**Functional design Vending Machine**

**Project: vending Machine**

**Client: Proftaak 2018**

**Projectnumber: 1**

**Auteur: Maarten Jakobs & Max van den Boom**

**Date: 16-12-2018**

**Version: 1**

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](#_Toc532759549)

[PROJECT DETAILS 2](#_Toc532759550)

[PROJECT DESCRIPTION 2](#_Toc532759551)

[REQUIREMENTS 3](#_Toc532759552)

[Must 3](#_Toc532759553)

[Schould 3](#_Toc532759554)

[Could 3](#_Toc532759555)

[Won’t 3](#_Toc532759556)

[CONTENT OF A REQUIREMENTS DOCUMENT 3](#_Toc532759557)

[USE CASE DIAGRAM AND SCENARIOS 3](#_Toc532759558)

[WIREFRAMES 4](#_Toc532759559)

[PERSISTENT DATA 5](#_Toc532759560)

[APPENDICES 5](#_Toc532759561)

[APPENDICE A 5](#_Toc532759562)

# PROJECT MANAGEMENT

Dit project is gemaakt door Maarten Jakobs en Max van den Boom wij zijn allebei studenten van Roc ter AA en doen beide de opleiding applicatieontwikkeling. We hebben de taken van dit project goed verdeelt Maarten doet volledig de Web kant en een beetje documentatie en Max doet volledig de Arduino kant en de meerderheid van de documentatie. Wij communiceren onze taken goed door mondeling elke dag even 10 minuten te praten over het project en hoe ver we er in staan daarnaast hebben we elke week samen met onze project begeleider een gesprek waar we afspraken maken.

# PROJECT DETAILS

Naam van het project is: My vending  
Naam van de client is: Roc ter AA  
Naam van de makers: Maarten Jakobs en Max van den Boom

# PROJECT DESCRIPTION

Wat is my vending. My vending is een makkelijke website waar je van een afstand een lijstje kan maken met producten die je uit een snoep/drank automaat wilt hebben. Waarom via een website het voordeel van een website is dat je via internet kan betalen en dus niet meer met kleingeld over straat moet. Ook ga je met dit concept de rijen tegen die bij die automaten kan staan. Daarnaast is het voor de beheerders van de machines makkelijk om in de gaten te houden welke producten er veel worden gekocht en makkelijker inschatten hoeveel ze moeten inkopen.

# REQUIREMENTS

## Must

* Kunnen betalen via een credit system
* Kunnen inloggen op de website
* QR code generen
* QR kunnen lezen
* Items kunnen toevoegen in lijstje
* Dingen uit het automaat kunnen halen
* Admin pagina

## Schould

* Kunnen zien waar de machines staan op een kaart
* Mobile friendly zijn
* Responsive
* Bestel geschiedenis zien
* Het bestellen van een product kunnen anuleren.

## Could

* Led Strip die aangeeft of er een product uit komt
* Totaal aantal prodcuten uit machine kunnen zien via een counter

## Won’t



## 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)

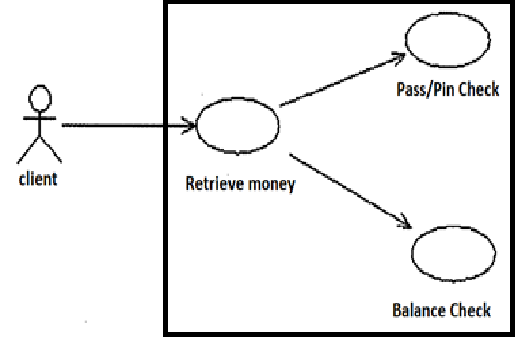
# USE CASE DIAGRAM AND SCENARIOS

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)

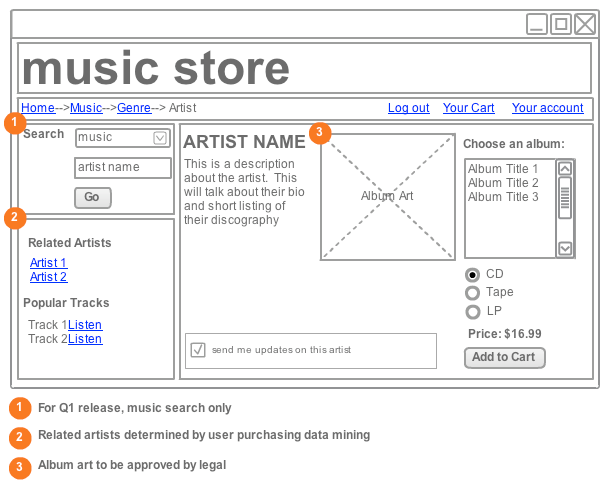
Result: << the expected result of the normal flow>>

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:



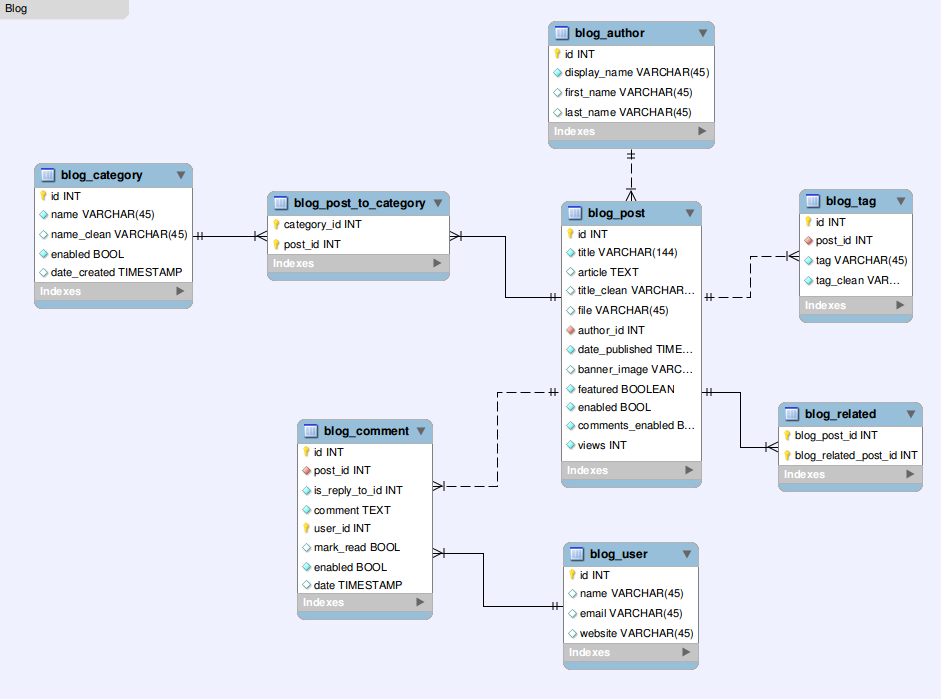
You should check if the wire frames cover ALL the use cases ( use a matrix to show that)

# 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.