|  |
| --- |
| Name:  Ratnayake K Basura N B |
| Student Reference Number:  10541973 |



|  |  |  |
| --- | --- | --- |
| Module Code: **SOFT336SL** | Module Name: **Cross Platform Application Development in C++** | |
| Coursework Title: Assignment - Technical Documentation | | |
| Deadline Date: **23/12/2015** | | Member of staff responsible for coursework:  Dr. Davide Marocco |
| |  |  | | --- | --- | | Programme:  **4872/3** |  | |  |  | | | |
| Please note that University Academic Regulations are available under Rules and Regulations on the University website [www.plymouth.ac.uk/studenthandbook](http://www.plymouth.ac.uk/studenthandbook). | | |
| Group work: please list all names of all participants formally associated with this work and state whether the work was undertaken alone or as part of a team. Please note you may be required to identify individual responsibility for component parts.  ***We confirm that we have read and understood the Plymouth University regulations relating to Assessment Offences and that we are aware of the possible penalties for any breach of these regulations. We confirm that this is the independent work of the group.***  Signed on behalf of the group: | | |
| Individual assignment: ***I confirm that I have read and understood the Plymouth University regulations relating to Assessment Offences and that I am aware of the possible penalties for any breach of these regulations. I confirm that this is my own independent work.***  Signed: | | |
| Use of translation software: failure to declare that translation software or a similar writing aid has been used will be treated as an assessment offence.  I \*have used/not used translation software.  If used, please state name of software………………………………………………………………… | | |
| **Overall mark \_\_\_\_\_% Assessors Initials \_\_\_\_\_\_ Date\_\_\_\_\_\_\_\_\_** | | |



**Degree:** BSc (Honours) Software Engineering

**Stage:** 3

**Batch:** 14.2

**THE**

**FORE RUNNERS**

Believe everything is possible

**INDEX NUMBER FULL NAME**

10541973 Basura Ratnayake

Rahasa 1.0 – Customizable Encryption

Technical Documentation

Table of Contents

[1. Home Screen 1](#_Toc470291618)

[2. Encrypt Screen 2](#_Toc470291619)

[3. Decrypt Screen 3](#_Toc470291620)

[Hierarchical Index 4](#_Toc470291621)

[Class Hierarchy 4](#_Toc470291622)

[Class Index 4](#_Toc470291623)

[Class Documentation 5](#_Toc470291624)

[Core Class Reference 5](#_Toc470291625)

[DecryptForm Class Reference 5](#_Toc470291626)

[EncryptForm Class Reference 6](#_Toc470291627)

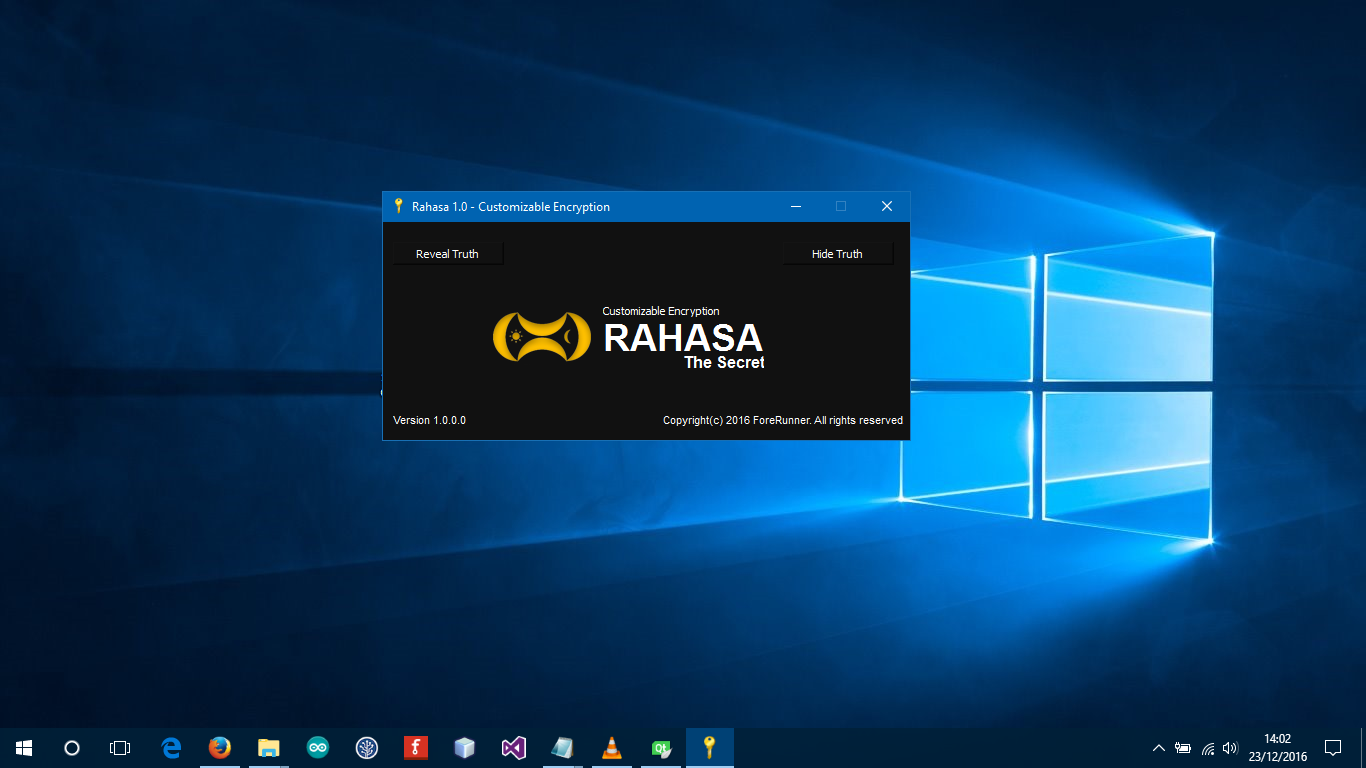
[Encryption Class Reference 7](#_Toc470291628)

**Description of the GUI Design**

The scenario for the project is an encryption and decryption system for text files. The system uses QWidget forms for its design. The project has been built to support on multiple operating systems using the Qt Creator IDE.

The main functionalities of the system, along with relevant screenshots are-

## Home Screen

From the home screen you can choose to **Encrypt** or **Decrypt** a file. All you need to do is click on the buttons on the top of the window.

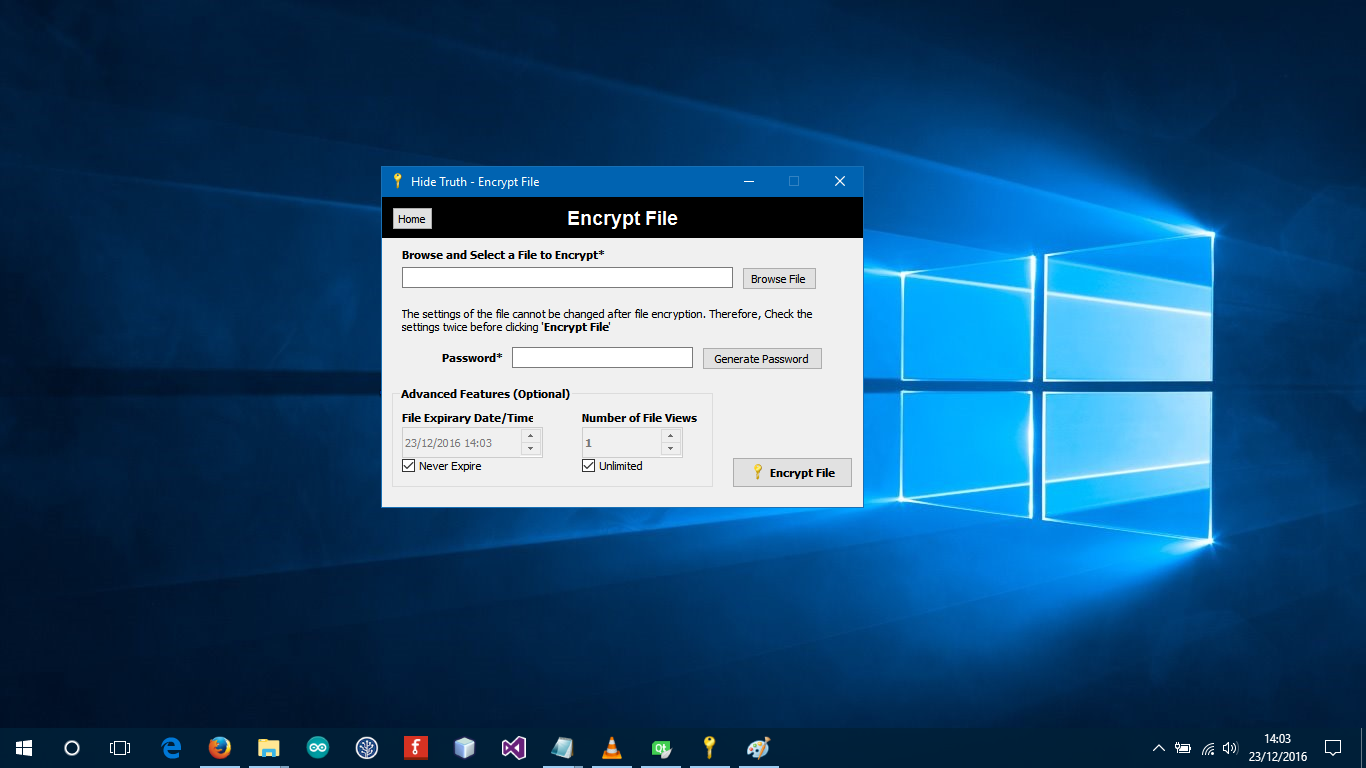
* To Encrypt: Click **‘Hide Truth’**

Opens the encryption window to browse file to encrypt.

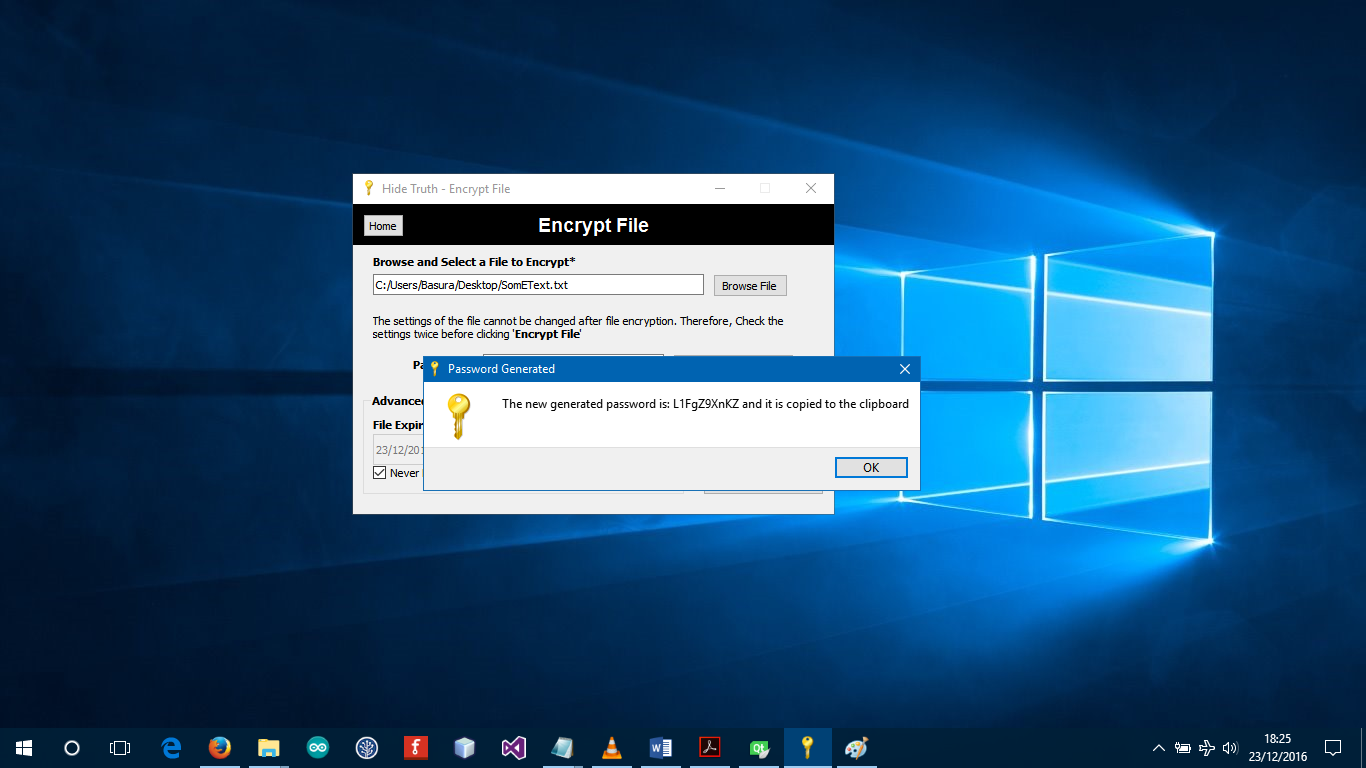
* To Decrypt: Click **‘Reveal Truth’**

Opens the decryption window to browse file to decrypt.

## Encrypt Screen



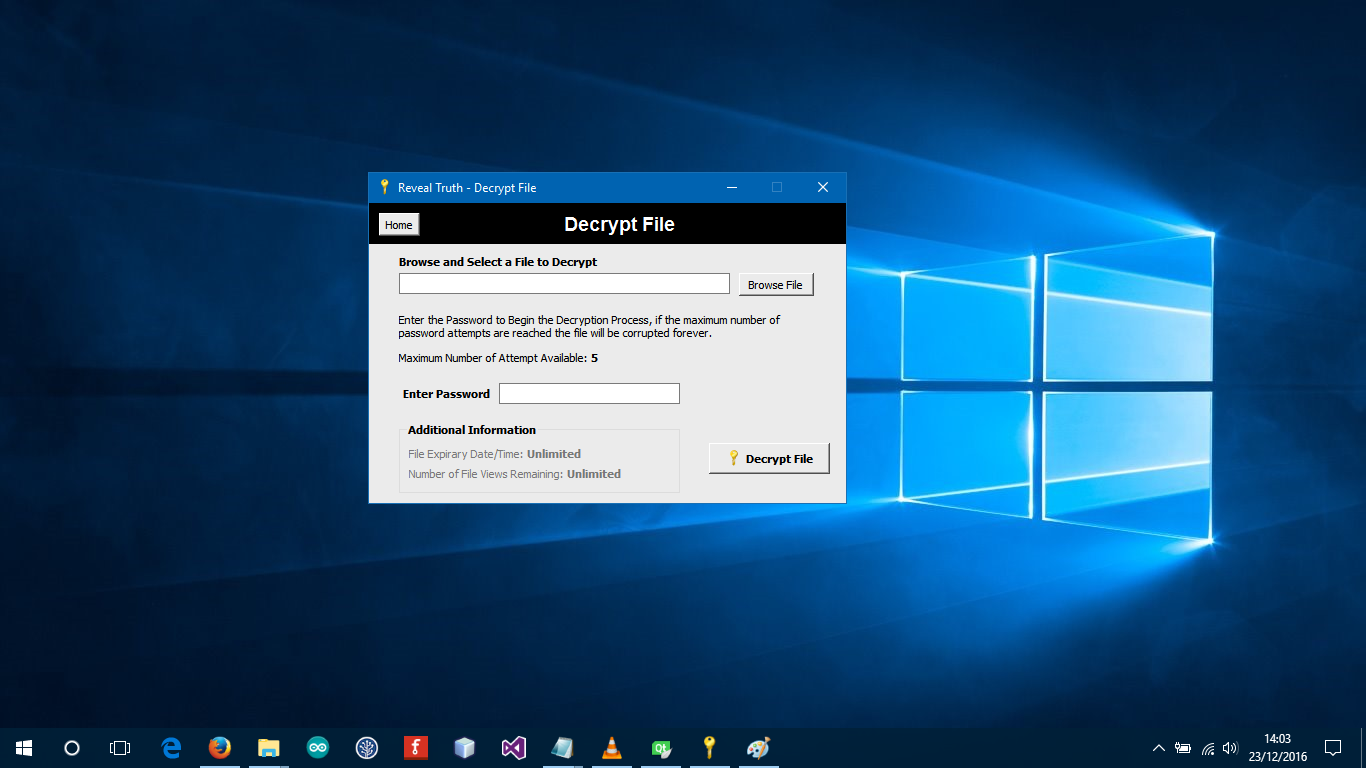
* **Browse File**

Select only text files from the dialog. Note: It is possible to encrypt any file type but cannot display video, audio and other file types on our Viewer for now.

* **Generate Random Password**

You can enter a password yourself or generate a random password by clicking ‘**Generate’**. Once generated the password is copied to your clipboard.

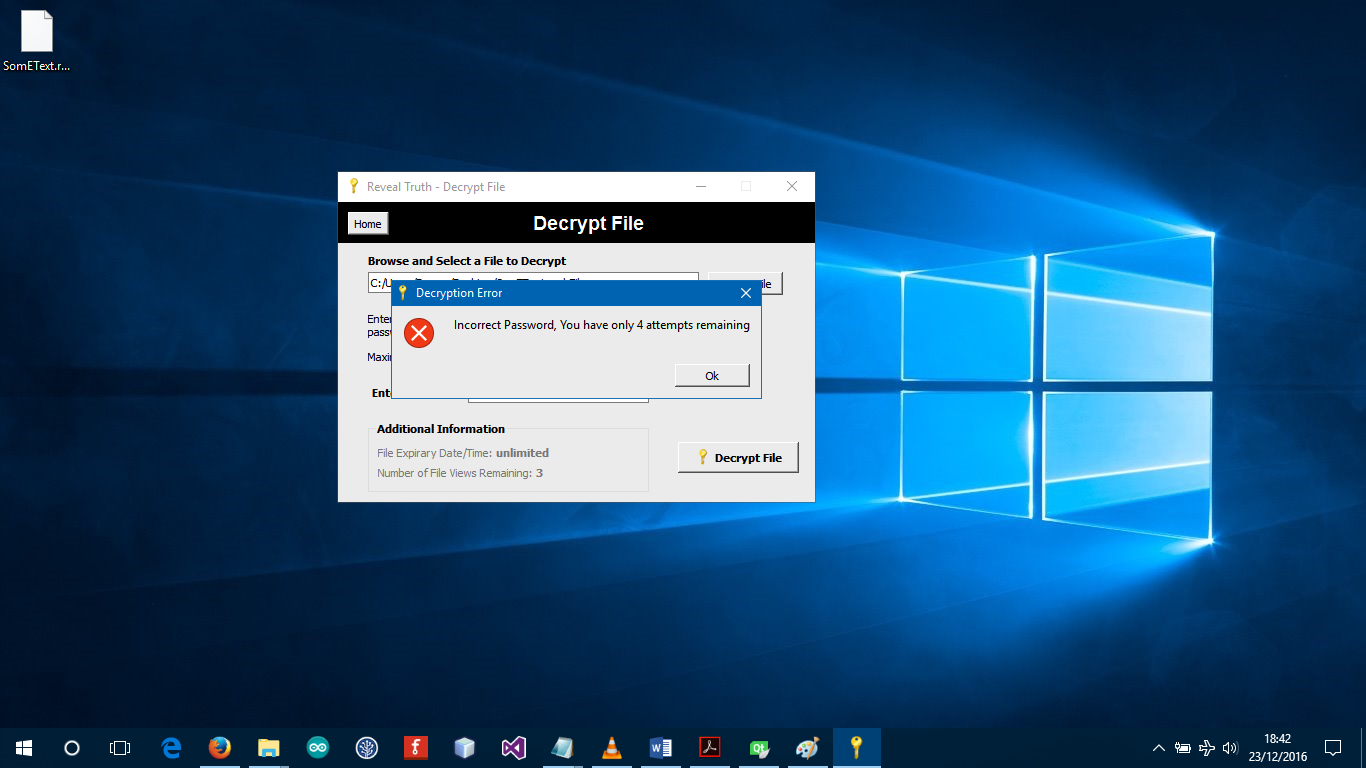
## Decrypt Screen



* **Browse the encrypted file**

Enter the password and click **‘Decrypt File’.**

**Note:** If file view limit or duration is exceeded the file be automatically corrupted.



* **Incorrect Password**

If the maximum password attempt limit is exceeded, the file will be corrupted.

**Documentation of the Code**

The application relies mainly on two core classes **“Core”** and **“Encryption”.** Core handles the GUI behaviour and the operation of files (read and write), the encryption class is inherited by the core class to enhance performance. There are three user forms in total. Additionally, a Core class is inherited to validate file inputs and generated keys.

Below mentioned details are well formed using Doxygen .

## Hierarchical Index

### Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

* **Encryption** 
  + Core

* **QWidget** 
  + DecryptForm
  + EncryptForm
  + StartUp

## Class Index

**Class List**

Here are the classes, structs, unions and interfaces with brief descriptions:

* **Core**
* **DecryptForm**
* **EncryptForm**
* **Encryption**
* **QWidget**
* **StartUp**

## Class Documentation

### Core Class Reference

Inheritance diagram and public functions.

**Public Member Functions**

* Core ()
* void showForm (QWidget \*, QWidget \*)
* void showFiledialog (QWidget \*, QString)
* bool isFileAvailable ()
* bool isFileAvailable (QWidget \*, QLineEdit \*)
* void copyToClipboard (QString)
* void zip (QString, QString)
* void unZip (QString, QString)
* QString getFilename ()

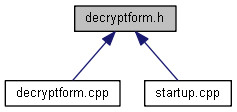
### DecryptForm Class Reference

Inheritance diagram and public functions.

**Public Member Functions**

* DecryptForm (QWidget \*parent=0)
* ~DecryptForm ()

**Protected Attributes**

* QStringList dataLine
* QString fileEx
* QString createD
* QString userKey
* QString prime1
* QString prime2
* QString expire
* QString maxView
* QString viewC
* QString lastView

### EncryptForm Class Reference

Inheritance diagram and public functions.

**Public Member Functions**

* EncryptForm (QWidget \*parent=0)
* ~EncryptForm ()

### Encryption Class Reference

Inheritance diagram and public functions.

**Public Member Functions**

* Encryption ()
* void setPassword (QString)
* void generateKeys (int, int)
* QString generatePassword (int)
* QString getPassword ()
* QString encrypt (QString)
* QString decrypt (QString)
* int getPrime1 ()
* int getPrime2 ()
* int getSysPrime1 ()
* int getSysPrime2 ()