TRACEABILITY TABLE

NeoTunes - Integrative Task 3

Functional Requirement	Class Name	Method name
Rf001 - Register Producer	Manager	registerProducer(): void
	Controller	registerArtist(name: String, documentId: String, pictureUrl: String): boolean
		registerContentCreator(name: String, pictureUrl: String): boolean
		searchUser(userName: String): User
	ContentCreator	ContentCreator(name: String, pictureUrl: String)
	Artist	Artist(name: String, documentId: String, pictureUrl: String)
RF002 - Register Consumer	Manager	registerConsumer():boolean
	Controller	registerStandardConsumer(): boolean
		registerPremiumConsumer(): boolean
		searchUser(userName: String): User
	StandardConsumer	StandardConsumer(nickname: String, documentId: String)

	PremiumConsumer	PremiumConsumer(name: String, pictureUrl: String)
RF003 - Register Audio	Manager	registerAudio(): void
	Controller	showUsers(): String
		showPodcastCategories(): String
		showSongGenres(): String
		showCatalogue(): String
		searchUser(userName: String): User
		searchAudio(audioName: String): Audio
		registerSong(songName: String, pictureUrl: String, duration: String, genre: int, album: String, saleValue: double, artistName: String): boolean
		registerPodcast(podcastName: String, pictureUrl: String, duration: String, category: String, description: String, contentCreatorName: String): boolean
	Artist	addAudio(newAudio: Audio): boolean
	ContentCreator	addAudio(newAudio: Audio): boolean

	Song	Song(songName: String, pictureUrl: String, duration: String, genre: int, album: String, saleValue: double, artistName: String)
	Podcast	Podcast(podcastName: String, pictureUrl: String, duration: String, category: String, description: String, contentCreatorName: String)
RF004 - Register Playlist	Manager	registerPlaylist(): void
	Controller	showUsers(): String
		registerPlaylist(consumerName: String, type: int, name: String): boolean
		searchUser(userName: String): User
		showConsumerPlaylists(consumerName: String): String
	Consumer	addPlaylist(newPlaylist: Playlist): boolean
	Playlist	Playlist(name: String)
	SongPlaylist	SongPlaylist(name:String)
	PodcastPlaylist	PodcastPlaylist(name: String)
RF005 - Edit Playlist	Manager	editPlaylist(): void
	Controller	showUsers(): String

		showConsumerPlaylists(consumerName: String): String
		addOrRemovePlaylistAudio(operation: int, consumerName: String, audioName: String, playlistId: String): boolean
		renamePlaylist(consumerName: String, playlistId: String, playlistId: String): boolean
		searchUser(userName: String): User
		searchPlaylist(id: String): Playlist
		searchAudio(audioName: String): Audio
	Playlist	addAudio(newAudio: Audio): boolean
		searchAudio(newAudio: Audio): Audio
RF006 - Share Playlist	Manager	sharePlaylist(): void
	Controller	showUsers(type: int): String
		showConsumerPlaylists(consumerName: String): String
		sharePlaylist(consumerName: String, playlistId: String): String
		searchUser(consumerName: String): User
		searchPlaylist(id: String): Playlist
	Playlist	updateId(): void

RF007 - Play Audio	Manager	playAudio(): void
	Controller	showCatalogue(type: int): String
		showUsers(type: int): String
		playAudio(consumerName: String, audioName: String): String[]
		searchUser(consumerName: String): User
		searchAudio(audioName: String): Audio
	Consumer	playAudio(audio: Audio): void
		showAdvertise(): boolean
	Record	Record(consumerName: String, audio: Audio)
	Audio	play(): String
RF008 - Buy Song	Manager	buySong(): void
	Controller	showCatalogue(type: int): String
		showUsers(type: int): String
		showPurchases(consumerName: String): String
		buySong(consumerName: String, songName: String): boolean
		searchUser(consumerName: String): User

		searchAudio(audioName: String): Audio
	Consumer	addSong(newSong: Song): boolean
	Record	Record(consumerName: String, audio: Audio)
RF009 - Show Statistics	Manager	showStatistics(): void
	Controller	showTotalPlaysPerType(): String
		calculateTotalPlaysPerType(): int[][]
		mostPlayedSongGenre(): String
		mostPlayedSongGenre(consumerName: String): String
		mostPlayedPodcastCategory(): String
		mostPlayedPodcastCategory(consumerName: String): String
		showUsers(type: int): String
		showUsersTop(limit: int): String
		showAudiosTop(limit: int): String
		showTotalSalesPerGenre(): String
		bestSellerInformation(): String
		sortProducers(): ArrayList<Producer>

		sortAudios(): void
	Producer	calculateTotalPlays(): int
		calculateTotalTimePlayed(): int
	Audio	getNumberOfPlays(): int
	Podcast	getCategory(): Category
	Song	getGenre(): Genre
		getNumberOfSales(): int
		calculateTotalIncome(): double