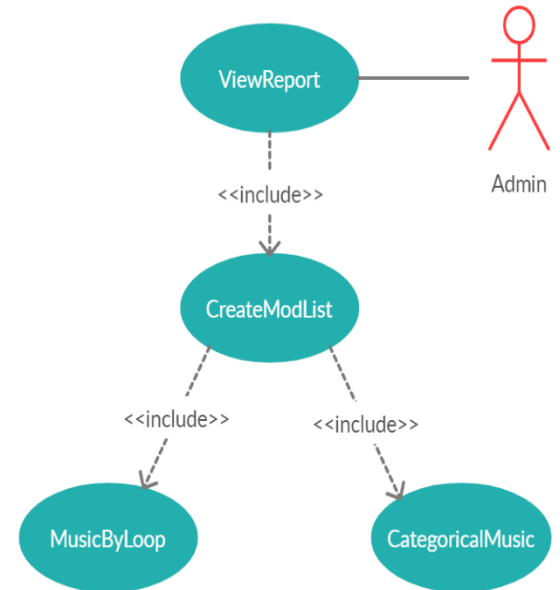


### 4-EXIT CONDITION:

- Admin adds toplist function in system.

### 5-FLOW OF EVENTS:

- Admin creates toplist function and runs.
- Admin list musics the most listened musics
- Admin selects the music listed according to certain criteria.
- Admin puts selected musics to toplist
- Admin adds toplist function in system.



### 6-EXCEPTIONS:

- While admin creates toplist function,it put to off trends musics.

### 1-USE CASE NAME: PLAYLIST

### 2-PARTICIPATING ACTOR: Costumer

### 3-ENTRY CONDITION:

- Costomer runs to playlist function
- Costomer has to account

### 4-EXIT CONDITION:

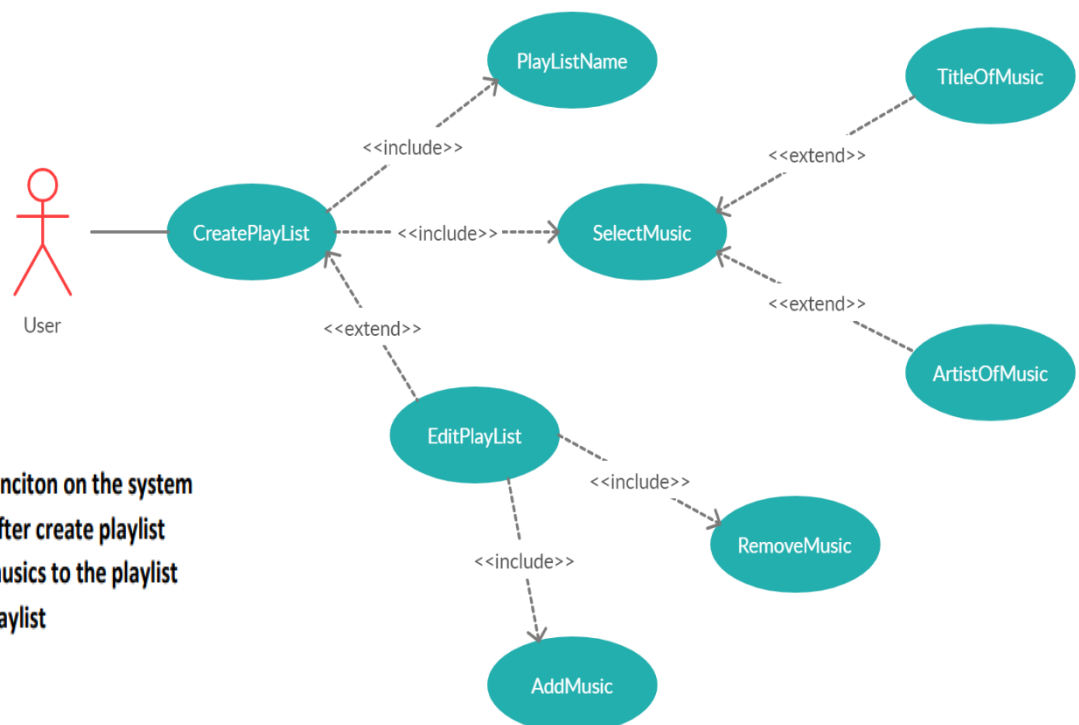
- Customer creates playlist.

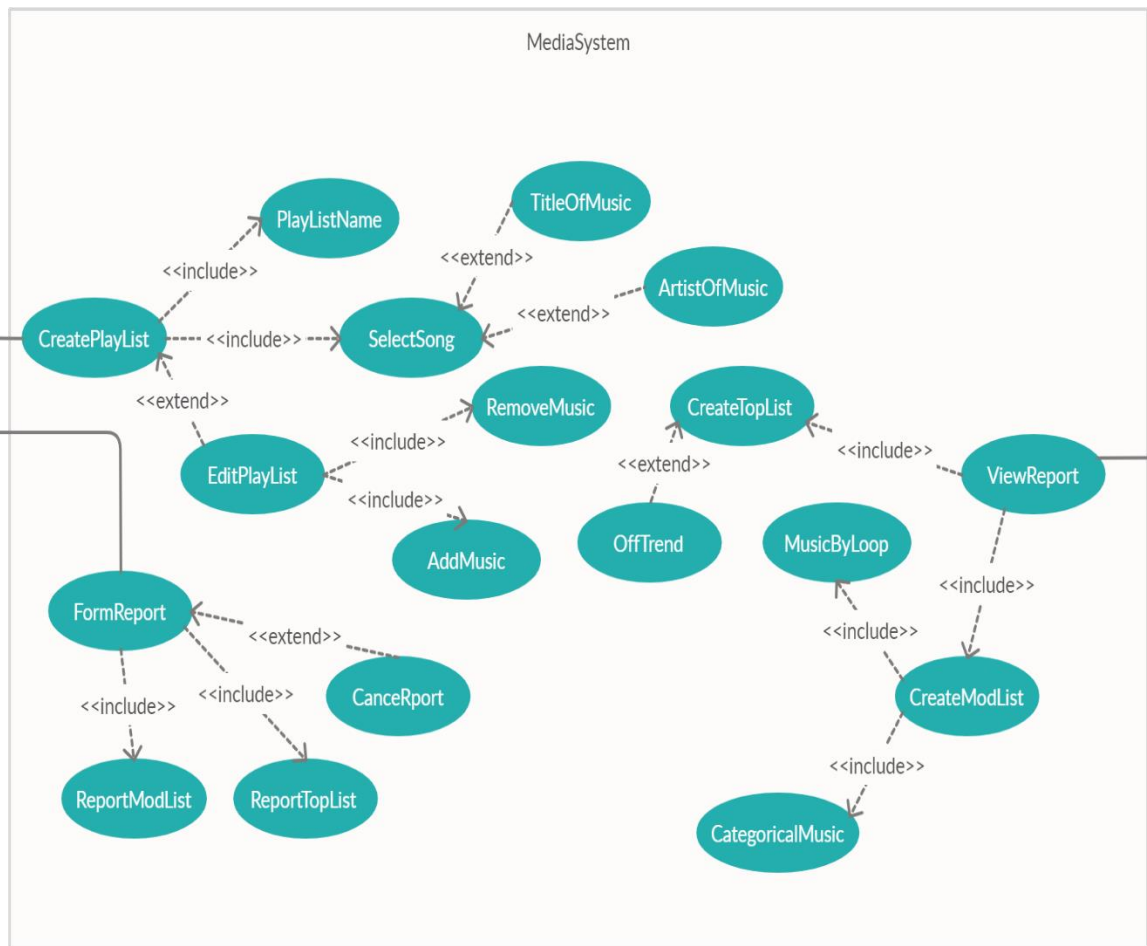
### 5-FLOW OF EVENTS:

- Customer Finds the Playlist functon on the system
- Customer use to Select song after create playlist
- Costomer adds their chosen musics to the playlist
- Costomer start to listen the playlist

### 6-EXCEPTIONS:

- If customer meet some problems,customer use to edits.
- While customer select music,It can be select according to title or Artist







### **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**

<Mock-ups illustrating the user interface of the system and navigational paths representing the sequence of screens.>

Step 4 activity

## **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