**Startsemester oriëntatieverslag**

**Password Manager  
  
  
Graphical user interface, text, application, chat or text message

Description automatically generated**

**Studentnaam: Marijn Verschuren  
Studentnummer: 510936  
Klas: PDB-11  
Vakdocent: Jeroen de Kort  
  
Versie: 1.1  
Datum: 13-10-2022**

**Versiebeheer**

|  |  |  |  |
| --- | --- | --- | --- |
| **Versienummer** | **Datum** | **Auteur** | **Veranderingen** |
| *1.0* | *13-10-2022* | *Marijn Verschuren* | *Document aangemaakt* |
| *1.1* | *14-10-2022* | *Marijn Verschuren* | *Keuzes, problemen en reflectie gemaakt* |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Inhoudsopgave

[1 Inleiding 4](#_Toc116646762)

[1.1 Aanleiding 4](#_Toc116646763)

[1.2 Onderwerp 4](#_Toc116646764)

[1.3 Leeswijzer 4](#_Toc116646765)

[2 Introductie 5](#_Toc116646766)

[3 Aantonen leerdoelen 6](#_Toc116646767)

[3.1 Idee 6](#_Toc116646768)

[3.2 code 7](#_Toc116646769)

[3.3 Keuzes: 10](#_Toc116646770)

[3.4 Problemen 11](#_Toc116646771)

[3.4.1 File structure 11](#_Toc116646772)

[3.4.2 CRC 11](#_Toc116646773)

[3.5 Bronnen: 11](#_Toc116646774)

[4 Reflectie / evaluatie 12](#_Toc116646775)

[4.1 Waar ben ik trots op? 12](#_Toc116646776)

[4.2 Wat doe ik een volgende keer anders? 12](#_Toc116646777)

[4.3 Welke formatieve indicatie zou ik mezelf geven voor de oriëntatie Software? 12](#_Toc116646778)

[4.4 Welk verdiepend(e) profiel(en) kies ik en waarom? 12](#_Toc116646779)

# Inleiding

## Aanleiding

*Dit verslag laat zien hoe mijn software project werkt en welke keuzes, problemen en oplossingen daar bij komen kijken.*

## Onderwerp

*Mijn project: Password Manager word een password manager die je wachtwoorden veilig op je eigen computer kan opslaan waardoor je niet perse internet nodig hebt om bij je wachtwoorden te komen dit is handig als je geen bereik hebt maar toch wilt inloggen op een andere offline service.*

## Leeswijzer

*Onder Aantonen leerdoelen worden eerst het idee en de code uitgelegd en dan wat voor problemen en keuzes daarbij kwamen kijken. Ten slotte de bronnen die gebruikt zijn*

*Onder Reflectie word de reflectie (wat moet volgende keer anders of juist niet) van dit project gegeven en welk profiel ik ga kiezen.*

# Introductie

Ik ben Marijn Verschuren, ik ben 18 jaar oud en woon in Eindhoven. En ik heb voor ICT op de Fontys gekozen omdat programmeren een van mijn hobby’s is. In dit start semester wil ik te weten komen welk vak het beste bij mij past, al heb ik al een idee namelijk: Tech. Ik ga denk ik Tech kiezen omdat ik het leuk vind om iets fysieks te maken en omdat ik het interessant vind om op low level te programmeren. Software is wel mijn tweede keuze omdat ik programmeren in het algemeen leuk vind, alleen vind ik het bij software jammer dat er meestal geen fysieke output is en eigenlijk alleen maar leeft op je scherm. Naast dat hoop ik dat ik hier op de Fontys een leuke stage kan vinden en dan natuurlijk slagen.

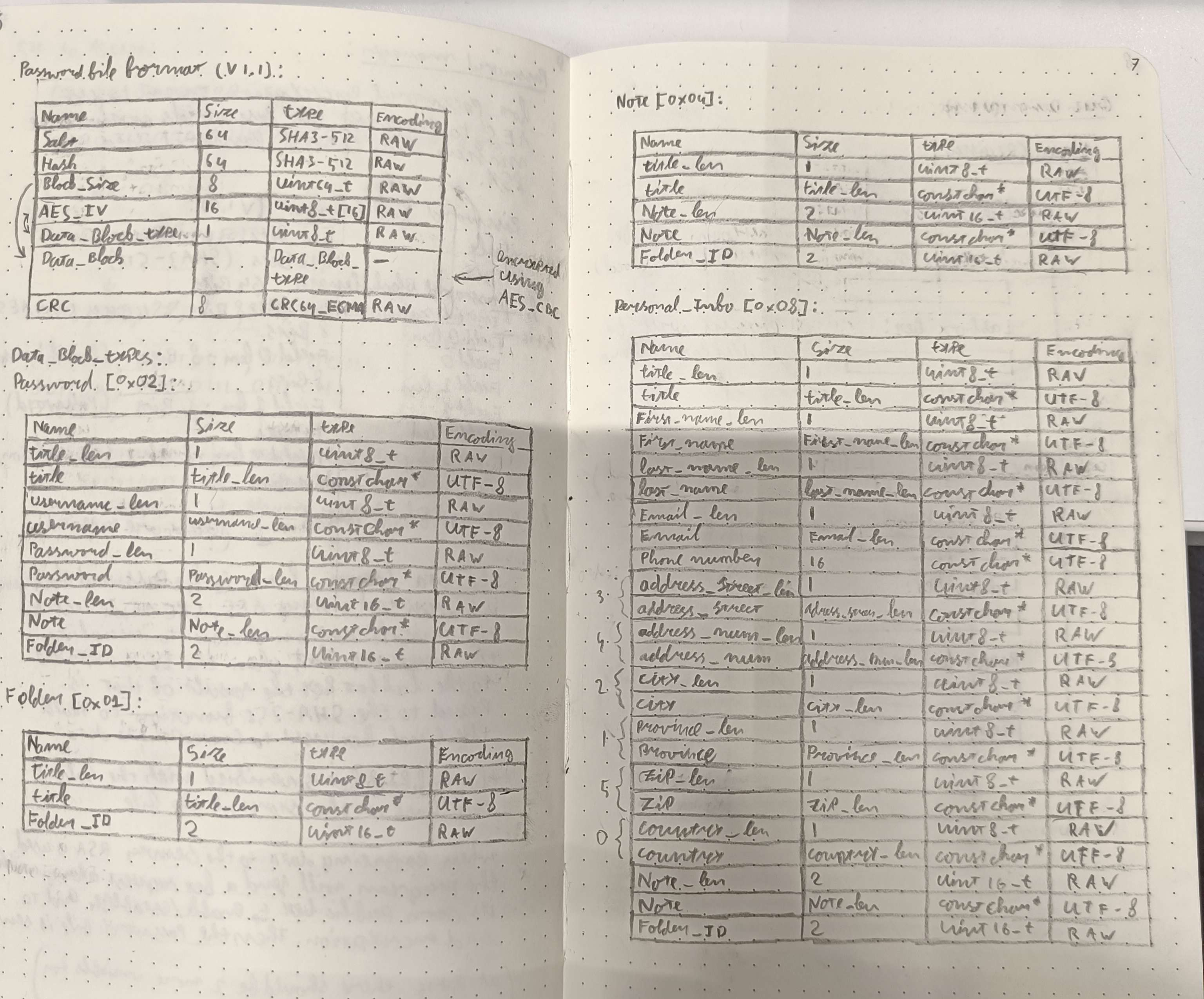
# Aantonen leerdoelen

## Idee

Deze password manager gaat passwords veilig op de pc zelf opslaan. Dit word gedaan met de volgende algoritmes / technieken:

* SHA-3 512
* AES CBC
* CRC64 ECMA

Hierbij hoort een file format:



Figuur file format

De file start met een salt en hash om het decrypt wachtwoord te checken en word verder opgedeeld in blocks die starten met: block\_size, IV (encryption init vector), block\_type, block\_data en ten slot een crc checksum. Alleen de data in block\_data word geencrypt de data die er in zit ligt aan block\_type.

## code

Text

Description automatically generated

Figuur getpass function @io.cpp

In deze functie word een input van de console opgenomen zonder dat die voor de gebruiker leesbaar is (zie figuur 2)

Leerdoelen: arrays, variables, datatypes, loops, conditionals, leesbaarheid



Figuur getpass output

Graphical user interface

Description automatically generated

Figuur bit\_reverse functions @math.cpp

Deze functies draaien de volgorde van de bits in een nummer om. (Dit word gebruikt bij de CRC algoritmes)

0100 1100 1011 1001 -> 1001 1101 0011 0010

Leerdoelen: datatypes, bitwise operations

A screenshot of a computer

Description automatically generated with medium confidence

Figuur Hash abstract class @hash.hpp

Deze class geeft aan welke functies een afgeleide (inheritance: <https://www.geeksforgeeks.org/inheritance-in-c/>) van deze class moet hebben. Dit is handig omdat de computer elke afgeleide class op de zelfde manier kan behandelen omdat deze dezelfde methods hebben.

Vincent: “ik snap de code niet maar ik snap wel wat het moet doen”, “ik weet niet hoe je het anders kan uitleggen”

Maddox: “ik snap classes niet”

Leerdoelen: classes, methods, inheritance, datatypes

Text

Description automatically generated

Figuur SHA256 class @hash.hpp

Deze class is afgeleid van de Hash abstract class en heeft dus alle verplichte methods (met override). Deze class maakt van data (strings of arrays) een hash (256 bits) deze hash is altijd hetzelfde met dezelfde input en kan niet gemakkelijk teruggedraaid worden.

Leerdoelen: classes, methods, inheritance, arrays, variables, datatypes

## Keuzes:

Waarom python en c++?

C++:

* Sneller
* Manual memory management en pointers
* Beter voor bit operations.
* Structures en andere vormen van data inpakken.
* Meer controle over dingen zoals de console (console buffer manipulation)

Python:

* Beter voor File manipulatie
* Beter voor het maken van GUI’s
* Veel predifined functions
* Garbage collection (automatic memory managment)
* Dynamically typed
* Cross platform (interpreted)

Waarom kies ik voor deze file structure?

A picture containing text

Description automatically generated

Figuur Library file structre

In de directory lib/compile/src/ zitten alle source files van de library. Deze source files zijn opgesplitst in verschillende groepen deze zijn: IO (input output), math, hash, encryption, check. Bind.cpp is een bijzondere file want in deze file word aan de compiler uitgelegd hoe de python module er uit moet zien. Al deze files worden gecompiled door een compile scrypt: lib/compile/compile.py deze maakt gebruik van lib/compile/CMakeLists.txt en output naar lib/compile/build de resulterende python binary (\*.pyd) word dan gekopieerd naar lib/ hier kan deze afgelezen worden door\_\_init\_\_.py waardoor deze leesbaar is voor andere python files die lib importen.

## Problemen

### File structure

voor dat ik deze file structure had (split files + split definition) had ik een source file met alle functions dit was het geval omdat mijn pre compiler (swig) geen meerdere files toeliet. Nu gebruik ik pybind11 dit is net als swig een C++ to C++ precompiler die mijn code compatible maakt met python. Met pybind11 kan ik wel split files en split definition toepassen wat veel overzichtelijker is.

### CRC

In mijn implementatie van de CRC algoritmes worden eerst lookup tables gemaakt en vervolgens gebruikt om de CRC uit te rekenen. Al deze lookup tables kloppen maar sommige CRC algoritmes kloppen niet omdat deze op een andere manier uitgerekend moeten worden. Ik heb alleen nog geen oplossing kunnen vinden. Gelukkig kloppen alle belangrijke algoritmes wel.

## Bronnen:

* <https://youtube.com/watch?v=_5T70cAXDJ0>
* <https://github.com/ETLCPP/crc-table-generator>
* <https://github.com/SergeyBel/AES>
* <https://github.com/stbrumme/hash-library>
* <https://github.com/hoffstadt/DearPyGui>

# Reflectie / evaluatie

## Waar ben ik trots op?

*Dat ik een goeie simpele library heb kunnen maken voor vershillende python projecten*

## Wat doe ik een volgende keer anders?

*Meteen pybind11 gebruiken, en alleen algoritmes toevoegen die compatible zijn (sommige crc algoritmes zijn (nog) niet compatible met mijn methode)*

## Welke formatieve indicatie zou ik mezelf geven voor de oriëntatie Software?

|  |  |  |
| --- | --- | --- |
| Leeruitkomst oriëntatie Software | | |
| Onderdeel | Criterium | Rating |
| Aandacht voor algoritmiek | Je kunt eenvoudige applicaties schrijven die stapsgewijs oplossingen voor problemen vinden door het uitvoeren van logische testen en eenvoudige stapsgewijze berekeningen. | *G, ik heb een algoritme library gemaakt voor mijn project* |
| Basisvaardigheden | Je begrijpt en past de volgende programmeerconcepten toe:Variables, conditional statements, loops, methods, lists/ arrays en enum’s. | G, al deze en meer concepten zitten in mijn library |
| Aantonen | Je vraagt feedback van een docent en laat zien dat je deze feedback verwerkt hebt. | G, ik heb bijna alle feedback in mijn project verwerkt |

## Welk verdiepend(e) profiel(en) kies ik en waarom?

*Tech omdat ik het leuk vind om echt iets fysieks te maken.*