

### INTEGRATOR 3

Nicolás Cuéllar Molina – A00394970

Davide Flamini Cazarán - A00381665

#### PROBLEM SPECIFICATION TABLE

CLIENT	NeoTunes
USER	Administrator
FUNCTIONAL REQUIREMENTS	<ul style="list-style-type: none"><li>- R1: Register producer user</li><li>- R2: Register consumer user</li><li>- R3: Register song</li><li>- R4: Register Podcast</li><li>- R5: Create a Playlist</li><li>- R6: Edit a Playlist</li><li>- R7: Share a playlist.</li><li>- A8: Simulate playing a song or podcast (standard and premium).</li><li>- R9: Buy a song.</li><li>- R10: Generate reports with the recorded data</li></ul>
CONTEXT OF THE PROBLEM	Develop a program that performs functions similar to music content applications, such as Apple Music or Spotify.
NON-FUNCTIONAL REQUIREMENTS	<ul style="list-style-type: none"><li>- RN1: It will be done in 2 deliveries, and their dates are stipulated by the client</li><li>- RN2: The code will be done in JAVA language.</li><li>- RN3: The documentation must be in the forms stipulated by the client.</li></ul>

## Traceability

Functional Requirement	Class	Name Method Name
R1: Register producer user	Manager	registerUser()
	Controller	searchUser()
	Manager	registerProducer()
	Manager	selectOptionTrueOrFalse()
	Controller	registerCreator()
	Creator	Creator()
	Controller	registerArtist()
	Artist	Artist()
R2: Register consumer user	Manager	registerUser()
	Controller	searchUser()
	Manager	registerConsumer()
	Manager	selectOptionTrueOrFalse()
	Controller	registerStandard()
	Standard	Standard()
	Controller	registerPremium()
	Premium	Premium()
R3: Register Song	Manager	registerAudio()
	Manager	selectOptionTrueOrFalse()
	Controller	searchAudio()
	Manager	registerSong()
	Controller	createSong()
	Song	Song()
R4: Register Podcast	Manager	registerAudio()

	Manager	selectOptionTrueOrFalse()
	Controller	searchAudio()
	Manager	registerPodcast()
	Controller	createPodcast()
	Podcast	Podcast()
R5: Create Playlist	Manager	createPlaylist( )
	Controller	showNickName()
	Controller	createPlaylist()
	Playlist	Playlist()
	Standard	addPlaylist()
	Premium	addPlaylist()
R6: Edit a Playlist	Manager	editPlaylist()
	Controller	showNickName()
	Controller	showNamePlaylist()
	Manager	editMenuPlaylist()
	Manager	selectPlaylist()
	Manager	addSong()
	Controller	showNameAudio()
	Manager	selectAudio( )
	Controller	addAudioToPlaylist()
	Playlist	addAudio()
	Manager	deleteSong()
	Controller	showAudiosToPlaylist()
	Controller	deleteAudioToPlaylist()
	Playlist	deleteAudio()
	Manager	renamePlaylist()
	Controller	setNamePlaylist()

	Playlist	setName()
R7: Share a Playlist	Controller	sharePlaylist()
	Controller	showUserPlaylist ()
	Controller	searchUser()
	Consumer	searchPlaylist()
	Playlist	showMatrix()
	Playlist	createPlaylistCode()
R8: Simulate playing a song or podcast (standard and premium)	Controller	simulateAudioPlayback()
	Controller	searchUser()
	Consumer	searchPlaylist()
	Playlist	searchAudio()
	Song	play()
	Podcast	play()
	Controller	addAmountPlayback()
R9: Buy a song	Controller	buySong()
	Controller	showAudioName()
	Controller	searchAudio()
	Consumer	addSong()
	Song	setSaleDate()
	Controller	addSales()

R10: Generate report with recorded data	Controller	generateInform()
	Controller	searchUser()
	Controller	serchSong()

## Functional requirements analysis table

Name or identifier	R1: Register producer user		
Summary	This method creates a new producer user within the system		
Inputs	Input name	Data type	Selection or repetition
	condition nickName	String	That the nickName already exists
	id	String	
	name	String	
	url	String	
	artistOrCreator	int	
General activities necessary to obtain the results	<ol style="list-style-type: none"> <li>1. Choose if it is Artist or Creator (1-Artist : 2-Creator)</li> <li>2. Request the data</li> <li>3. Create the Artist or Creator</li> <li>4. Add it to the list of users</li> </ol>		
Result or postcondition	A new registered producer user		
Outputs	Input name	Data type	Selection or repetition condition
	Confirmation	String	That all data has been registered correctly

Name or identifier	R2: Register User Consumer		
Summary	This registered a new User consumer(standard or Premium)		
Inputs	Input name	Data type	Selection or repetition
	condition nickName	String	That the nickName already exists
	id	String	
	name	String	
	url	String	
	standardOrPremium	int	(1-Standard : 2-Premium m)
General activities necessary to obtain the results	<ol style="list-style-type: none"> <li>1. Choose if it is Standard or Premium (1-Standard : 2-Premium)</li> <li>2. Request the data</li> <li>3. Create the Standard or Premium</li> <li>4. Add it to the list of users</li> <li>5. Show the message of the process</li> </ol>		
Result or postcondition	A new consumer user registered		
Outputs	Input name	Data type	Selection or repetition condition
	Confirmation	String	That all data has been correctly registered

Name or identifier	R3: Register song		
Summary	This request will register a song to an artist		
Inputs	Input name	Data type	Selection or repetition
	condition numberUser	int	
	name	String	
	url	String	
	duration	int	
	album	String	
	saleValue	double	
	typeGenre	int	
General activities necessary to obtain the results	<ol style="list-style-type: none"> <li>1. Check that artist users are successful</li> <li>2. Select the artist that will create the song</li> <li>3. Request the song data</li> <li>4. Show the process message</li> </ol>		
Result or postcondition	A new registered song		
Outputs	Input name	Type of data	Selection or repetition condition
	Confirmation	String	That all data has been registered correctly

Name or identifier	R4: Register Podcast		
Summary	Registers a podcast to a Creator		
Entries	Entry name	Data type	Selection or repetition
	condition numberUser	int	
	name	String	
	url	String	
	duration	int	
	description	String	
	type Category	int	
General activities necessary to obtain the results	<ol style="list-style-type: none"> <li>1. Check that there are registered</li> <li>2. Creators Select the creator that will create the Podcast</li> <li>3. Request the Podcast data</li> <li>4. Show the process message</li> </ol>		
Result or postcondition	a new registered podcast		
Outputs	Input name	Data type	Selection condition or repetition
	Confirmation	String	That all the data have been registered correctly



Name or identifier	R5: Create a Playlist		
Summary	This request registers a new playlist to a consumer user		
Inputs	Input name	Data type	Selection or repetition
	condition numberUser	int	
	nam e	String	
General activities necessary to obtain the results	<ol style="list-style-type: none"> <li>1. Select the User who is going to create a Playlist</li> <li>2. If he is a standard user verify that he has not exceeded the limit of 20 playlists</li> <li>3. Request the Playlist data</li> <li>4. Show the message of the process</li> </ol>		
Result or postcondition	A new registered playlist to a selected user		
Outputs	Input name	Data type	Selection or repetition condition
	Confirmation	String	That all the data has been registered correctly

Name or identifier	R6: Edit a Playlist		
Summary	This requirement consists of allowing the editing of a consumer user's playlist. The user enters the option, which allows them to select what action they want to perform with the Playlist.		
Inputs	Input name	Data type	Selection or repetition
	condition		
	numberUser	int	
	numberPlaylist	int	
	numberSong	int	
	newName	string	
General activities necessary to obtain the results	<ol style="list-style-type: none"> <li>1. Select the user who has the Playlist</li> <li>2. Select the Playlist you want to edit</li> <li>3. Select the menu option (add - delete - rename )</li> <li>4. Select the Song to edit (if you choose the add - delete option)</li> <li>5. Enter the new Playlist name (if you choose the rename option)</li> <li>6. Display the process message</li> </ol>		
Result or postcondition	An edited playlist, depending on the option chosen by the user		
Outputs	Input name	Data type	Selection or repetition condition
	Message	String	That all data has been registered correctly

Name or identifier	R7: Sharing a Playlist		
Summary	<p>A playlist must be able to be shared. To share it, the system must generate a code for the playlist, through specific routes to a matrix depending on the content of the playlist:</p> <p>A playlist with only Songs: Tour in letter N  A playlist with only Podcasts: Tour in letter T  A playlist with Songs and Podcasts: the number of cells <math>i,j</math>, traversing the matrix from bottom to top and from right to left, when the sum <math>i+j</math> is an odd number greater than 1.</p>		
Entries	Entry name	Data type	Selection or repetition
	condition	int	
	numberUser	int	
	playlistPos	int	
	numberSong	int	
General activities needed to get the results	newName	string	
	<ol style="list-style-type: none"> <li>1. Select the user who has the playlist</li> <li>2. Select the playlist to share</li> <li>3. Show the playlist matrix</li> <li>4. Show the code of Playlist</li> </ol>		
Result or Postcondition	One Playlist		
Outputs	Input Name	Data Type	Selection or Repeat Condition ón
	Message	String	If everything goes well, the approval message appears.

Name or identifier	R8: Simulate the reproduction of a song or podcast (standard and premium).		
Summary	This requirement simulates playing a song or podcast.		
Inputs	Input name	Data type	Selection or repetition
	condition		
	numberUser	int	
	playlistPos	int	
	numberSong	int	
General activities needed to get the results	<ol style="list-style-type: none"> <li>1. Select the user that has the playlist</li> <li>2. Select the playlist that has the audio to play</li> <li>3. Select the name of the Audio to play</li> <li>4. Add play and play time to NeoTunes and user</li> <li>5. audio Play audio or advertisement before (if applicable)</li> </ol>		
Result or postcondition	System plays audio chosen by user		
Outputs	Input name	Data type	Selection or repeat condition
	Message	String	If all goes well the approval message appears.

Name or identifier	R9: Buy a song		
Summary	The system must allow the consumer user to be able to buy a NeoTunes song		
Inputs Input	name	Data type	Selection or repetition
	condition numberUser	int	
	numberSong	int	
General activities necessary to obtain the results	<ol style="list-style-type: none"> <li>1. Select the user who will buy the song</li> <li>2. Select the song to buy</li> <li>3. Verify that there is space available in the ArrayList of consumer songs</li> <li>4. Add the song to the ArrayList of user songs</li> <li>5. Add a new sale to the song, increase the variable by 1 unit</li> </ol>		
Result or postcondition	Song purchased by the consumer ( the song enters the ArrayList of the consumer)		
Outputs	Input name	Data type	Selection or repetition condition
	Message	String	If everything goes well, the approval message appears.

Name or identifier	R10: Generate reports with the registered data		
Summary	<p>With this requirement it is possible to generate reports through the registered data in the system. There will be a variable called option that will allow you to choose the required report.</p> <p>The available options are:</p> <ul style="list-style-type: none"> <li>- For each type of audio, songs and podcast, inform the total accumulated reproductions in the entire platform.</li> <li>- Report the most listened song genre (name and number of plays) for a specific user and for the entire platform and its number of plays.</li> <li>- Report the most listened to podcast category (name and number of views) for a specific user and for the entire platform.</li> <li>- For each of the members of the Top 5 artists and Top 5 content creators on the platform, report the name and number of total reproductions.</li> <li>- For each of the members of the Top 10 songs and Top 10 podcasts, report the name, genre or category, and total number of reproductions.</li> <li>- For each genre, report the number of songs sold and the total value of sales (\$).</li> <li>- Of the best-selling song on the platform, report the total number of sales and the total sale value (\$).</li> </ul>		
Inputs	Input name	Data type	Selection or repetition condition
	option	int	
General activities necessary to obtain the results	<ol style="list-style-type: none"> <li>1. Read the option that the user will choose for the report</li> <li>2. Generate reports</li> <li>3. Show the generated report on the screen</li> </ol>		

Result or postcondition	The generated report is displayed.		
Outputs	Input name	Data type	Selection or repetition condition
	Message	String	That all data has been entered correctly.