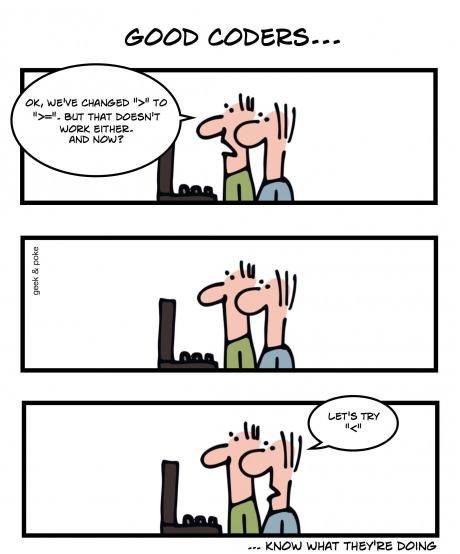
Enriching the CVM-4 with a graphical interface

productdocument

****

Student: Youri Claes

629836

Vak: Inleiding Software Engineering, D-B-INSE-O

Docent: Jos Onokiewicz

17-1-2020

Documenthistorie

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Versie** | **Wie** | **Veranderingen** |
| 9-1-2020 | 1v | Youri Claes | De initiale niet gedocumenteerde productdocument bijwerken |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Samenvatting

<Zakelijke samengevatte beschrijving van het project, eisen, wat is gerealiseerd, resultaat testen, belangrijkste aanbevelingen, vermijd de ik-stijl, moet op zichzelf leesbaar zijn zonder kennis van het project, niet verwijzen naar onderdelen in dit document, gebruik alinea's, opsommingen>

## Eisen

Initiale eisen:

* Het maken van een GUI wrapper om CLI programma
* Functie die print naar het GUI programma
* Uitbreiding van oorspronkelijk programma

De eisen waren aanvankelijk het maken van een GUI wrapper om een CLI programma. Echter kwam wij er al gauw achter dat dit niet te doen was met de API die wij gebruikte. Van daar dat we de eisen heb veranderd naar het uitbreiden en compleet nieuw ontwikkelen van een GUI programma met de functionaliteit van de CVM als basis deze is oorspronkelijk ontwikkeld is door Jos Onokiewicz.

Eisen:

* Registreer en verwerk van het ingeworpen geldstuk.
* Registreer van pinpas.
* Registreer van pincode als pinpas is geregistreerd.
* Check of genoeg is betaal bij selecteren van een product uit machine.
* Check of pinpas en pincode kloppen.
* Wisselgeld teruggeven wanneer dat benodigds.

# Voorwoord

Goedendag, beste lezer.

Dit vak liep eigenlijk prefect. Door dat ik al aardig wat programmeer kennis had, hetzij voor mijn plezier en niet in C maar in C++ en shell. Heeft dit mij enorme geholpen. Door het eerste blok heen heb ik echt de meeste finesses geleerd. Hier door was het ook niet moet om het tentamen met een goed resultaat af te sluiten. Aangezien ik eigenlijk wat meer uitdaging wou ging ik het mezelf moeilijker maken en de opdrachten uitbreiden. Ook ging ik eigenproject je opzetten. Hierdoor leerde ik eigenlijk meer dingen dat noodzakelijk was voor dit vak. Misschien dat ik volgende keer zou kijken of ik niet meteen al de lat wat hoger kan leggen.

De opdracht die we kregen wat het maken van een state machine voor een apparaat naar keuze. Maar ook hier ging ik het me zelf moeilijker maken.

Ik ging in overleg met Jos win32api gebruiken om een GUI te maken voor de CVM-4. Het is idee was natuurlijk en leuk, maar de uitvoering was wat minder. Winapi is enorm lastig en kan als een taal an sich worden bezien. Naar veel research, documentatie lezen en vragen stellen aan mensen die er verstand van hadden is het me gelukt. Door het gebruik van winapi heb ik niet alleen veel geleerd over windows, ik heb ook enorme veel geleerd over C en hoe je je code leesbaar kun houden. Als ik het over zou doen had ik een ander api gekozen, windows api is best wel achterhaalt vandaar dat ik dan een moderne api zou gebruiken.

Al met al was dit een erg leuk vak, en ik vind het dan ook jammer dat het vak niet klaar is. Ik hen enorm veel geleerd en ga zeker als hobby door met programmeren

Youri Claes

Inhoudsopgave

[Samenvatting 3](#_Toc29999825)

[Eisen 3](#_Toc29999826)

[Voorwoord 4](#_Toc29999827)

[1. Inleiding 6](#_Toc29999828)

[2. Definitiefase 7](#_Toc29999829)

[2.1 Kennismaking en doel van Cola Vending Machine 7](#_Toc29999830)

[2.2 Functionele eisen 7](#_Toc29999831)

[2.3 Technische eisen 7](#_Toc29999832)

[2.4 Schets user interface 7](#_Toc29999833)

[3. Ontwerpen 8](#_Toc29999834)

[3.1 Architectuur 8](#_Toc29999835)

[3.2 State chart 8](#_Toc29999836)

[4. Realisate en testen 9](#_Toc29999837)

[4.1 Realisatie 9](#_Toc29999838)

[4.2 Acceptatietesten 9](#_Toc29999839)

[5. Eindresultaat en aanbevelingen 10](#_Toc29999840)

[5.1 Eindresultaat 10](#_Toc29999841)

[5.2 Aanbevelingen 10](#_Toc29999842)

# Inleiding

Dit hoofdstuk gaat in op de opbouw van dit document.

Hoofdstuk 2, gaat dieper in op de technisch aspecten en realisatie van de CVM-5

Hoofdstuk 3, gaat dieper in op hoe de deelsystemen met elkaar samenwerken

Hoofdstuk 4, In dit hoofdstuk kijken hoe de realisatie van het product ging en aan welke testen die onder worpen is

Hoofdstuk 5, spreek over het hoe het resultaat uiteindelijk geworden is en wat we de volgende keer ander zouden aan pakken

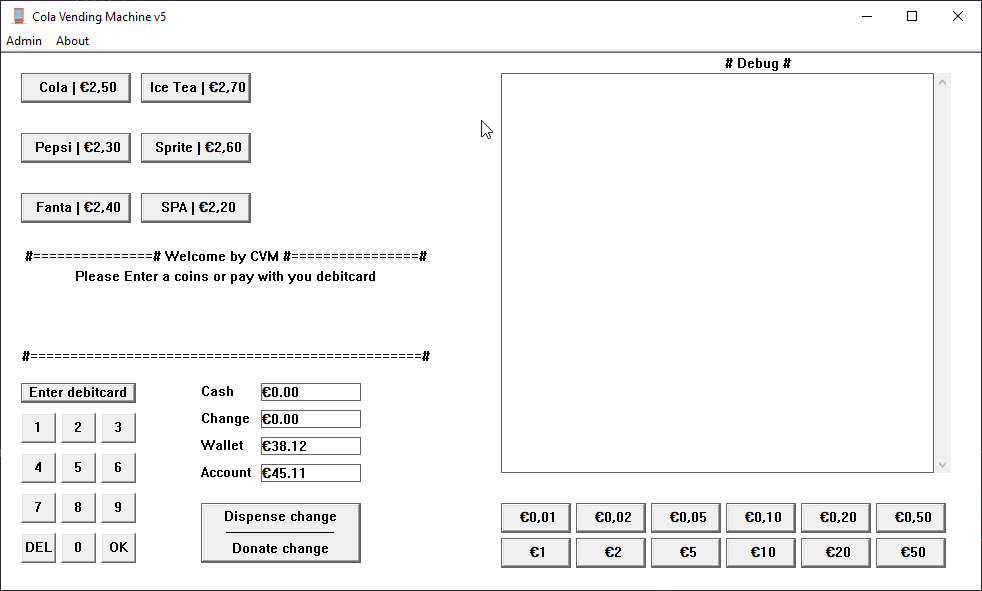
# Definitiefase

Dit hoofdstuk gaat in op de gestelde eisen aan de te realiseren besturingssoftware van Cola Vending Machine Gui

## Kennismaking en doel van Cola Vending Machine

<Korte niet-technische beschrijving apparaat, doel? Welk probleem lost het op? Door wie gebruikt? Wat levert het op?>

Het apparaat zorg voor de mogelijkheid om drankjes te kunnen kopen zonder dat het bemant hoeft te zijn. Hierdoor kunnen je 24/7 drinken kopen en hoeft de verkopen niet aanwezig te zijn. Het is mogelijk dat het zou worden gebruikt door winkels en/of winkelcentra en op soortgelijke pleken



Rechts zitten de echte componenten, Pin menu, Portemonnee, Bank, knopen voor kopen en wisselgeld knop.

Links zitten de gesimuleerde componenten munten in werpen en debug scherm

## Functionele eisen

De functionele eisen zijn als volgt:

1. Koop van frisdrank met contant geld
   1. Het geven van de frisdrank
   2. Het geven wisselgeld
2. Koop van drankje met bankpas
   1. Het geven van de frisdrank
   2. Het verwerken met bankrekening
3. Debug functie
   1. Het kunnen inzien van wat er op de achtergrond gebeurt
   2. Extra geld voor testen
4. Het schrijven naar een display
   1. Gebruiker op inlichten via menu

## Technische eisen

De technische eisen zijn als volgt:

1. De programmeertaal C moet worden gebruikt.
2. De ontwikkelomgeving Visual Studio 2019 Professional
3. Gebruikte API: winapi

## Schets user interface

Debug scherm, Weergave scherm voor geld, inwerp sleuven voor contant geld, pin knoppen, sleuf pinpas.

# Ontwerpen

Dit hoofdstuk gaat in op de Cola Vending Machine

## Architectuur

…..

<overzicht subsystemen in lijstvorm>

* …
* …

<UML + figuur titel + nr>

….

## State chart

…..

<UML + figuur titel + nr>

….

# Realisate en testen

Dit hoofdstuk bevat …

## Realisatie

<Korte weergave: Qt gebruik, instellingen, versies? Inhoud .pro file, Modulaire opbouw code, welke files? Directory structuur? Wat niet gemaakt? Bijv HAL, Bekende bugs? plantUML-gebruik, welke tools? is er codedocumentatie? Doxygen, Gebruikershandleiding beschikbaar?>

## Acceptatietesten

De volgende testen zijn …. <Volgorde? Voor wie? Hoe uitgevoerd?>

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test** | **Action/input** | **Expected result** | **Pass Fail** | **Actual result if test has failed** |
| 1. |  |  |  |  |
| 2. |  |  |  |  |
| 3. |  |  |  |  |
| 4. |  |  |  |  |
| 5. |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

<Bovenstaande tabel mogelijk in landscape, onderzijde naar buiten toe>

<Samenvattende conclusie(s) over de testen, meer testen nog te doen? Waarom niet gedaan? >

# Eindresultaat en aanbevelingen

Dit hoofdstuk bevat ….

## Eindresultaat

<moet redelijk op zichzelf leesbaar zijn>

….

## Aanbevelingen

….

<wat nog te doen? verbeteringen? Bugs oplossen? Volgende versie? ….>