**Functional design <Geef de bedrijfsnaam van de klant op.>**

**Project:** <Geef de projectnaam op.>

**Client:** <Geef de bedrijfsnaam van de klant op.>

**Projectnumber:** <Geef het projectnummer op.>

**Auteur:** <Geef de auteursnaam op.>

**Date:** <Geef de datum op wanneer het document opgesteld is.>

**Version:** <Geef het versienummer op.>

The undersigned declare their agreement with the content of this functional design.

**Client Projectmanager**

***Initial Seen: Initial Seen:***

Date: <Geef de datum op.> Date: <Geef de datum op.>

Place: <Geef de plaats op.> Place: <Geef de plaats op.>

Functional design document

Inhoud

[PROJECT MANAGEMENT 2](#_Toc498596712)

[PROJECT DETAILS 2](#_Toc498596713)

[PROJECT DESCRIPTION 2](#_Toc498596714)

[REQUIREMENTS 2](#_Toc498596715)

[CONTENT OF A REQUIREMENTS DOCUMENT 3](#_Toc498596716)

[USE CASE DIAGRAM AND SCENARIOS 3](#_Toc498596717)

[WIREFRAMES 4](#_Toc498596718)

[PERSISTENT DATA 4](#_Toc498596719)

[APPENDICES 5](#_Toc498596720)

[APPENDICE A 5](#_Toc498596721)

# PROJECT MANAGEMENT

* Who are the members of the project.
* What are the responsibilities of each member within the project.
* How do the project members communicate with each other
* When do the project members have a regular meeting

# PROJECT DETAILS

* Name of project/assignment
* Name of the client
* Name of the contractor (You)

# PROJECT DESCRIPTION

* How is it done until this project within the organization of the client (Short impression of the current workflow/situation).
* What Is the intention of the project.

# REQUIREMENTS

* THINGS AND ACTIONS A FINISHED PRODUCT MUST HAVE OR MUST BE ABLE TO PERFORM

Requirements can be divided in the following categories:

* Must
* Should
* Could
* Won’t

We called this: MoSCoW

## CONTENT OF A REQUIREMENTS DOCUMENT

* Name of project/assignment
* Name of the client
* Name of the contractor (You)
* Short impression of the current workflow/situation
* Short impression of what the client expects from your results
* The bullets of the requirements (MoSCoW)

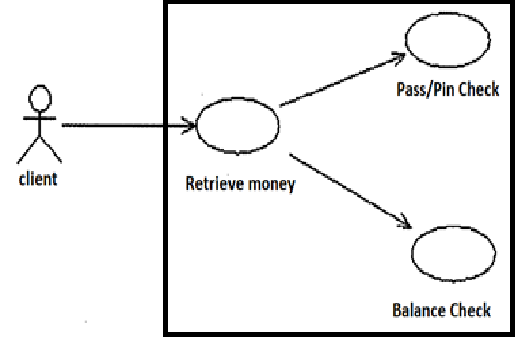
# USE CASE DIAGRAM AND SCENARIOS

Based on the required functionality of the system.

Functionality: what the system DOES ( NOT how it will do it)!!

How to make them:

1. Which ACTOR(s) will work with the system?
2. What functionality will an ACTOR use/have available.
3. Each functionality translates to a use case.
4. When a (sub)functionality is used by multiple actors you must make it a separate use case.
5. Each use case diagram has the following general outline



1. For each use case you should write a scenario. (What has to be done to use the functionality in steps).
2. The general outline of a scenario:

Name: << name of the use case >>

Actor : << name of the actor(s) who use this >>

Pre : << conditions which should be satisfied before the use case can be used >>

Description: << the normal flow of the use case in steps (functional) >>

Alternative: << the alternative flow(s) if possible (e.g.: error flow)

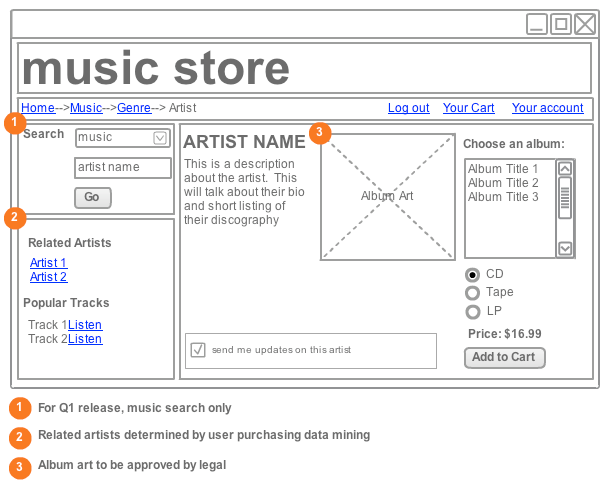
Result: << the expected result of the normal flow>>

You should check if the use cases cover ALL the requirements ( use a matrix to show that)

# WIREFRAMES

Sketch the primary forms you will use in your system. Where will be which information.

Example:



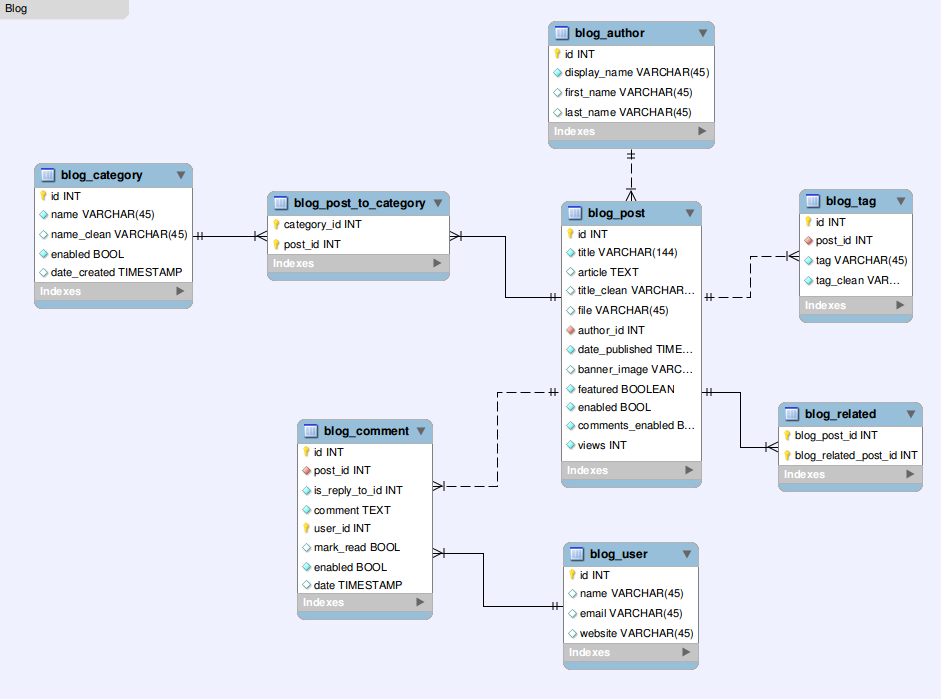
You should check if the wire frames cover ALL the use cases ( use a matrix to show that)

# PERSISTENT DATA

Description of which data will be stored and can be read/written to.

N.B. The type of storage that will be used is NOT relevant at this moment (This will be relevant in the technical design paper)

The UML diagram which can be used is an ERD.



A less formal description is sometimes acceptable. (during first year of the education) e.g.

Student(Number, Name, Address, Zipcode, Telephone,…)

Employee(Number, Name, Address, .., Rol,…)

Subject(Number, Name)

Group(Number, Name, Mentor)

# APPENDICES

## APPENDICE A

Approval.

If You agree with the content of this Functional Design Document, please return a signed copy of it.