**SOFTWARE ENGINEERING PROBLEM SPECIFICATION TABLE**

|  |  |
| --- | --- |
| CLIENT | NeoTunes Company |
| USER | Neotunes administrators and customers |
| FUNCTIONAL REQUIREMENTS | 1. Register producer users. 2. Register consumer users. 3. Register songs 4. Register podcasts. 5. Create a playlist. 6. Add audio to playlist 7. Delete audio from a playlist 8. Share a playlist. 9. Simulate playback of a song or podcast 10. Buy a song. 11. Report the total accumulated number of reproductions on the entire platform. 12. Report the most listened song genre 13. Report the most listened podcast category for a specific user and for the entire platform. 14. Top 5 artists and content creators with the highest number of reproductions 15. Top 10 songs and podcasts with the highest number of plays. 16. Report the number of songs sold and the total sales value ($) by song genre. 17. Report the total number of sales and the total sales value ($) of the best-selling song on the platform. |
| CONTEXT OF THE PROBLEM | NeoTunes, a Denmark-based company, needs to develop a software prototype that will enable it to compete in the streaming music and audio content market. The company's business model has two focuses, user subscriptions and the sale of songs through the platform. Unlike other competitors in the market, NeoTunes allows users to truly own their music catalog. |
| NON-FUNCTIONAL REQUIREMENTS |  |

**Functional Requirements Analysis Table (Note: One table for each functional requirement)**

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | RF1. Register producer users. | | |
| Summary | The system must allow the registration of users who are producers, artists and content creators. | | |
| Inputs | Name of entry | Data type | Selection or repetition condition |
| type | int |  |
| nickname | String |  |
| id | String |  |
| date | String |  |
| name | String |  |
| urlPhoto | String |  |
| General activities necessary to obtain the results | 1. Data reading.  2. An object of type Artist or ContentCreator is instantiated, according to the user's decision, with the data given by the user and the default values of number of reproductions and time of reproduction (these attributes are initialized in 0).  3. This object is stored in the arrayList of type User in the controller. | | |
| Result or postcondition | Register the user in the arrayList of controller users. | | |
| Outputs | Name of entry | Data type | Selection or repetition condition |
| Registration report | boolean |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | RF2. Register consumer users. | | |
| Summary | The system must allow the registration of consumer, standard and premium users. | | |
| Inputs | Name of entry | Data type | Selection or repetition condition |
| type | int |  |
| nickname | String |  |
| id | String |  |
| date | String |  |
| General activities necessary to obtain the results | 1. Data reading.  2. A Standard or Premium type object is instantiated, according to the user's selection, with the data given by the user.  3. This object is stored in the arrayList of type User in the controller. | | |
| Result or postcondition | Register the user in the arrayList of controller users. | | |
| Outputs | Name of entry | Data type | Selection or repetition condition |
| Registration report | boolean |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | RF3. Register songs | | |
| Summary | The system must allow the registration of songs, which are the audios produced by the artists. | | |
| Inputs | Name of entry | Data type | Selection or repetition condition |
| idArtist | int |  |
| name | String |  |
| url | String |  |
| duration | String |  |
| price | double |  |
| album | String |  |
| genreIndicative | int |  |
| General activities necessary to obtain the results | 1. Data reading.  2. A Song type object is instantiated with the data given by the user.  3. This object is stored in the arrayList of type Audio in the controller and in the selected artist. | | |
| Result or postcondition | Record the song in the controller's audio arrayList. | | |
| Outputs | Name of entry | Data type | Selection or repetition condition |
| Registration report | boolean |  |
|  |  |  |
|  |  |  |
|  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | RF4. Register podcasts | | |
| Summary | The system must allow the registration of podcasts, which are audios produced by content creators. | | |
| Inputs | Name of entry | Data type | Selection or repetition condition |
| idCCreator | int |  |
| name | String |  |
| url | String |  |
| duration | String |  |
| description | String |  |
| categoryIndicative | int |  |
| General activities necessary to obtain the results | 1. Data reading.  2. A Podcast type object is instantiated with the data given by the user.  3. This object is stored in the arrayList of type Audio in the controller and in the selected content creator. | | |
| Result or postcondition | Registration of the podcast in the controller's audio arrayList. | | |
| Outputs | Name of entry | Data type | Selection or repetition condition |
| Registration report | boolean |  |
|  |  |  |
|  |  |  |
|  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | RF5. Create a playlist. | | |
| Summary | The system must allow the creation of a playlist for consumer users. The standard user has a limit of 20 playlists, the premium user can create unlimited playlists. | | |
| Inputs | Name of entry | Data type | Selection or repetition condition |
| idUserConsumer | int |  |
| Name | String |  |
| General activities necessary to obtain the results | 1. Users of the Consumer class, whether Standard or Premium, are displayed on the screen. 2. Data reading. 3. A Playlist type object is instantiated and sent as a parameter to the addPlaylist method of the UserConsumer type object, either Standard or Premium. 4. The playlist is added to the Playlist type object containing structure of the UserConsumer object, either of the Standard or Premium class. | | |
| Result or postcondition | Creation of the playlist and association of the object to the selected user | | |
| Outputs | Name of entry | Data type | Selection or repetition condition |
| Creation report | boolean |  |
|  |  |  |
|  |  |  |
|  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | RF6. Add audio to playlist | | |
| Summary | The system must allow the consuming user to add an audio to a playlist. | | |
| Inputs | Name of entry | Data type | Selection or repetition condition |
| idUser | int |  |
| idPlaylist | int |  |
| idAudio | int |  |
| General activities necessary to obtain the results | 1. The following users are displayed on the screen 2. The user is selected from one of the child classes of UserConsumer 3. The playlists of the selected user are displayed 4. Select the playlist to be modified 5. The audios of the platform are shown 6. The audio to be added to the playlist is selected 7. The selected audio is added to the selected playlist. | | |
| Result or postcondition | The selected audio is added to the selected playlist. | | |
| Outputs | Name of entry | Data type | Selection or repetition condition |
| Process report | boolean |  |
|  |  |  |
|  |  |  |
|  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | RF7. Removing audio from a playlist | | |
| Summary | The system shall allow the consuming user to remove an audio from a playlist. | | |
| Inputs | Name of entry | Data type | Selection or repetition condition |
| idUser | int |  |
| idPlaylist | int |  |
| idAudio | int |  |
|  |  |  |
|  |  |  |
| General activities necessary to obtain the results | 1. The following users are displayed on the screen 2. The user is selected from one of the child classes of UserConsumer 3. The playlists of the selected user are displayed 4. Select the playlist to be modified 5. The audios of the selected playlist are displayed. 6. Select the audio to be deleted from the playlist. 7. The selected audio is removed from the selected playlist. | | |
| Result or postcondition | Removing an Audio object from the Playlist class object selected by the user | | |
| Outputs | Name of entry | Data type | Selection or repetition condition |
| Elimination report | boolean |  |
|  |  |  |
|  |  |  |
|  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | RF8. Share a playlist. | | |
| Summary | The system must allow a playlist to be shared through a 16-digit code, and depending on the content of the playlist the code must appear in a 6x6 matrix in a certain pattern. | | |
| Inputs | Name of entry | Data type | Selection or repetition condition |
| idUser | int |  |
| idPlaylist | int |  |
|  |  |  |
|  |  |  |
|  |  |  |
| General activities necessary to obtain the results | 1. The following users are displayed on the screen 2. The user is selected from one of the child classes of UserConsumer 3. The playlists of the selected user are displayed 4. Select the playlist to share | | |
| Result or postcondition | The numerical code that identifies the playlist is shown on the screen with its respective 6x6 matrix containing this code. | | |
| Outputs | Name of entry | Data type | Selection or repetition condition |
| playlistCode | String |  |
| Error alert | String | In case of an error such as missing User or Playlist |
|  |  |  |
|  |  |  |
|  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | RF9. Simulate playback of a song or podcast | | |
| Summary | The system must be able to simulate the reproduction of an audio, either song or podcast. It should be noted that the Standard users must apply the advertisement system. | | |
| Inputs | Name of entry | Data type | Selection or repetition condition |
| idAudio | int |  |
| idUser | int |  |
| General activities necessary to obtain the results | Verify that there is at least one UserConsumer and one Audio object in the program, otherwise terminate the process.  2. The consuming users are displayed so that the entity that will play an audio is selected.  3. Display on the screen the Audios available for listening, so that the user can select the one he/she wants to play.  4. If the audio played is a song and the user is a Standard user, verify if the user has listened to 2 songs in a row, if so, display a randomly chosen ad and set the number of songs played by the selected user to 0. If the audio is a podcast, always play an ad.  5. Verify if the audio was previously listened by the user, if not, create a clone of the listened audio with number of reproductions in 1 in the playlist of reproducedAudios of the selected user. If the audio had been listened, increase the reproductions of the clone of the selected audio of the initial user.  6. Increase the time played and the number of reproductions of the producer to whom the listened audio corresponds. | | |
| Result or postcondition | Audio playback is simulated, taking into account the reports to be incremented and the advertisements to be displayed. | | |
| Outputs | Name of entry | Data type | Selection or repetition condition |
| reproduceAudio | String |  |
| Error alert | String | In the event that there are no users or audios in the system |
| Advertisement | String | In the event that a standard user's song plays equal 2 |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | RF10. Buy a song | | |
| Summary | The system must allow a consumed user to buy a song | | |
| Inputs | Name of entry | Data type | Selection or repetition condition |
| idSong | int |  |
| idUser | int |  |
| General activities necessary to obtain the results | Verify that there is at least one UserConsumer and one Audio object in the program, otherwise terminate the process.  2. The consumer users are displayed so that the entity that will purchase an audio is selected.  3. Display on screen the Audios available for listening, so that the user can select the one he/she wishes to purchase.  4. Associate the selected audio to the list of the user who manages the purchases with its respective purchase date. In addition, the sales number attribute of the purchased song must be increased. | | |
| Result or postcondition | A song is associated with a user's shopping list. | | |
| Outputs | Name of entry | Data type | Selection or repetition condition |
| Purchase report | boolean |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | RF11. Report the total accumulated number of reproductions on the entire platform. | | |
| Summary | The system must allow reporting the total accumulated number of reproductions on the entire platform. | | |
| Inputs | Name of entry | Data type | Selection or repetition condition |
|  |  |  |
| General activities necessary to obtain the results | 1. Traverse the structure containing Audio class objects. 2. Obtain the number of reproductions of each Audio 3. Keeping the sum of the number of reproductions of each Audio | | |
| Result or postcondition | The total accumulated number of reproductions on the entire platform is displayed on the screen. | | |
| Outputs | Name of entry | Data type | Selection or repetition condition |
| totalReproductions | int |  |
| Error alert | String | In case there are no audios |
|  |  |  |
|  |  |  |
|  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | RF12. Report the most listened song genre | | |
| Summary | The system must report the most listened song genre for a specific user and for the entire platform and its number of plays. | | |
| Inputs | Name of entry | Data type | Selection or repetition condition |
| idUser | int | In case you want to know the most listened genre of a specific user. |
| General activities necessary to obtain the results | Activities for the platform   1. Traverse the structure containing Audio class objects. 2. Obtain the number of reproductions and genre of each Audio. 3. Keeping the sum of the number of reproductions of each Audio   A user's favorite genre activities.   1. Check that there is at least one userConsumer, otherwise terminate the process. 2. The consumer users are displayed so that you can select the user whose favorite genre of songs you want to know. 3. Verify that the user has played at least one song, if this is not the case, report this event. 4. If the user has listened to at least one song, from the user's list of played audios count how many songs he/she has listened to by genre taking into account its numReproductions attribute. 5. Report the genre of the songs with the highest number of reproductions. | | |
| Result or postcondition | It identifies and reports the genre most listened to by a specific user and by the entire platform. | | |
| Outputs | Name of entry | Data type | Selection or repetition condition |
| Error alert | String | In case there are no songs or the user selected |
| favoriteUserGenre | String | In case a user's favorite genre has been requested. |
| favoriteGlobalGenre | String | In case the favorite genre of the program has been requested. |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | RF13. Report the most listened podcast category for a specific user and for the entire platform. | | |
| Summary | The system should allow reporting the most listened podcast category for a specific user and for the whole platform by showing the name and number of plays. | | |
| Inputs | Name of entry | Name of entry | Name of entry |
| idUser | idUser | idUser |
| General activities necessary to obtain the results | Process a user's favorite category.   1. Check that there is at least one userConsumer, otherwise terminate the process. 2. The consumer users are displayed so that you can select the user whose favorite category of podcasts you want to know. 3. Verify that the user has played at least one podcast, if this is not the case, report this event. 4. If the user has listened to at least one podcast, from the user's list of played audios, count how many podcasts the user has listened to by category, taking into account its numReproductions attribute. 5. Report the category of podcasts with the highest number of plays.   Favorite category process at the program level.   1. Count how many songs have been listened to by genre taking into account its numReproductions attribute. 2. Report the category of podcasts with the highest number of plays. | | |
| Result or postcondition | The most listened category by a specific user is identified and reported, and the most listened category at the program level is identified and reported. | | |
| Outputs | Name of entry | Name of entry | Name of entry |
| Error alert | Error alert | Error alert |
| favoriteUserCategory | favoriteUserCategory | favoriteUserCategory |
| favoriteGlobalCategory | favoriteGlobalCategory | favoriteGlobalCategory |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | RF14. Top 5 artists and content creators with the highest number of plays. | | |
| Summary | The system must inform the members of the Top 5 artists and Top 5 content creators on the platform, of which the name and number of total reproductions are reported. | | |
| Inputs | Name of entry | Data type | Selection or repetition condition |
| General activities necessary to obtain the results | 1. Depending on the type of producing user to which you want to identify the top 5, verify that there are registered objects of the corresponding nature; if there are none, terminate the execution. 2. Create a temporary array containing the objects of the required producer user type, either artist or content creator. 3. Having filtered the array, organize it from highest to lowest according to the number of reproductions of each user. 4. If there are less than 5 identified users, report all existing users. If the number of users exceeds 5, report only the top 5. | | |
| Result or postcondition | The top 5 producing users with the highest number of reproductions, either for artists or content creators, are identified and reported. In case there are less than 5 users identified, the size of the top is adjusted to the number of objects. | | |
| Outputs | Name of entry | Data type | Selection or repetition condition |
| reportTop5 | String |  |
| Error alert | String | In case there are no registered producer users |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | RF15. Top 10 songs and podcasts with the highest number of plays. | | |
| Summary | The system must inform each one of the members of the Top 10 songs and Top 10 podcasts, of which the name, genre or category and total number of reproductions must be informed. | | |
| Inputs | Name of entry | Data type | Selection or repetition condition |
| General activities necessary to obtain the results | Depending on the type of audio for which you want to identify the top 10, verify that there are objects of the corresponding nature registered, if there are not, terminate the execution.  2. Create a temporary array containing objects of the required audio type, either a song or podcast.  3. Having filtered the array, organize it from highest to lowest according to the number of reproductions of each audio.  In case there are less than 5 audios identified, report all the existing ones. If the audios exceed 5 units, report only the top 5. | | |
| Result or postcondition |  | | |
| Outputs | Name of entry | Data type | Selection or repetition condition |
| reportTop10 | String |  |
| Error alert | String | In case there are no audios recorded |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | RF16. Report the number of songs sold and total sales value ($) by song genre. | | |
| Summary | The program must report the number of songs sold and the total sales value ($) for each song genre. | | |
| Inputs | Name of entry | Data type | Selection or repetition condition |
| General activities necessary to obtain the results | 1. Verify that at least one song has been registered, if not, notify and terminate the process.  2. Count for each genre the songs sold associated with that genre. In addition, count and accumulate the revenue generated by each song of each type of genre.  3. Report the report of revenues and songs sold by genre. | | |
| Result or postcondition | Revenues and songs sold by genre are reported. | | |
| Outputs | Name of entry | Data type | Selection or repetition condition |
| report | String |  |
| Error alert | String | In case there are no registered songs |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | RF17. Report the total number of sales and the total sales value ($) of the top selling song on the platform. | | |
| Summary | The system must report the total number of sales and the total sales value ($) of the best-selling song on the platform. | | |
| Inputs | Name of entry | Data type | Selection or repetition condition |
| General activities necessary to obtain the results | 1. Verify the existence of registered songs.  In case there are registered songs, identify the song with the highest number of sales by comparing the numSales attribute of each song.  3. Return the information and revenue generated from the best-selling song. | | |
| Result or postcondition | The total number of sales and the total sales value ($) of the best-selling song on the platform are reported. | | |
| Outputs | Name of entry | Data type | Selection or repetition condition |
| bestSellingSong | String |  |
| Error alert | String | In case there are no registered songs |
|  |  |  |
|  |  |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **Requerimiento Funcional** | **Nombre de la Clase** | **Nombre del método** |
| Register producer users. | Class Manager | menu():void  registerUser():void |
| Class Controller | registerUser(name : String, id : String, typeUser : int):boolean |
| Class Class User | User(name : String, id : String, date : Date) |
| Register consumer users. | Class Manager | menu():void  registerUser():void |
| Class Controller | registerUser(name : String, id : String, urlPhoto : String, typeUser : int):boolean |
| Class User | User(name : String, id : String, date : Date) |
| Register song | Class Manager | menu():void  registerAudio():void |
| Class Controller | registerAudio(userArtist : int, nameSong : String, album : String, posGenre : int, url : String, duration : double, price : double):boolean |
| Class  Audio | Audio(name : String, url : String, duration : double, numReproduction : int) |
| Class User | registerSong(song :Song) : boolean |
| Register podcast | Class Manager | menu():void  registerAudio():void |
| Class Controller | registerAudio(userArtist : int, nameSong : String, album : String, posCategory : int, url : String, duration : double):boolean |
| Class  Audio | Audio(name : String, url : String, duration : double, numReproduction : int) |
| Class User | registerPodcast(podcast :Podcast) : boolean |
| Create a playlist. | Class Manager | menu():void  createPlaylist():void |
| Class Controller | createPlaylist(idUserConsumer : int, name : String) : boolean |
| Class Playlist | Playlist(name : String, code : String) |
| Class User | createPlaylist(playlist : Playlist) : boolean |
| Add audio to playlist | Class Manager | menu():void  editPlaylist():void |
| Class Controller | addAudioToPlaylist(idUserConsumer : int, idPlaylist : int, idAudio : int) : boolean |
| Class Playlist | addAudio(audio : Audio) : boolean |
| Class User | addAudio(idPlaylist : int, audio :Audio) : boolean |
| Delete audio from a playlist | Class Manager | menu():void  editPlaylist():void |
| Class Controller | addAudioToPlaylist(idUserConsumer : int, idPlaylist : int, idAudio : int) : boolean |
| Class Playlist | deleteAudioToPlaylist(idUserConsumer : int, idPlaylist : int, idAudioPlaylist : int) : |
| Class User | deletaAudio(idPlaylist : int, idAudio : int) : boolean |
| Share a playlist. | Class Manager | menu():void  sharePlaylist():void |
| Class Controller | sharePlaylist(idUserConsumer : int, idPlaylist : int) : String |
| Simulate playback of a song or podcast | Class Manager | menu():void  simulateReproduced() : void |
| Class Controller | simulateReproduced(idConsumer : int, idAudio : int) : String |
| Class Playlist | deleteAudioToPlaylist(idUserConsumer : int, idPlaylist : int, idAudioPlaylist : int) : |
| Class User | reproduceAudio(audio :Audio) : String |
| Buy a song. | Class Manager | menu():void  buySong () : void |
| Class Controller | buyAudio(idUser : int, idAudio : int) : boolean |
| Class User | buyAudio(sellable :Sellable) : boolean |
| Class purchase | Purchase(purchase : model.Sellable, dateOfPurchase : Date) |
| Class Audio | sell():void |
| Report the total accumulated number of reproductions on the entire platform. | Class Manager | Menu():void  getStatistics() : void |
| Class Controller | songReproductions() : int  podcastReproductions() : int |
| Report the most listened song genre | Class Manager | Menu():void  getStatistics() : void |
| Class Controller | favoriteUserSongGenre () : int |
| Report the most listened song genre for a specific user and for the entire platform | Class Manager | Menu():void  getStatistics() : void |
| Class Controller | favoriteUserSongGenre () : int  favoriteGlobalSongGenre() : String |
| Report the most listened podcast category for a specific user and for the entire platform | Class Manager | Menu():void  getStatistics() : void |
| Class Controller | favoriteUserPodcastCategory() : favoriteGlobalPodcastCategory() : String |
| Top 5 artists and content creators with the highest number of reproductions | Class Manager | Menu():void  getStatistics() : void |
| Class Controller | top5ContentCreator(): String  top5Artist (): String |
| Top 10 songs and podcasts with the highest number of plays. | Class Manager | Menu():void  getStatistics() : void |
| Class Controller | top10Songs() : String  top10Podcast() : String |
| 1. Report the number of songs sold and the total sales value ($) by song genre. | Class Manager | Menu():void  getStatistics() : void |
| Class Controller | incomesByGenre () : String |