**Università degli Studi di Salerno**

Corso di Ingegneria del Software

**POORIFY**

Object Design Document

Versione 0.3 / 29.12.2022





|  |  |
| --- | --- |
| Anna Linda Brenga | 0512109669 |
| Alessandro Farina | 0512109741 |
| Christian Mascolo | 0512111139 |
| Jessica Zampetti | 0512109894 |

**INDICE**

# **1. INTRODUCTION**

## **OBJECT DESIGN TRADE-OFFS**

Il principale trade-off individuato in fase di System Design è Space vs Speed, ovvero garantire la fruizione di una grande quantità di contenuti in tempi rapidi. Ovviamente un sistema distribuito scalabile deve garantire all’utenza tempi di risposta ottimali, ma per la nostra applicazione possiamo tollerare dei rallentamenti dovuti allo streaming o al download dei contenuti multimediali dal server remoto.

## **DESIGN PATTERNS**

Elenchiamo di seguito i Design Pattern utilizzati per realizzare l’applicazione. In molti casi si è deciso di non attenersi all’implementazione canonica ma piuttosto di seguire la filosofia dietro il design pattern scelto.

* **COMMAND**

Per permettere all’utente la navigazione delle pagine già visitate tramite *Navigator.java,* una struttura dati basata su due stack che conservano oggetti *Page.java,* wrapper che memorizza tipo e id del contenuto visitato.

* **STRATEGY**

All’interno del metodo *getTracklist(Order order)* della classe *PlaylistBean.java* così da poter restituire diversi ordinamenti degli *AddedBean.java* che compongono la lista di canzoni presenti nella playlist. Il metodo crea un *TreeSet* passando al costruttore un *Comparator* diverso a seconda dell’ordinamento da realizzare in base al parametro *Order*.

* **PROXY**

Per semplificare le query di retrieve delle canzoni all’interno di una playlist il metodo di *PlaylistDAO.java* restituisce degli *AddedBeanProxy.java* che contengono solo l’id dei contenuti desiderati. In secondo momento, nelle Servlet, si provvede a costruire i relativi *AddedBean.java* utilizzando gli altri DAO per recuperare gli oggetti veri e propri partendo dagli id dell’oggetto proxy.

* **ADAPTER**

Per incapsulare in un unico punto la logica di upload dei file su blob container remoto Azure all’interno di *Uploader.java*, classe singleton che si interfaccia con le API del componente off-the-shelf.

* **OBSERVER**

Per notificare l’utente di alcuni eventi, quali l’aggiunta di un brano a una playlist, tramite un pop up Javascript che può essere attivato da qualunque altro script passando come parametro il messaggio di notifica.

## **INTERFACE DOCUMENTATION GUIDELINES**

## **1.2.1 NAMING CONVENTIONS**

Forniamo delle linee guida per la stesura del codice durante la fase di implementazione così da poter essere consistenti e renderne più semplice la lettura sia agli sviluppatori interni al team che a terzi.

* **CLASSES**
  + I nomi delle classi devono rispettare la notazione *UpperCamelCase.*
  + Le classi Bean/DTO devono chiamarsi *NameBean.*
  + Le classi che interagiscono con il layer della persistenza devono chiamarsi *NameDAO*.
  + Le classi che modellano gli oggetti Control, ossia le nostre Servlet, devono essere denominate con un nome che ne esplicita la funzionalità offerta.
* **METHODS**
  + I nomi dei metodi devono rispettare la notazione *lowerCamelCase.*
* **PACKAGE ORGANIZATION**
  + Deve essere creato un package per ogni sottosistema.
  + Le classi che realizzano il sottosistema, principalmente Servlet, DAO e Bean, devono essere contenute nello stesso package.

## **DEFINITIONS, ACRONYMS, AND ABBREVIATIONS**

* OCL: Object Constraint Language
* DAO: Data Access Object
* DTO: Data Transfer Object

## **1.4 REFERENCES**

Nel corso del documento facciamo riferimento a artefatti precedenti quali il **Requirements Analysis Document** (RAD) e al **System Design Document** (SDD).

# **2. PACKAGES**

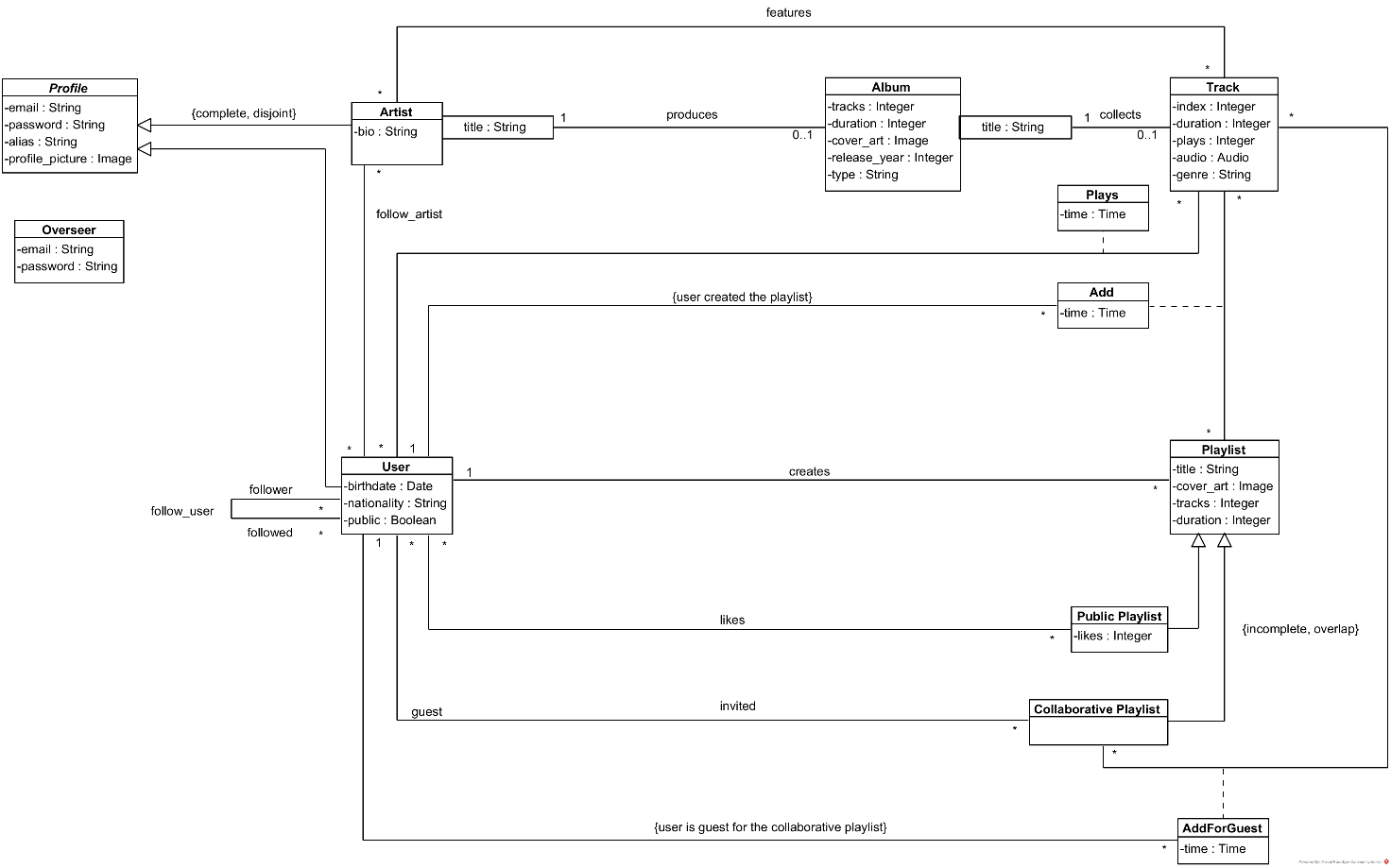
Con riferimento alla divisione in sottosistemi riportata nel SDD elenchiamo le componenti che compongono ciascun package.

* **PROFILE**
  + ProfileBean.java
  + UserBean.java
  + ArtistBean.java
  + OverseerBean.java
  + NationBean.java
  + ProfileDAO.java
  + Login.java
  + Logout.java
  + Signup.java
  + CheckNations.java
  + CheckCredentials.java
  + GetUser.java
  + GetArtist.java
  + FollowArtist.java
  + UnfollowArtist.java
  + FollowUser.java
  + UnfollowUser.java
  + EditProfile.java
  + SearchForGuests.java
* **PLAYLIST**
  + PlaylistBean.java
  + AddedBean.java
  + AddedBeanProxy.java
  + PlaylistDAO.java
  + GetPlaylist.java
  + CreatePlaylist.java
  + DeletePlaylist.java
  + EditPlaylist.java
  + AddTrack.java
  + RemoveTrack.java
  + MakePrivate.java
  + MakePublic.java
  + MakeSingle.java
  + MakeCollaborative.java
  + LikePublicPlaylist.java
  + UnlikePublicPlaylist.java
  + AddGuest.java
  + CheckLike.java
  + ChangeOrder.java
* **ALBUM**
  + AlbumBean.java
  + AlbumDAO.java
  + GetAlbum.java
* **TRACK**
  + TrackBean.java
  + ListeningQueue.java
  + TrackDAO.java
  + Play.java
  + PlayAlbum.java
  + PlayPlaylist.java
  + AddToQueue.java
  + Skip.java
* **NAVIGATION**
  + Page.java
  + Navigator.java
  + PrevPage.java
  + NextPage.java
  + ResultsContainer.java
  + Search.java
* **MAIN** (UTILITY)
  + ContextListener.java
  + Uploader.java
  + DateFormatter.java

# **3. CLASS INTERFACES GLOSSARY**

## **3.1 CLASS DIAGRAM TO EER DIAGRAM**

Partendo dal Class Diagram mostrato in precedenza nel RAD e nel SDD, effettuiamo le dovute modifiche per poter mappare le classi su delle entità da rendere persistenti su database relazionale SQL.



Rimuoviamo i seguenti attributi che NON andranno memorizzati su database bensì su file system. Il razionale dietro tale scelta è discusso in dettaglio nel SDD.

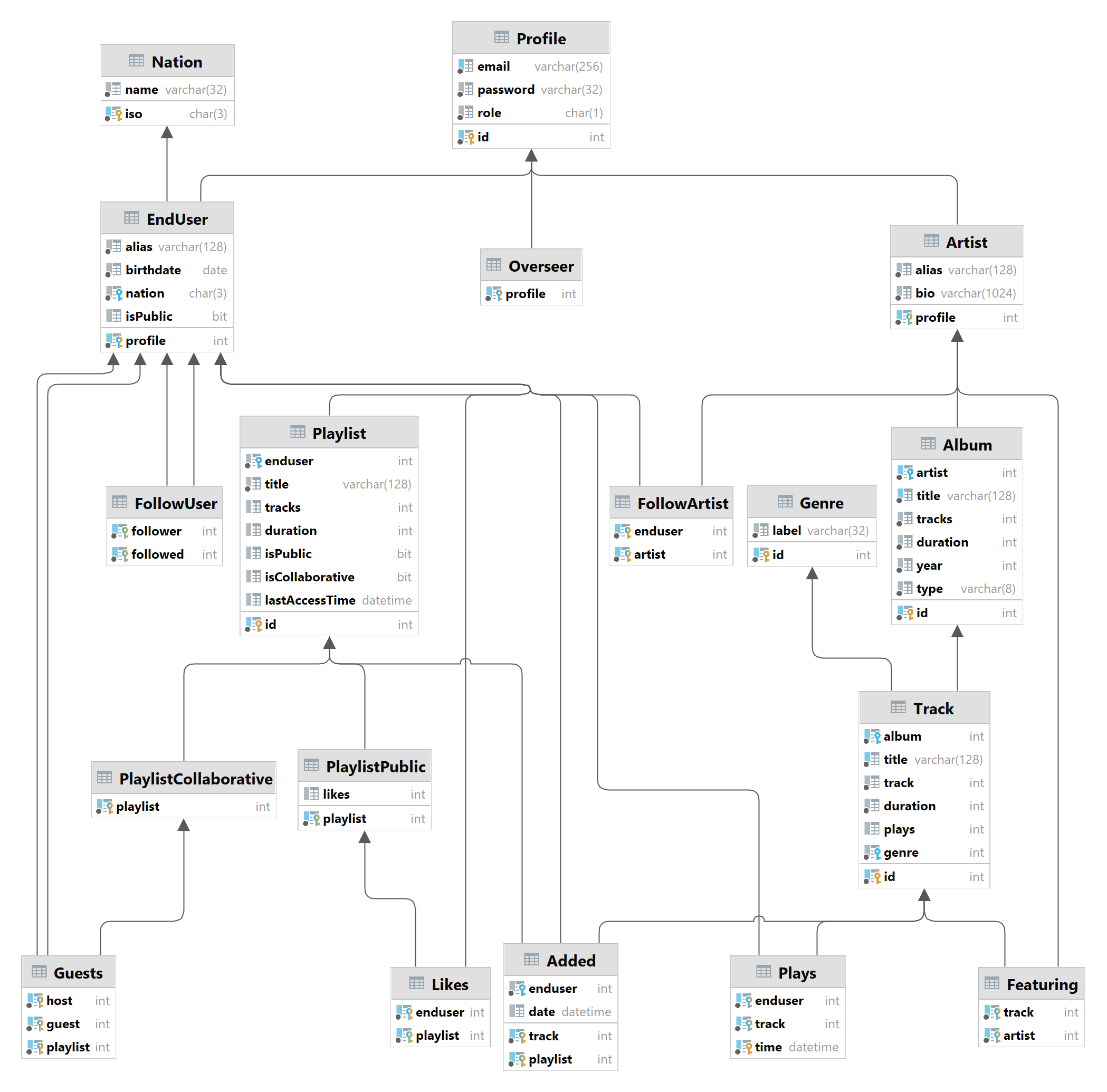
* *profile\_picture* della classe **Profile**
* *cover\_art* della classe **Playlist**
* *cover\_art* della classe **Album**
* *audio* della classe **Track**

Decidiamo di trattare gli attributi *nationality* della classe **User** e *genre* della classe **Track** come delle entità separate ai fini della normalizzazione del modello relazionale.

Modifichiamo inoltre la gerarchia che coinvolge le classi **Profile**, **Artist** e **User** includendo anche **Overseer** per facilitare la gestione delle credenziali e in seguito del login alla piattaforma.

Dotiamo infine le entità principali di un id che funge da chiave primaria.

Questo è il diagramma ER risultante generato dal tool integrato di IntelliJ:



## **3.2 INTERFACES SPECIFICATION**

## **3.2.1 BEAN**

|  |  |
| --- | --- |
| Nome Classe | ProfileBean |
| Modulo | Profile |
| Variabili di Istanza | -id: int  -email: String  -password: String  -role: Role |
| Descrizione | Questa entity contiene i metodi per creare un oggetto di tipo *Profile*, con i relativi metodi per ottenere e impostare i suoi parametri. |
| Firme dei Metodi | * ProfileBean(role: Role): ProfileBean * ProfileBean(id: int, email: String, password: String, role: Role): ProfileBean * getId(): int * getEmail(): String * getPassword(): String * getRole(): Role * setId(id: int): void * setEmail(email: String): void * setPassword(password: String): void * setRole(role: Role): void |
| Pre e Post Condizioni | * ProfileBean(role)   **pre:** true  **post:** result.id = null and result.email = null and  result.password = null and result.role = role   * ProfileBean(id, email, password, role)   **pre:** true  **post:** result.id = id and result.email = email  and result.password = password and result.role = role   * getId()   **pre:** true  **post:** result = this.id   * getEmail()   **pre:** true  **post:** result = this.email   * getPassword()   **pre:** true  **post:** result = this.password   * getRole()   **pre:** true  **post:** result = this.role   * setId(id)   **pre:** true  **post:** result.id = id   * setEmail(email)   **pre:** true  **post:** result.email = email   * setPassword(password)   **pre:** true  **post:** result.password = password   * setRole(role)   **pre:** true  **post:** result.role = role |
| Invarianti |  |

|  |  |
| --- | --- |
| Nome Classe | UserBean<<extends>>ProfileBean |
| Modulo | Profile |
| Variabili di Istanza | -alias: String  -birthdate: String  -nation: NationBean  -isPublic: boolean  -playlists: Collection<PlaylistBean>  - likedPlaylists: Collection<PlaylistBean>  -artists: Collection<ArtistBean>  -followers: Collection<UserBean>  -following: Collection<UserBean> |
| Descrizione | Questa entity contiene i metodi per creare un oggetto di tipo *User*, con i relativi metodi per ottenere e impostare i suoi parametri. |
| Firme dei Metodi | * UserBean(): UserBean * UserBean(id: int, email: String, password: String, alias: String, birthdate: String, nation: NationBean, isPublic: boolean): UserBean * getAlias(): String * getBirthdate(): String * getNation(): NationBean * isPublic(): boolean * getPlaylists)(): Collection<PlaylistBean> * getLikedPlaylists(): Collection<PlaylistBean> * getArtists(): Collection<ArtistBean> * getFollowers(): Collection<UserBean> * getFollowing(): Collection<UserBean> * setAlias(alias: String): void * setBirthdate(birthdate: String): void * setNation(nation: NationBean): void * setPublic(aPublic: boolean): void * setPlaylists(playlists: Collection<PlaylistBean>): void * setLikedPlaylists(likedPlaylists: Collection<PlaylistBean>): void * setArtists(artists: Collection<ArtistBean>): void * setFollowers(followers: Collection<UserBean): void * setFollowing(following: Collection<UserBean): void |
| Pre e Post condizioni | * context UserBean::UserBean()   **pre:** true  **post:** result.id = null and result.email = null and  result.password = null and result.alias = null and  result.birthdate = null and result.nation = null and  result.isPublic = null   * context UserBean::UserBean(id, email, password, alias, birthdate, nation, isPublic)   **pre:** true  **post:** result.id = id and result.email = email and  result.password = password and result.alias = alias and  result.birthdate = birthdate and result.nation = nation and  result.isPublic = isPublic   * context UserBean::getAlias()   **pre:** true  **post:** result = this.alias   * context UserBean::getBirthdate()   **pre:** true  **post:** result = this.birthdate   * context UserBean::getNation()   **pre:** true  **post:** result = this.nation   * context UserBean::isPublic()   **pre:** true  **post:** result = this.isPublic   * context UserBean::getPlaylists()   **pre:** true  **post:** result = this.playlists   * context UserBean::getLikedPlaylists()   **pre:** true  **post:** result = this.likedPlaylists   * context UserBean::getArtists()   **pre:** true  **post:** result = this.artists   * context UserBean::getFollowers()   **pre:** true  **post:** result= this.followers   * context UserBean::getFollowing()   **pre:** true  **post:** result = this.following   * context UserBean::setAlias(alias)   **pre:** true  **post:** result.alias = alias   * context UserBean::setBirthdate(birthdate)   **pre:** true  **post:** result.birthdate = birthdate   * context UserBean::setNation(nation)   **pre:** true  **post:** result.nation = nation   * context UserBean::setPublic(aPublic)   **pre:** true  **post:** result.isPublic = aPublic   * context UserBean::setPlaylists(playlists)   **pre:** true  **post:** result.playlists = playlists   * context UserBean::setLikedPlaylists(likedPlaylists)   **pre:** true  **post:** result.likedPlaylists = likedPlaylists   * context UserBean::setArtists(artists)   **pre:** true  **post:** result.artists = artists   * context UserBean::setFollowers(followers)   **pre:** true  **post:** result.followers = followers   * context UserBean::setFollowing(following)   **pre:** true  **post:** result.following = following |
| Invarianti |  |

|  |  |
| --- | --- |
| Nome Classe | ArtistBean <<extends>> ProfileBean |
| Modulo | Profile |
| Variabili di Istanza | -alias: String  -bio: String  -albums: Collection<AlbumBean>  -topTracks: Collection<TrackBean> |
| Descrizione | Questa entity contiene i metodi per creare un oggetto di tipo *Artist*, con i relativi metodi per ottenere e impostare i suoi parametri. |
| Firme dei Metodi | * ArtistBean(): ArtistBean * ArtistBean(id: int, email: String, password: String, alias: String, bio: String): ArtistBean * getAlias(): String * getBio(): String * getAlbums(): Collection<AlbumBean> * getTopTracks(): Collection<TrackBean> * setAlias(alias: String): void * setBio(bio: String): void * setAlbums(albums: Collection<AlbumBean>): void * setTopTracks(topTracks: Collection<TrackBean): void |
| Pre e Post Condizioni | * context ArtistBean()::ArtistBean   **pre:** true  **post**: result.id = null and result.email = null and  result.password = null, result.alias = null and  result.bio = null   * context ArtistBean()::ArtistBean(id, email, password, alias, bio)   **pre:** true  **post**: result.id = id and result.email = email and  result.password = password, result.alias = alias and result.bio = bio   * context ArtistBean()::getAlias()   **pre:** true  **post:** result = this.alias   * context ArtistBean()::getBio()   **pre:** true  **post:** result = this.bio   * context ArtistBean()::getAlbums()   **pre:** true  **post:** result = this.albums   * context ArtistBean()::getTopTracks()   **pre:** true  **post:** result = this.topTracks   * context ArtistBean()::setAlias(alias)   **pre:** true  **post:** result.alias = alias   * context ArtistBean()::setBio(bio: String)   **pre:** true  **post:** result.bio = bio   * context ArtistBean()::setAlbums(albums)   **pre:** true  **post:** result.albums = albums   * context ArtistBean()::setTopTracks(topTracks)   **pre:** true  **post:** result.topTracks = topTracks |
| Invarianti |  |

|  |  |
| --- | --- |
| Nome Classe | OverseerBean<<extends>>ProfileBean |
| Modulo | Profile |
| Variabili di Istanza |  |
| Descrizione | Questa entity contiene i metodi per creare un oggetto di tipo *Overseer*, con i relativi metodi per ottenere e impostare i suoi parametri. |
| Firme dei Metodi | * OverseerBean(): OverseerBean * OverseerBean(id: int , email: String, password: String): OverseerBean |
| Pre e Post condizioni | * context OverseerBean::OverseerBean()   **pre:** true  **post:** result.id = null and result.email = null and  result.password = null   * context OverseerBean::OverseerBean(id, email, password)   **pre:** true  **post:** result.id = id and result.email = email and  result.password = password |
| Invarianti |  |

|  |  |
| --- | --- |
| Classe | NationBean |
| Modulo | Profile |
| Variabili di Istanza | - iso: String  - name: String |
| Descrizione | Entità che rappresenta la nazionalità di un utente. |
| Firma dei Metodi | * NationBean(): NationBean * NationBean(iso: String, name: String): NationBean * getIso(): String * getName(): String * setIso(iso: String): void * setName(name: String): void |
| Pre e Post Condizioni | * context NationBean::NationBean()   **pre:** true  **post:** result.iso = null and result.name = null   * context NationBean::NationBean(iso, name)   **pre:** true  **post:** result.iso = iso and result.name = name   * context NationBean::getIso()   **pre:** true  **post:** result = this.iso   * context NationBean::getName()   **pre:** true  **post:** result = this.name   * context NationBean::setIso(iso)   **pre:** true  **post:** result.iso = iso   * context NationBean::setName(name)   **pre:** true  **post:** result.name = name |
| Invarianti | * context NationBean   **inv: if** self.iso <> null **then** self.iso.size = 3 **endif** |

|  |  |
| --- | --- |
| Nome Classe | AlbumBean |
| Modulo | Album |
| Variabili di sistema | -id: int  -title: String  -tracks: int  -duration: int  -year: int  -type: String  -artist: ArtistBean  -tracklist: Collection<TrackBean> |
| Descrizione | Questa entity contiene i metodi per creare un oggetto di tipo *Album*, con i relativi metodi per ottenere e impostare i suoi parametri. |
| Firme dei Metodi | * AlbumBean(): AlbumBean * AlbumBean(id: int, title: String, tracks: int, duration: int, year: int, type: String): AlbumBean * getId(): int * getTitle(): String * getTracks(): int * getDuration(): int * getYear(): int * getType(): String * getArtist(): ArtistBean * getTracklist(): Collection<TrackBean> * setId(id: int): void * setTitle(title: String): void * setTracks(tracks: int): void * setDuration(duration: int): void * setYear(year: int): void * setType(type: String): void * setArtist(artist: ArtistBean): void * setTracklist(tracklist : Collection<TrackBean>): void |
| Pre e Post-condizioni | * context AlbumBean()::AlbumBean()   **pre:** true  **post**: result.id = id and result.title = null and result.tracks = null and result.duration = null and result.year = null and result.type = null   * context AlbumBean()::AlbumBean(id: int, title: String, tracks: int, duration: int, year: int, type: String)   **pre:** true  **post**: result.id = id and result.title = title and result.tracks = tracks and  result.duration = duration and result.year = year and result.type = type   * context AlbumBean::getId()   **pre:** true  **post**: result = this.id   * context AlbumBean::getTitle()   **pre:** true  **post**: result = this.title   * context AlbumBean::getTracks()   **pre:** true  **post**: result = this.tracks   * context AlbumBean::getDuration()   **pre:** true  **post**: result = this.duration   * context AlbumBean::getYear()   **pre:** true  **post**: result = this.year   * context AlbumBean::getType()   **pre:** true  **post**: result = this.type   * context AlbumBean::getArtist()   **pre:** true  **post**: result = this.artist   * context AlbumBean::getTracklist()   **pre:** true  **post**: result = this.tracklist   * context AlbumBean::setId(id)   **pre:** true  **post**: this.id = id   * context AlbumBean::setTitle(title)   **pre:** true  **post**: this.title = title   * context AlbumBean::setTracks(tracks)   **pre:** true  **post**: this.tracks = tracks   * context AlbumBean::setDuration(duration)   **pre:** true  **post**: this.duration = duration   * context AlbumBean::setYear(year)   **pre:** true  **post**: this.year = year   * context AlbumBean::setType(type: String)   **pre:** true  **post**: this.type = type   * context AlbumBean::setArtist(artist)   **pre:** true  **post**: this.artist = artist   * context AlbumBean::setTracklist(tracklist):   **pre:** true  **post**: this.tracklist = tracklist |
| Invarianti | N/A |

|  |  |
| --- | --- |
| Nome Classe | PlaylistBean |
| Modulo | Playlist |
| Variabili di Istanza | -id: int  -host: UserBean  -title: String  -tracks: int  -duration: int  -likes: int  -lastAccessTime: String  -isPublic: Boolean  -isCollaborative: Boolean  -guests: Collection<UserBean>  -tracklist: Collection<AddedBean> |
| Descrizione | Questa entity contiene i metodi per creare un oggetto di tipo *Playlist*, con i relativi metodi per ottenere e impostare i suoi parametri. |
| Firme dei Metodi | * PlaylistBean(): PlaylistBean * PlaylistBean(id: int, title: String, tracks: int, duration: int, isPublic: boolean, isCollaborative: boolean, lastAccessTime: String) : PlaylistBean * getId():int * getTitle(): String * getTracks(): int * getDuration(): int * isPublic(): boolean * getLikes(): int * isCollaborative(): boolean * getLastAccessTime(): String * getHost(): UserBean * getGuests(): Collection<UserBean> * getTracklist(): Collection<AddedBean> * getTracklist(order: Order): Collection<AddedBean> * setId(id: int): void * setTitle(title: String): void * setTracks(tracks: int): void * setDuration(duration : int) : void * setPublic(aPublic: boolean): void * setLikes(likes: int): void * setCollaborative(collaborative: boolean): void * setLastAccessTime(lastAccessTime: String): void * setHost(host: UserBean): void * setGuests(guests: Collection<UserBean>): void * setTracklist(tracklist: Collection<AddedBean>): void |
| Pre e Post condizioni | * context PlaylistBean::PlaylistBean()   **pre:** true  **post:** result.id = null and result.title = null and result.tracks = null and result.duration = null and  result.isPublic = null and result.isCollaborative = null and result.lastAccessTime = null   * context PlaylistBean::PlaylistBean(id, title, tracks, duration, isPublic, isCollaborative, lastAccessTime)   **pre:** true  **post:** result.id = id and result.title = title and result.tracks = tracks and result.duration = duration and  result.isPublic = isPublic and result.isCollaborative = isCollaborative and result.lastAccessTime = lastAccessTime   * context PlaylistBean::getId()   **pre:** true  **post**: result = this.id   * context PlaylistBean::getTitle()   **pre:** true  **post**: result = this.title   * context PlaylistBean::getTracks()   **pre:** true  **post**: result = this.tracks   * context PlaylistBean::getDuration()   **pre:** true  **post**: result = this.duration   * context PlaylistBean::isPublic()   **pre:** true  **post**: result = this.isPublic   * context PlaylistBean::getLikes()   **pre:** true  **post**: result = this.likes   * context PlaylistBean::isCollaborative()   **pre:** true  **post**: result = this.isCollaborative   * context PlaylistBean::getLastAccessTime()   **pre:** true  **post**: result = this.lastAccessTime   * context PlaylistBean::getHost()   **pre:** true  **post**: result = this.host   * context PlaylistBean::getGuests()   **pre:** true  **post**: result = this.guests   * context PlaylistBean::getTracklist()   **pre:** true  **post**: result = this.tracklist   * context PlaylistBean::getTracklist(order)   **pre:** true  **post**: result = this.tracklist   * context PlaylistBean::setId(id)   **pre:** true  **post**: this.id = id   * context PlaylistBean::setTitle(title)   **pre:** true  **post**: this.title = title   * context PlaylistBean::setTracks(tracks)   **pre:** true  **post**: this.tracks = tracks   * context PlaylistBean::setDuration(duration)   **pre:** true  **post**: this.duration = duration   * context PlaylistBean::setPublic(aPublic)   **pre:** true  **post**: this.isPublic = aPublic   * context PlaylistBean::setLikes(likes)   **pre:** true  **post**: this.likes = likes   * context PlaylistBean::setCollaborative (collaborative)   **pre:** true  **post**: this.isCollaborative = collaborative   * context PlaylistBean::setLastAccessTime (lastAccessTime)   **pre:** true  **post**: this.lastAccesTime = lastAccessTime   * context PlaylistBean::setHost(host)   **pre:** true  **post**: this.host = host   * context PlaylistBean::setGuests(guests)   **pre:** true  **post**: this.guests = guests   * context PlaylistBean::setTracklist(tracklist)   **pre:** true  **post:** this.tracklist = tracklist |
| Invarianti |  |

|  |  |
| --- | --- |
| Nome Classe | AddedBean |
| Modulo | Playlist |
| Variabili di Istanza | -user: UserBean  -track: TrackBean  -playlist: PlaylistBean  -date: String |
| Descrizione |  |
| Firme dei Metodi | * AddedBean(): AddedBean * AddedBean(user: UserBean, playlist: PlaylistBean, date: String, track: TrackBean): AddedBean * getUser(): UserBean * getTrack(): TrackBean * getPlaylist(): PlaylistBean * getDate(): String * setUser(user: UserBean): void * setTrack(track: TrackBean): void * setPlaylist(playlist: PlaylistBean): void * setDate(date: String): void |
| Pre e Post condizioni | * context AddedBean::AddedBean()   **pre:** true  **post:** result.user = null and result.playlist = null and result.date = null and result.track = null   * context AddedBean::AddedBean(user, playlist, date, track)   **pre:** true  **post:** result.user = user and result.playlist = playlist and result.date = date and result.track = track   * context AddedBean::getUser()   **pre:** true  **post:** result = this.user   * context AddedBean::getTrack()   **pre:** true  **post:** result = this.track   * context AddedBean::getPlaylist()   **pre:** true  **post:** result = this.playlist   * context AddedBean::getDate()   **pre:** true  **post:** result = this.date   * context AddedBean::setUser(user)   **pre:** true  **post:** result.user = user   * context AddedBean::setTrack(track)   **pre:** true  **post:** result.track = track   * context AddedBean::setPlaylist(playlist)   **pre:** true  **post:** result.playlist = playlist   * context AddedBean::setDate(date)   **pre:** true  **post:** result.date = date |
| Invarianti |  |

|  |  |
| --- | --- |
| Nome Classe | TrackBean |
| Modulo | Track |
| Variabili di Istanza | -id: int  -title: String  -index: int  -duration: int  -plays: int  -genre: String  -album: AlbumBean  -featuring: Collection<ArtistBean> |
| Descrizione | Questa entity contiene i metodi per creare un oggetto di tipo *Track*, con i relativi metodi per ottenere e impostare i suoi parametri. |
| Firme dei Metodi | * TrackBean(): TrackBean * TrackBean(id: int , title: String, index: int, duration: int, plays: int, genre : String): TrackBean * getId(): int * getTitle(): String * getIndex(): int * getDuration(): int * getPlays(): int * getGenre(): String * getAlbum(): AlbumBean * getFeaturing(): Collection<ArtistBean> * setId(id: int): void * setTitle(title: String): void * setIndex(index: int): void * setDuration(duration: int) : int * setPlays(plays: int): void * setGenre(genre : String) : void * setAlbum(album: AlbumBean): void * setFeaturing(featuring: Collection<ArtistBean>): void |
| Pre e Post condizioni | * context TrackBean()::TrackBean()   **pre:** true  **post:** result.id = null and result.title = null and  result.index = null and result.duration = null and  result.plays = null and result.genre = null   * context TrackBean()::TrackBean(id, title, index, duration, plays, genre)   **pre:** true  **post:** result.id = id and result.title = title and  result.index = index and result.duration = duration and result.plays = plays and  result.genre = genre   * context TrackBean()::getId()   **pre:** true  **post:** result = this.id   * context TrackBean()::getTitle()   **pre:** true  **post:** result = this.title   * context TrackBean()::getIndex()   **pre:** true  **post:** result = this.index   * context TrackBean()::getDuration()   **pre:** true  **post:** result = this.duration   * context TrackBean()::getPlays()   **pre:** true  **post:** result = this.plays   * context TrackBean()::getGenre()   **pre:** true  **post:** result = this.genre   * context TrackBean()::getAlbum()   **pre:** true  **post:** result = this.album   * context TrackBean()::getFeaturing()   **pre:** true  **post:** result = this.featuring   * context TrackBean()::setId(id)   **pre:** true  **post:** result.id = id   * context TrackBean()::setTitle(title)   **pre:** true  **post:** result.title = title   * context TrackBean()::setIndex(index)   **pre:** true  **post:** result.index = index   * context TrackBean()::setDuration(duration)   **pre:** true  **post:** result.duration = duration   * context TrackBean()::setPlays(plays)   **pre:** true  **post:** result.plays = plays   * context TrackBean()::setGenre(genre)   **pre:** true  **post:** result.genre = genre   * context TrackBean()::setAlbum(album)   **pre:** true  **post:** result.album = album   * context TrackBean()::setFeaturing(featuring)   **pre:** true  **post:** result.featuring = featuring |
| Invarianti |  |