Project “Music Store”

Iteration 1: analysis and inception of design

Members:

Karl Kangur

Lukáš Škuta

Adilzhan Shukenov

Anthony Lassoudière

Ilja Gužovski

1. Analysis of business processes

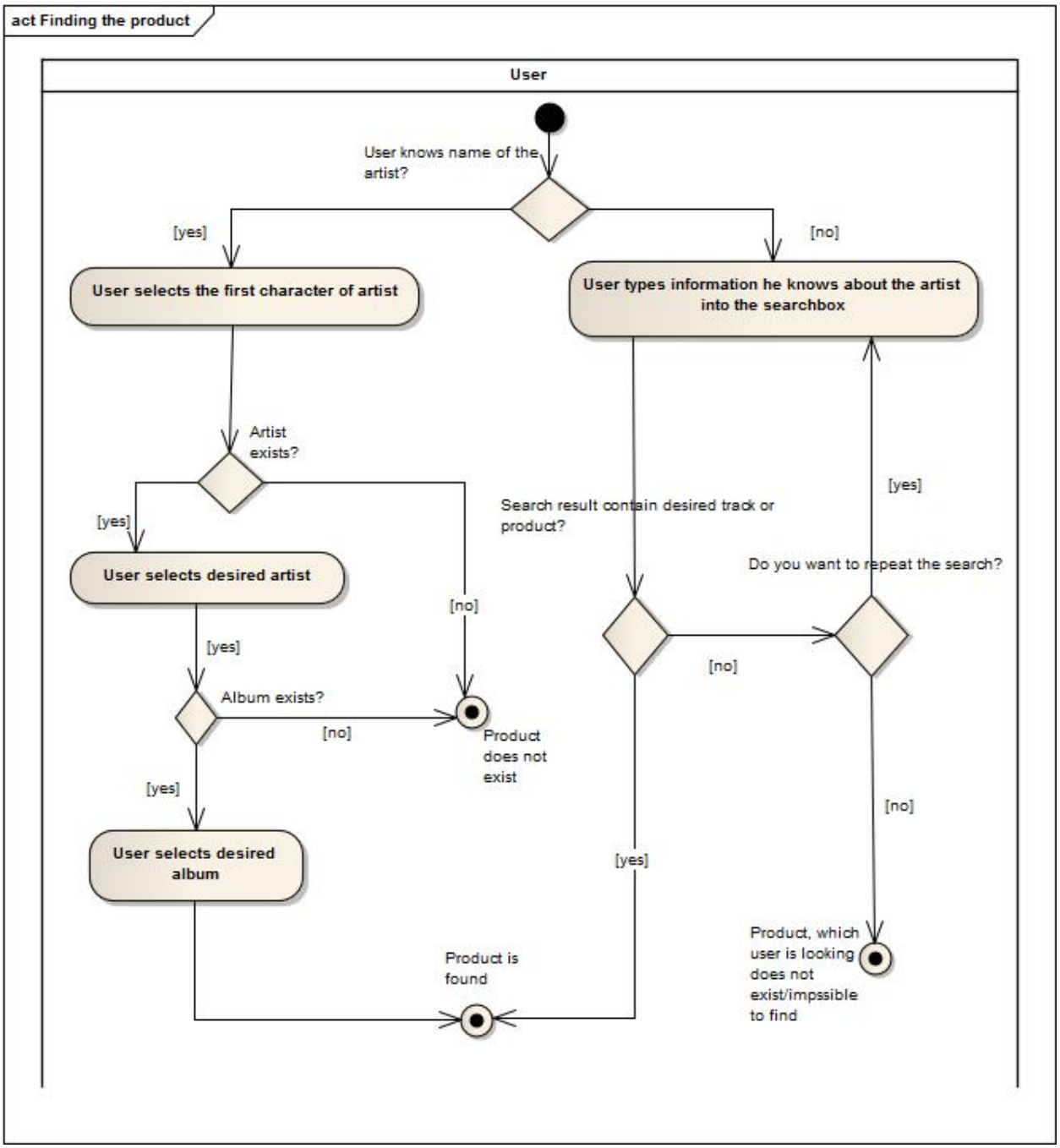
This chapter includes a description of processes that must be performed in music store during its normal operation. It includes:

- Finding the product.
- Creating a new product.
- Deleting the product.
- Changing the information about the product.

Finding the product.

This business process is the most important and will be repeated in several other processes. There are two possibilities to find you product. If the user knows the name of the artist and the album, it will be convenient for him to browse for the artist, however he could use searching method also. Searching is used if user did not remember author name, or the album, but know's the track name/lyrics.

1. Browsing method: user sees the characters in alphabetical order, which represent the first letter of the artist. By clicking the letter user sees a list of artists with the same first letter in their names. By clicking on the artist user will get the information about the artist and the list all his albums(in alphabetical order) where he is represented. By clicking on the album user will get basic information about the album (like price, quantity in different shops, length, date of release, artist who participated in its making, label and etc) and the list of tracks (ordering of tracks is predefined by label company and the artist(s)). If information about the artist and album is correct, then the search is done.
2. Searching method: user sees searchbar, where he could type and/or artist, album, track name, lyrics or product-id. After submitting user will get the list of search results. Each search result could be: an artist, an album(with name of the artist) or a track(with the name of the album and the artist).

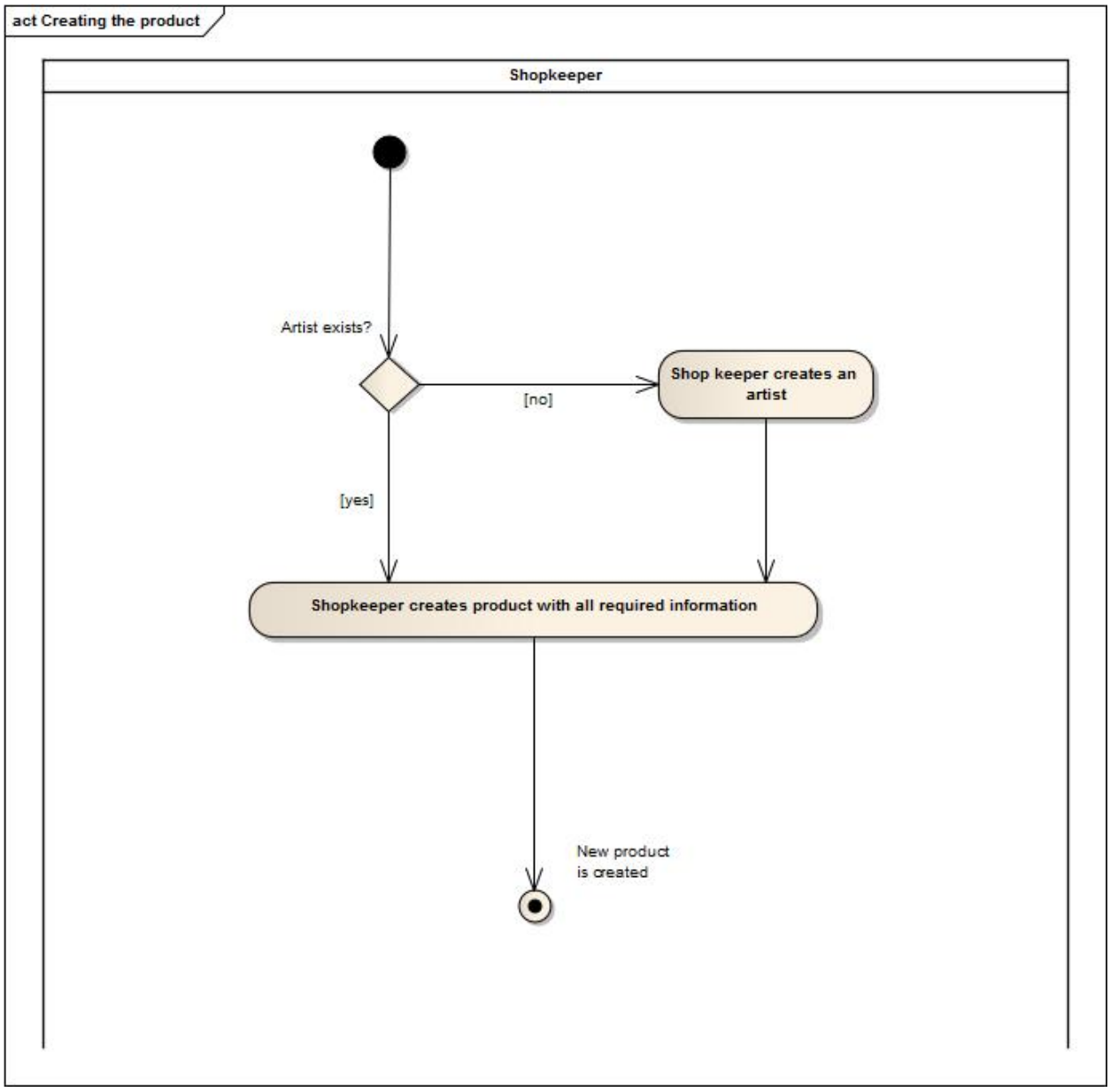


Finding the product activity diagram.

Creating the product.

Pre-requirement: Shopkeeper must be logged in.

If shopkeeper wants to create the product, he opens a page where he had to fill in artist, album name, price and quantity arrived(all required information etc). If artist does not exist, then he had to create him. If album does not exist he had to create an album with all required information. After, all the procedures, the product should be created.

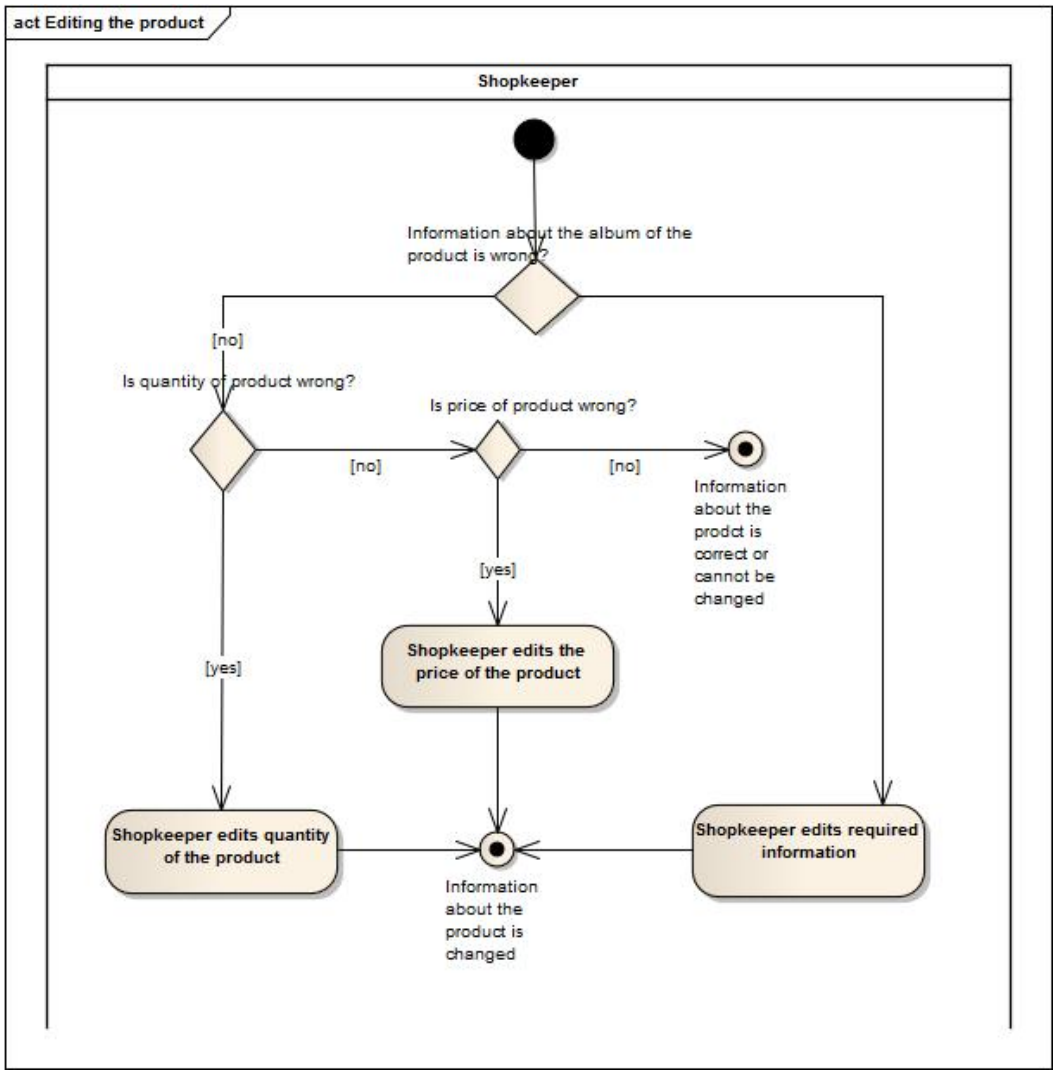


Creating the product activity diagram.

Editing the product.

Pre-requirement: Shopkeeper must be logged in.

If shopkeeper want to change information about the product(change artist name, price, quantity, album name), he/she goes to the page of the product(For instance, he could do this by typing into search bar special id of the product or use the bar-code machine). There he has 2 buttons: increase and decrease, by clicking one of them he changes the quantity of the product in his shop. If he wants to change other information about the product, then he clicks button “Edit Page”, where he is given all possible solutions, how he could make changes for the rest of the properties.

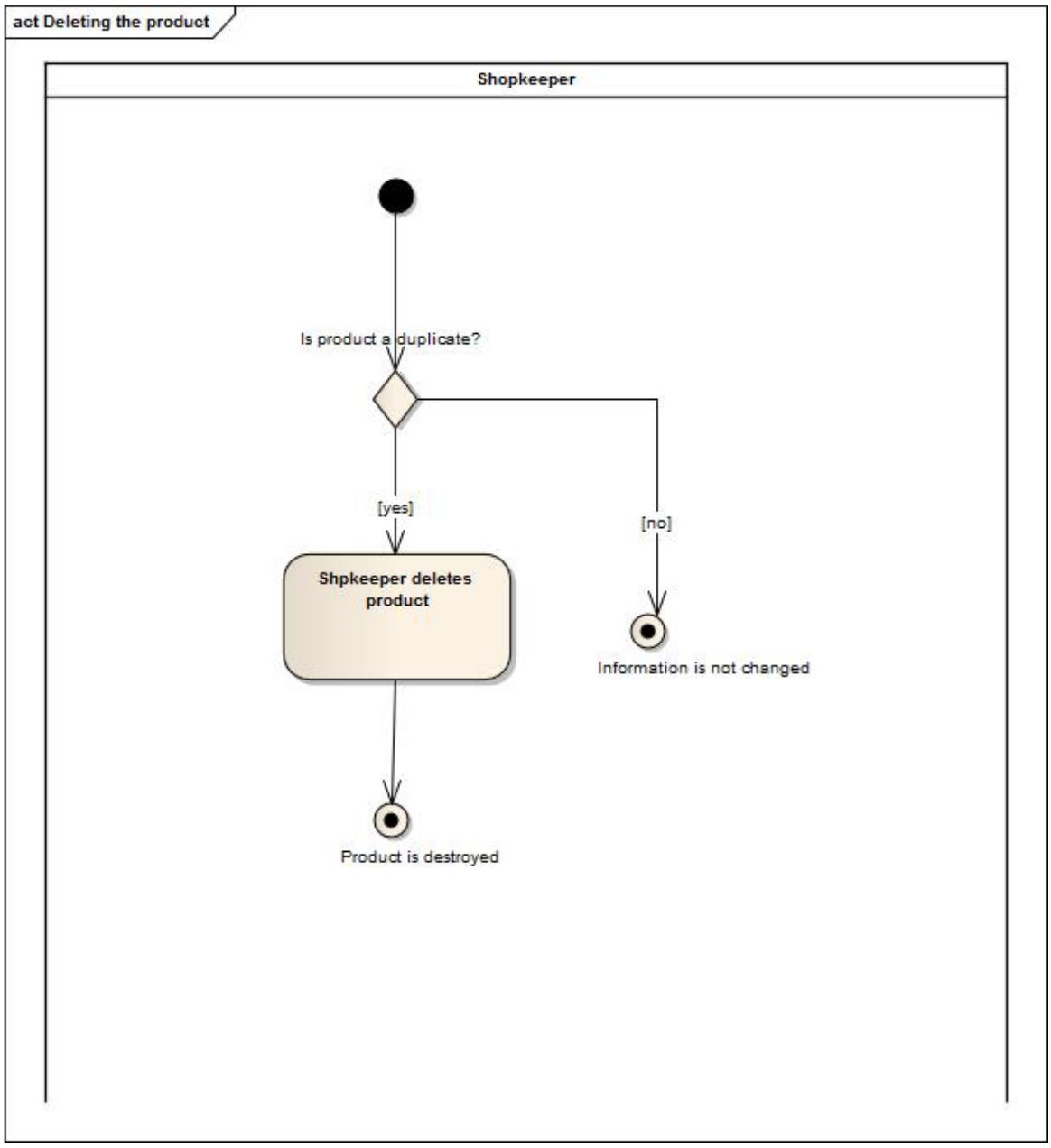


Editing the product activity diagram.

Deleting the product.

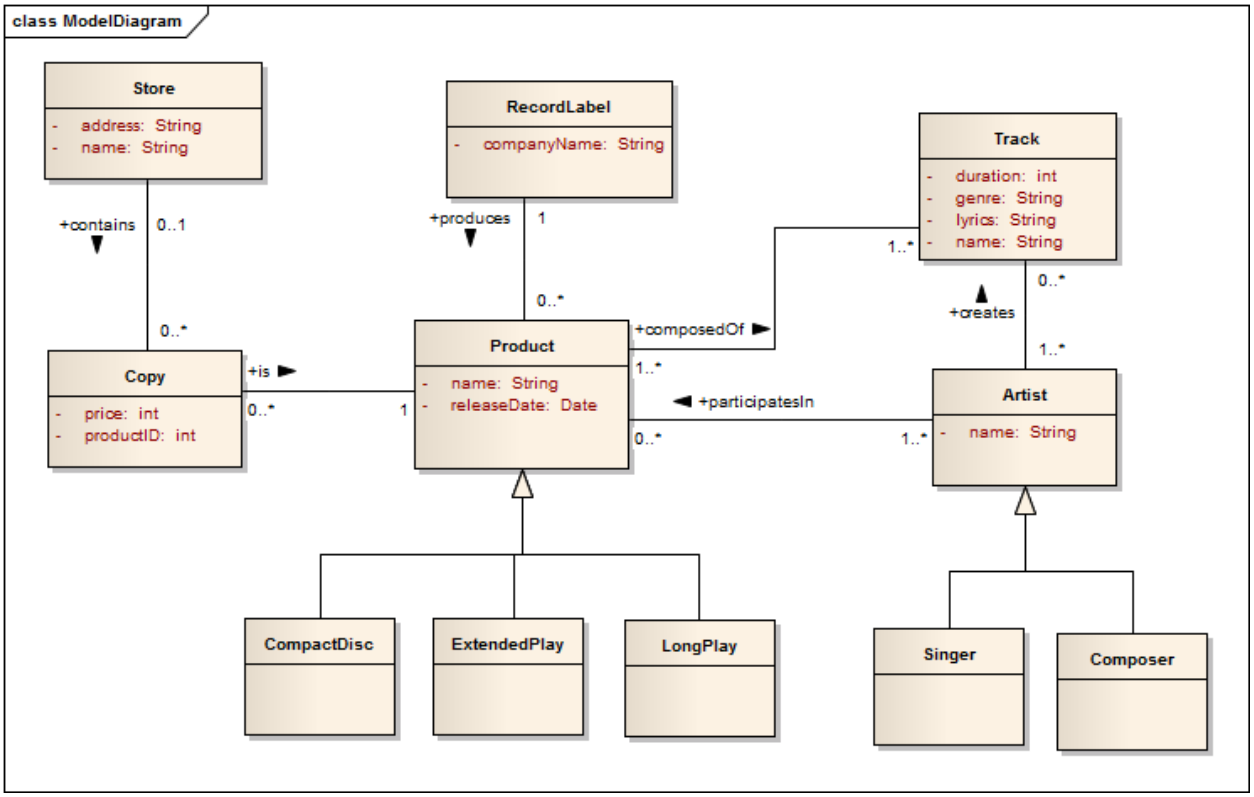
Pre-requirement: Shopkeeper must be logged in.

If shopkeeper made a mistake, by creating non-existing product, then shopkeeper can go to the edit page “Delete”, which will delete the product completely.



Deleting the product activity diagram.

2. Domain model



Domain model diagram

In the domain model we decided to separate the product from the physical product. This was decided to differentiate the price and the quantity, depending on the store where you would be able to find the product.

The stock associative class is here to indicate the stock of one particular store for one particular product. It will be useful to know the stock of the nearest stores and eventually the total number of one product, especially useful for collectors.

The product is represented by it's name and release date but we may add some informations to allow other ways to search the products (e.g : how many sales, ratings, view). We could even add some list of commentaries but this would need another class. These ideas may be implemented later.

Artist

An artist takes part in the creation of a product by producing the tracks that composed it. Artist takes part in different ways depending if they are singer or compositor.

Attributes

Attribute	Notes
name String Private	Name of the Artist

CompactDisc

Compact disc, or CD for short, is a digital optical disc data storage format. Contains up to 80 minutes of play.

Composer

The compositor is an artist who will take in charge the instrument's part of a track. There should always be compositors for a track, but "a cappella" exists.

Note : the compositor entity include single compositors as much as groups or orchestras.

Copy

The physical product is the final product, the physical part which is sold to the customer. The physical product has a quantity indicator to ensure that the product is not out of stock.

Attributes

Attribute	Notes
price int Private	Price of a particular physical product.
productID int Private	Unique identifier for particular copy of the product.

ExtendedPlay

An extended play (EP) is a musical recording that contains more music than a single, but is too short to qualify as a full studio album or LP (Long Play). The EP is typically 3 or 4 tracks.

LongPlay

The LP (Long Play), or vinyl record, is a format for phonograph (gramophone) records, an analog sound storage medium. Contains up to 45 minutes of play.

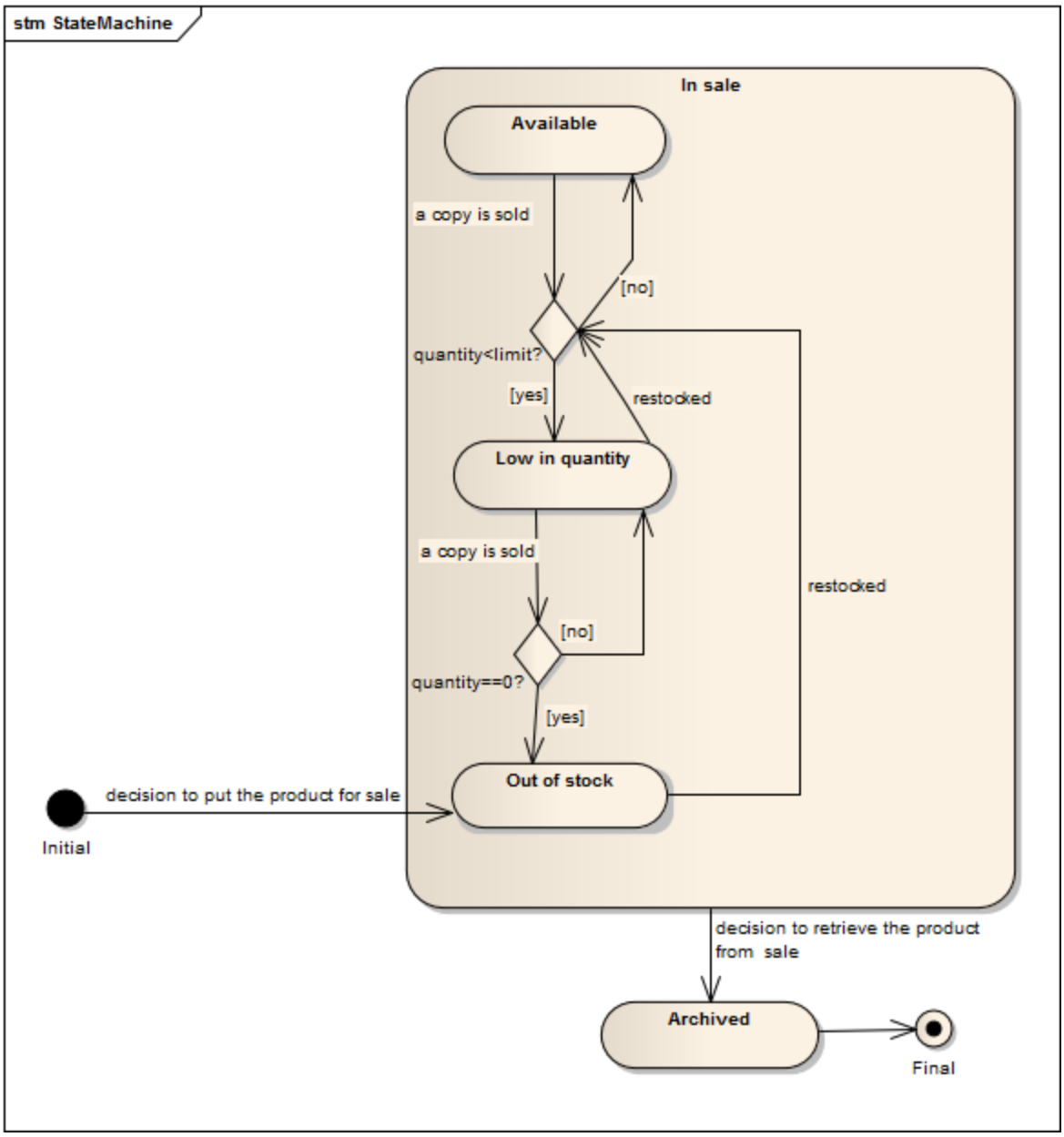
Product

The product is the numerical element representing a physical product. For this store we have three different types of products : CD (Compact Disc), EP (Extended Play) and LP (Long Play). Each product is identified by it's name and release date, helping the searching of the customers.

Attributes

Attribute	Notes
name String Private	Title of the album
releaseDate Date Private	Date when the product was first released.

StateMachine : product



RecordLabel

The record label represent the name of the company associated to the marketing and most of the time to the publishing of a product.

Attribute	Notes
companyName String Private	Official company name of the Record Label.

Singer

The singer is an artist who will take in charge the lyric's part of a track. The number of singers on a track can vary without limits and can even be zero if the track has no lyrics.

Note : the singer entity include bands, which are groups of singers under a single name.

Store

A store is the physical place where customers will be able to buy a product or pick it up if they bought it from the application.

Attribute	Notes
address String Private	Information about the location of the store (country, city, street, post code).
name String Private	Name of the store.

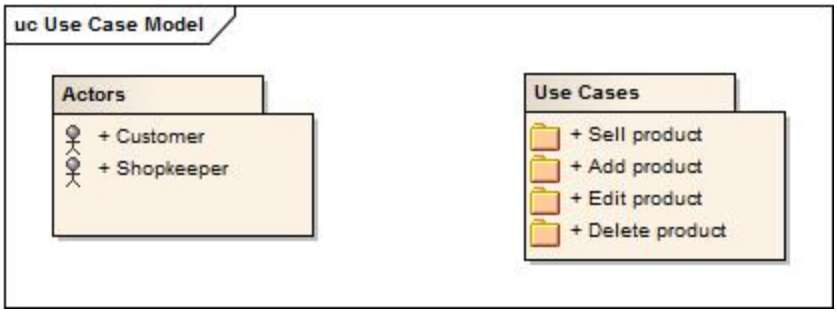
Track

A track is a part of a product. It is one of the songs on the product. The track is identified by it's name. The duration and the genre are also indicated to help for searching. The lyrics are not always linked to the track.

Attribute	Notes
duration int Private	Duration of the track in seconds.
genre String Private	Music genre of the track.
lyrics String Private	Lyrics of the track.
name String Private	Title of the track.

3. Use case model

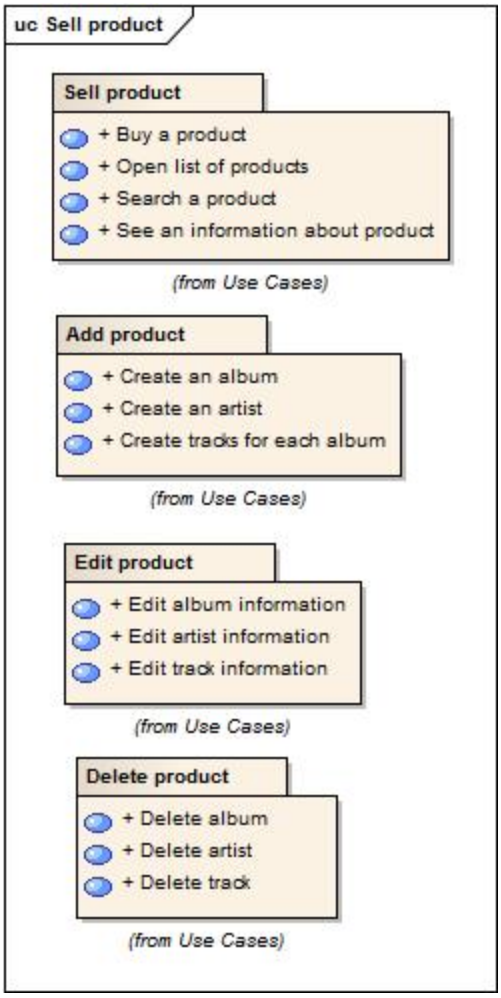
A Use Case Model describes the proposed functionality of the system. A Use Case represents a discrete unit of interaction between a user (human or machine) and the system. This interaction is a single unit of meaningful work, such as Sell Product or Delete Product.
Each Use Case describes the functionality to be built in the proposed system, which can include another Use Case's functionality or extend another Use Case with its own behavior.



Use Case packages

Use Cases

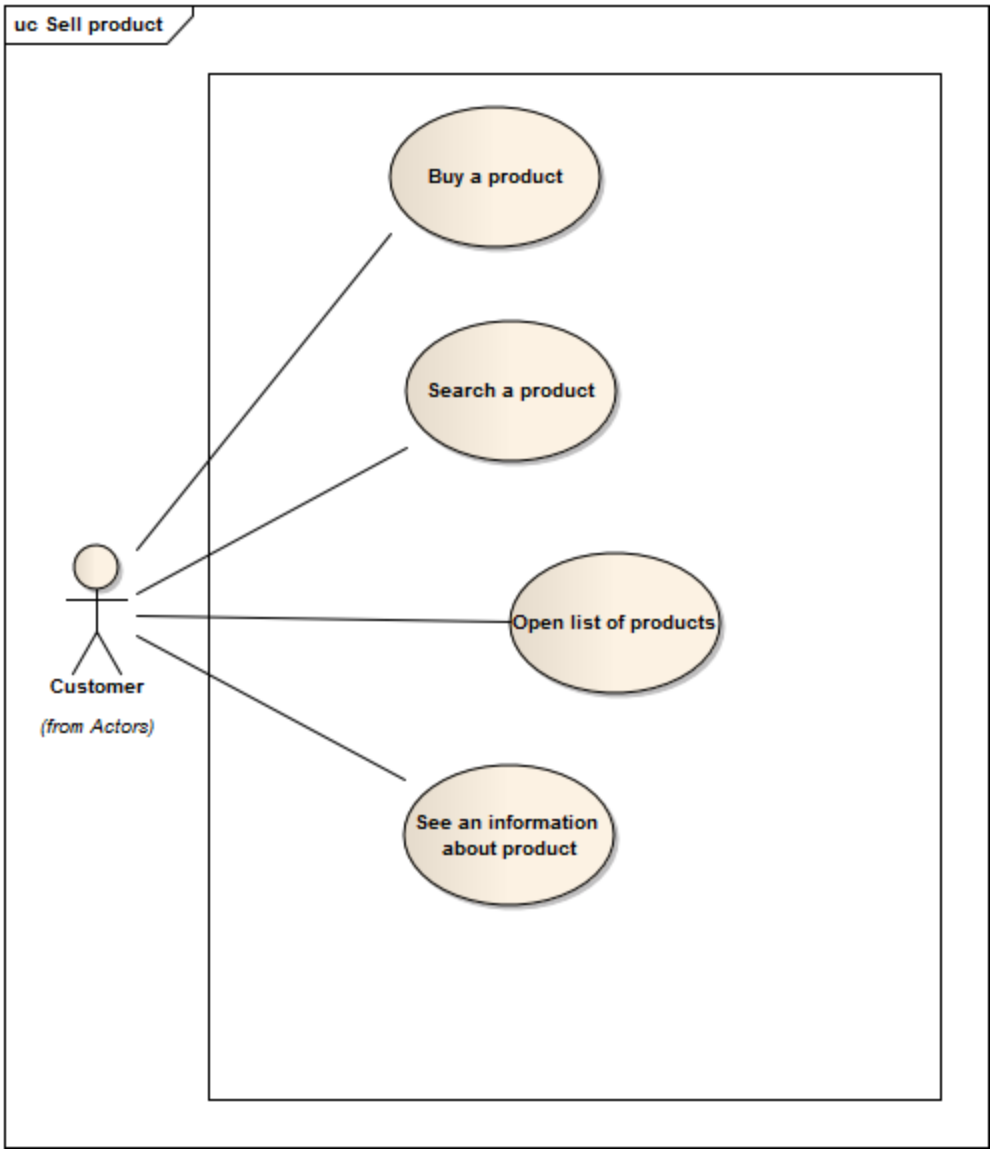
This chapter contains a description of use cases in Music Store. There are use cases which describes functions of users in Music Store.



Use Cases

Sell Product.

Chapter contains a description of the system functionality related to selling of products.



Sell product use case diagram.

Buy a product.

This function related to Shopkeeper and Customer. Customer can buy any product and at the same time Shopkeeper sell it.

1. Customer comes to Music Store.
2. Checks the price of product.
3. Pay for product
4. Shopkeeper sells the product.

Search a product.

Customer tries to find product. Shopkeeper also can search a product.

1. Customer uses system to find product.
2. Customer uses options to find product by artist, album or song
3. System checks request of customer
4. System shows the result of search
5. Customer sees the result.

Open the list of products.

In Music Store exists list of products. For customer it can be easier to choose or find product.

1. Customer clicks the button of opening the list of products.
2. System shows the list of products
3. Customer chooses product.

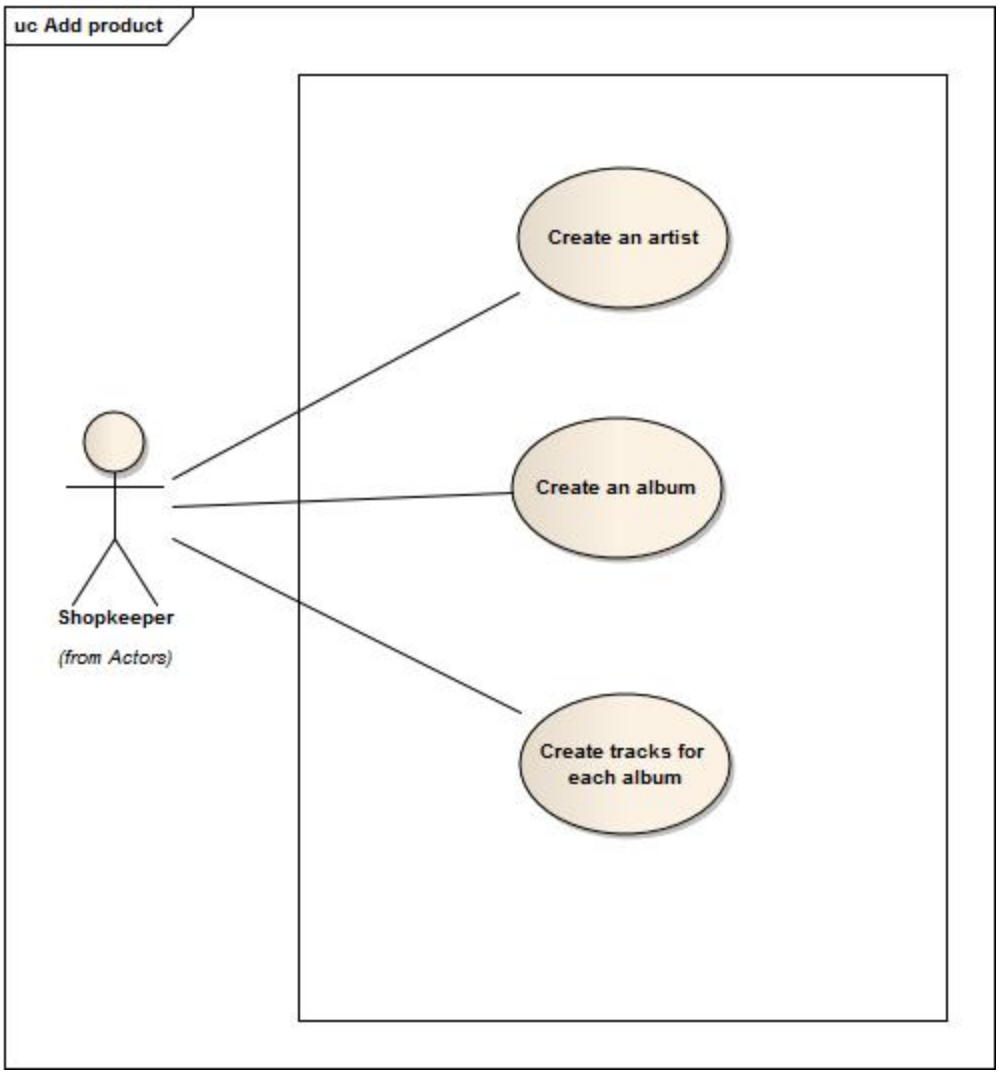
See an information of product.

Customer and shopkeeper sees an information about product. For instance, price or year of album.

1. When customer chose a product, he can find an information about it.
2. Customer clicks button of product
3. System shows the information of product

Add product.

The chapter contains description of functions related to add product.



Add product use case diagram.

Create an artist.

Shopkeeper creates an artist in database.

1. Shopkeeper opens system of Music Store.
2. Shopkeeper clicks button (Create artist)
3. System shows form of creating artist
4. Shopkeeper fills up an information about artist
5. Shopkeeper clicks the button to confirm
6. System checks an information and adds an artist.

Create an album.

Shopkeeper creates an album in database.

1. Shopkeeper opens system of Music Store.
2. Shopkeeper clicks button (Create album)
3. System shows form of creating album
4. Shopkeeper inserts an information about an album into the form.
5. Shopkeeper clicks the button to confirm.
6. System checks an information and adds an artist.

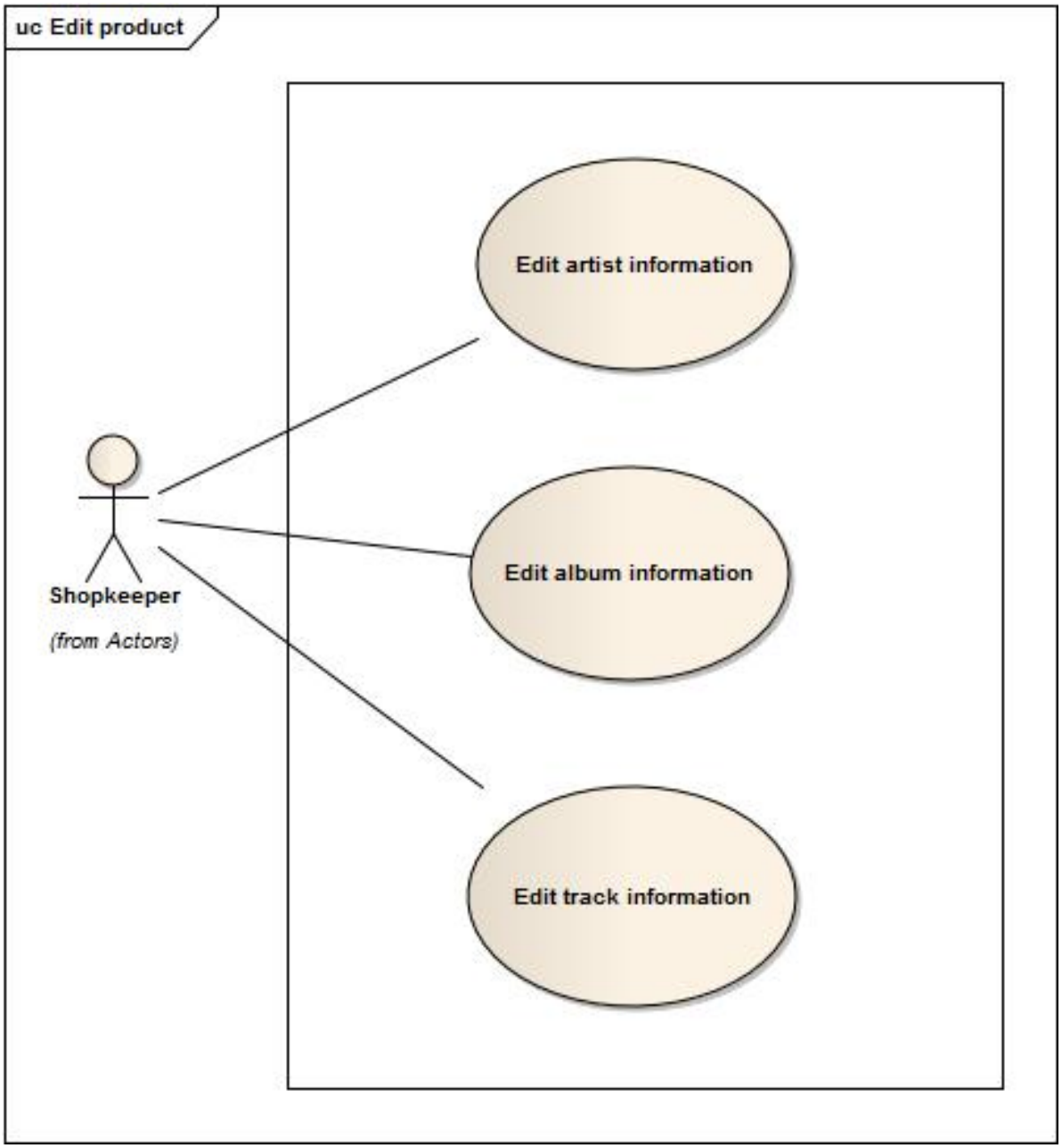
Create tracks.

Shopkeeper creates tracks in database.

1. Shopkeeper opens system of Music Store.
2. Shopkeeper clicks button (Upload file)
3. System uploads tracks into database system.
4. System checks an information and adds an track.

Edit product.

The chapter contains description of functions related to edit product.



Edit product use case diagram.

Edit artist information.

Shopkeeper edits information about artist.

1. Shopkeeper opens system of Music Store.
2. Shopkeeper clicks button of product.
3. System checks what product shopkeeper want to open.
4. System shows an information about product
5. Shopkeeper edits information about artist
6. System checks changes.
7. System edits an information about artist.

Edit album information.

Shopkeeper edits information about album.

1. Shopkeeper opens system of Music Store.
2. Shopkeeper clicks button of product.
3. System checks what product shopkeeper want to open.
4. System shows an information about product
5. Shopkeeper edits information about album
6. System checks changes.
7. System edits an information about album.

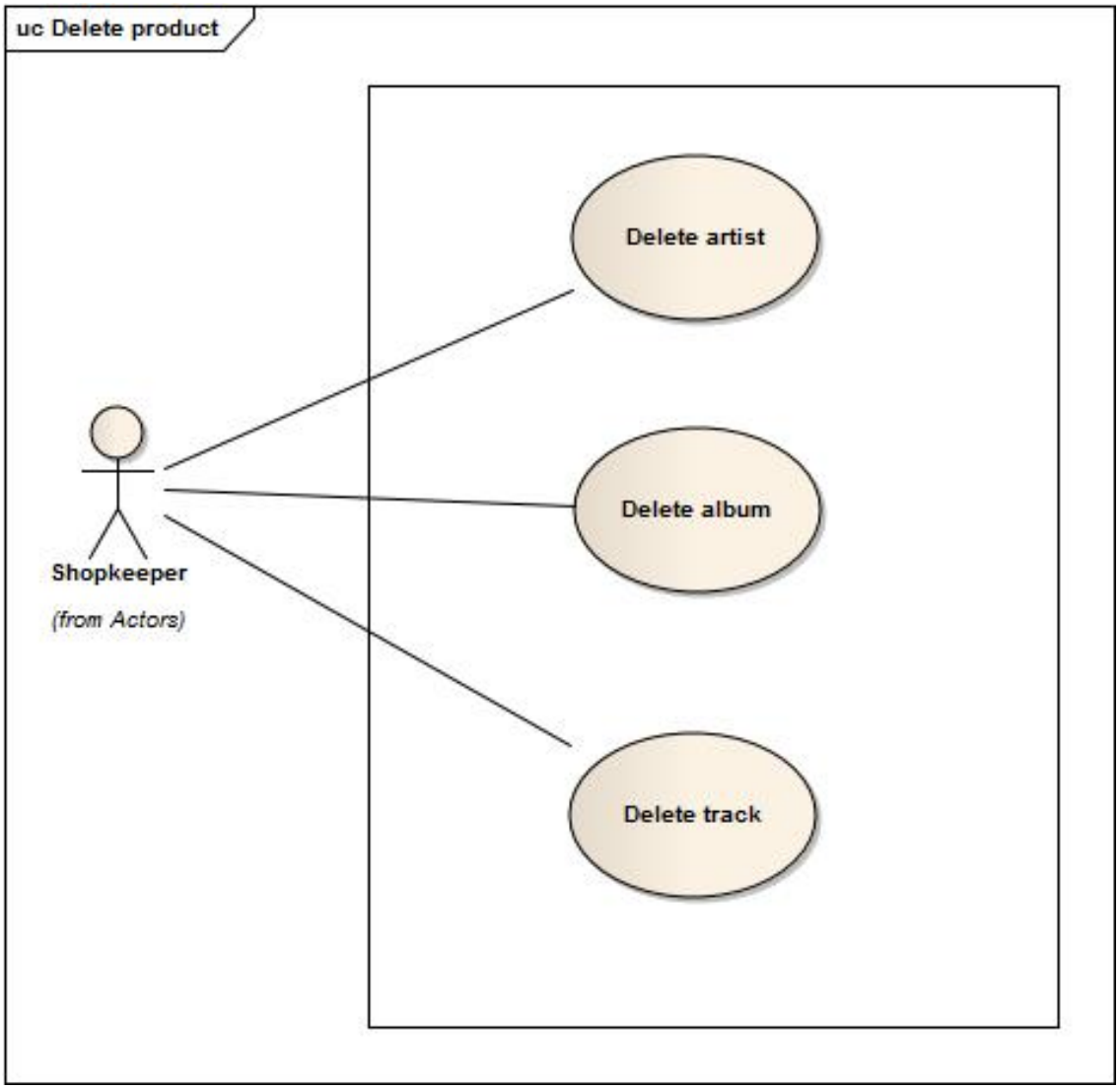
Edit track information.

Shopkeeper edits information about track.

1. Shopkeeper opens system of Music Store.
2. Shopkeeper clicks button of product.
3. System checks what product shopkeeper wants to open.
4. System shows an information about product
5. Shopkeeper edits information about track
6. System checks changes.
7. System edits an information about track.

Delete product.

In this chapter is written about process of deleting product.



Delete product use case diagram.

Delete artist.

Shopkeeper deletes an artist from database.

1. Shopkeeper opens system of Music Store.
2. Shopkeeper clicks button (Delete artist)
3. System shows message of deleting artist
4. Shopkeeper clicks the button (Delete artist)
5. Shopkeeper clicks the button to confirm
6. System checks an information and deletes an artist.

Delete album.

Shopkeeper deletes an album from database.

1. Shopkeeper opens system of Music Store.
2. Shopkeeper clicks button (Delete album)
3. System shows message of deleting artist
4. Shopkeeper clicks the button (Delete album)
5. Shopkeeper clicks the button to confirm
6. System checks an information and deletes an album.

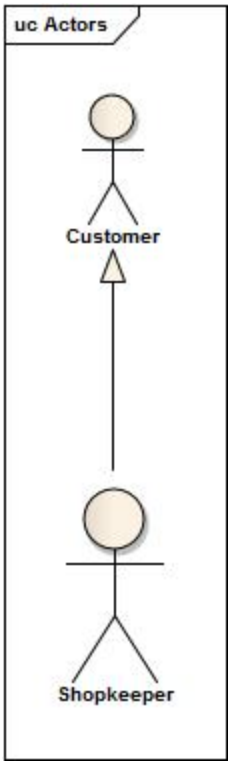
Delete track.

Shopkeeper deletes a track from database.

1. Shopkeeper opens system of Music Store.
2. Shopkeeper clicks button of product.
3. System checks what product shopkeeper wants to open.
4. System shows an information about product
5. Shopkeeper clicks he button to delete track (Delete track)
6. System shows message of deleting artist
7. Shopkeeper clicks the button (Delete album)
8. Shopkeeper clicks the button to confirm
9. System checks an information and deletes an album.

Actors

There are two actors: Shopkeeper and Customer. They are both users of system.



Actors diagram.

Shopkeeper

Shopkeeper is a person, who is responsible for uploading tracks, creating albums and artists in database of Music Store. He has such functions as selling product, add product, edit product and delete product.

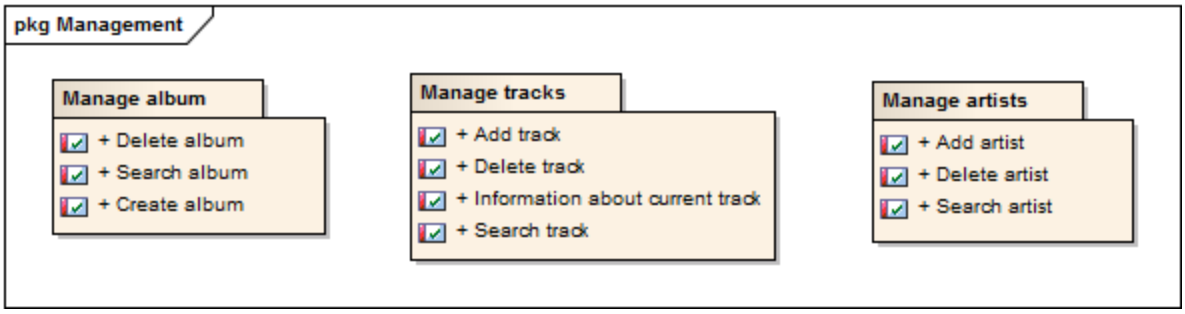
Customer

Customer is a person, who can buy, search product. He is restricted in using system of Music Store.

4. Requirements analysis

Model of requirements

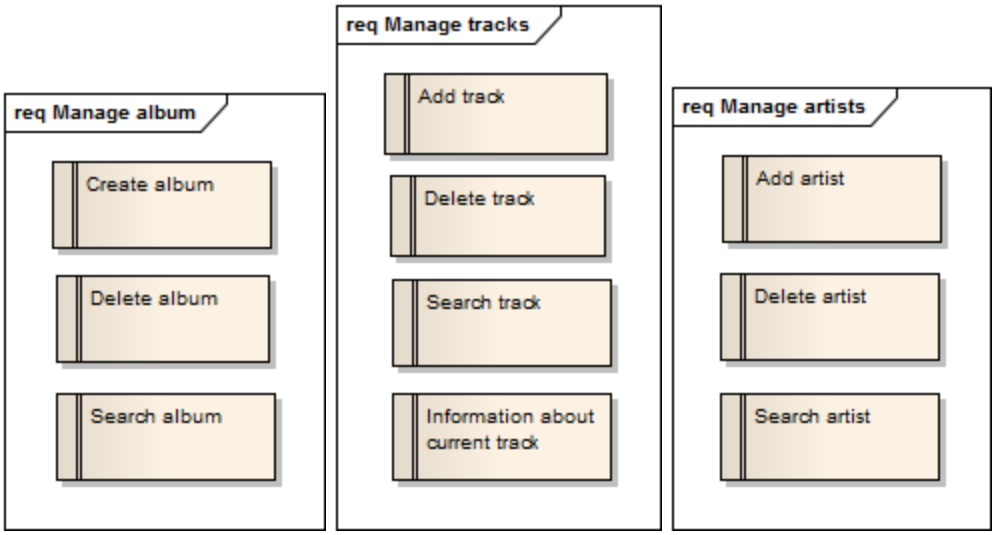
The chapter contains description of all requirements, which new system have to contain. These requirements are divided into two basic parts: to functional and nonfunctional requirements.



Requirements model diagram.

Functional requirements

The chapter contains description of function requirements that system has to contain. Functional requirements define what a system is supposed to do.



Functional requirements diagram.

Create album

- System enables to create an album, where users can store their favorite tracks.
- Every created album has to contain name of the album and the count of the tracks that current album contains.

Add track

- System enables to add a track to the existing album, adding will be executed by import

Delete track

- System enables to delete a track from existing album.

Delete album

- System enables to delete current album

Add artist

- System enables to add an artist, where adding will be executed by import

Delete artist

- System enables to delete the artist from existing album.

Search artist

- System enables to seacrh for artist according to name of artist, or his tracks

Search track

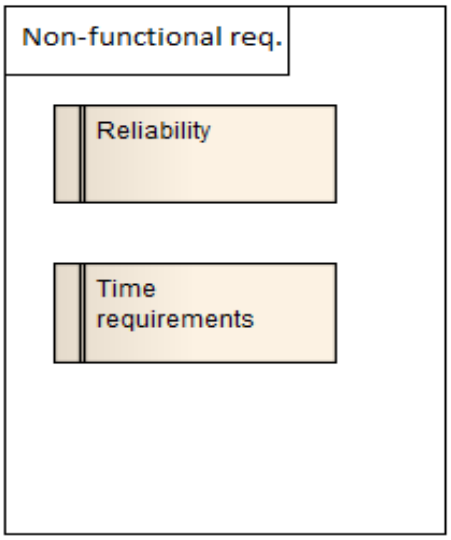
- System enables to search a track according to given parameters like name of the track or according to the author.

Information about track

- System enables to provide information about current track like name of the track, author, length and size.

Non-functional requirements

The chapter contains description of function requirements that system has to contain. Non-functional requirements define how a system is supposed to be



non-functional requirements diagram

Reliability

- The system will run 7 days a week, 24 hours a day every.

Time requirements

- Every request shouldn’t take more that 5 seconds.