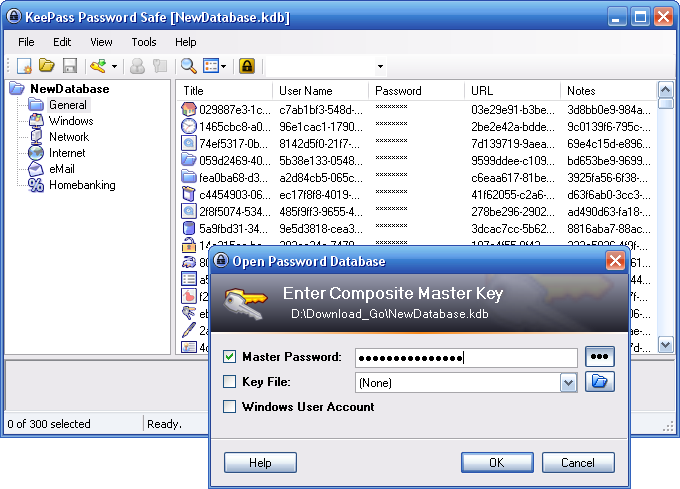
Computing Coursework

## Alex Burton



Exam No : 2018

Contents

[Alex Burton 1](#_Toc516645915)

[Analysis 4](#_Toc516645916)

[Introduction 4](#_Toc516645917)

[Initial assessment of suitability for computational methods 4](#_Toc516645918)

[Stakeholders 4](#_Toc516645919)

[Young People 5](#_Toc516645920)

[Senior Citizens 6](#_Toc516645921)

[Stakeholder Requirements 6](#_Toc516645922)

[Research of existing solutions 7](#_Toc516645923)

[1. KeePass 1.35 7](#_Toc516645924)

[2. Password Generator Plus Beta 11](#_Toc516645925)

[3. The Password Meter 13](#_Toc516645926)

[Essential Features 14](#_Toc516645927)

[Limitations of Proposed Solution 15](#_Toc516645928)

[System Requirements 15](#_Toc516645929)

[Minimum Hardware and Software Requirements 15](#_Toc516645930)

[Justification 15](#_Toc516645931)

[Requirements Specification 15](#_Toc516645932)

[Measurable Success Criteria 16](#_Toc516645933)

[Be stable 16](#_Toc516645934)

[Save and load files accurately 16](#_Toc516645935)

[Respond promptly to user input 17](#_Toc516645936)

[Have a concise and intuitive interface 17](#_Toc516645937)

[Encrypt and decrypt files accurately 17](#_Toc516645938)

[Include a built-in password generation tool 17](#_Toc516645939)

[Use data validation 17](#_Toc516645940)

[Use Tooltips 17](#_Toc516645941)

[Include an Entry Management Form 17](#_Toc516645942)

[Include a Basic Entry Management Dropdown List 17](#_Toc516645943)

[Record Searching 17](#_Toc516645944)

[Record Grouping 17](#_Toc516645945)

[Master Key Management 18](#_Toc516645946)

[User-Scheduled Maintenance Reminders 18](#_Toc516645947)

[Lock/Unlock Workspace 18](#_Toc516645948)

[Clear Clipboard 18](#_Toc516645949)

[Compression 18](#_Toc516645950)

[Tool Icons 18](#_Toc516645951)

[Design 18](#_Toc516645952)

[Breaking down the application 18](#_Toc516645953)

# Analysis

## Introduction

For my task I will create a password manager application. My intention is that my application will be useful and intuitive. It will take inspiration from other well-known password managers by including the cornerstone features found in them.

Unfortunately, I believe that other password managers can have a steep learning curve for those that need them the most. Both the young and elderly may struggle with maintaining a list of unique and secure passwords. This application could teach transferable skills about the use of password managers. I think that it is important that the application can teach the user in a short amount of time in a memorable way.

The primary purpose of the application will be to educate users about password managers which would allow users to move on to more complex and expensive password managers. From my application, I would like users to learn:

* The general layout and use of password managers
* Methods of password generation
* Experimental features

## Initial assessment of suitability for computational methods

|  |  |
| --- | --- |
| Feature | Justification of Suitability |
| User Interface | Allows complex tasks to be accessed by intuitive graphical representations. This is important to group different tasks under the same category. This is also important to reduce the amount of learning the user must do by logically dividing actions and features in the interface, rather than a list. |
| File Handing | All of the data about passwords and how they are managed must be secure during and after execution of the application. This will require complex encryption and decryption techniques. |
| Encryption/Decryption | All of the encryption and decryption techniques will be implemented by functions, some of which will be imported from libraries. This is because all of the functions will need the parsed string input and produce the output. The functions themselves are complicated mathematical algorithms which would be either impossible or incredibly time consuming to do by hand. |
| Subroutines/Functions | Many tasks will have overlapping sub-algorithms which could require previous inputs and outputs. File handling is likely to be the initial or final routine for some algorithms, and one feature could represent different actions under different circumstances. |
| Password Generation Algorithms | There are many simple and mathematically secure algorithms, some of which I would be able to implement myself, others through other people’s libraries. Information could be provided on methods of password generation through tooltips or further web links, where after creating the password the user only needs to input this into a database. |
| Teaching Tools | Some users will have a learning preference like explanation rather than intuition, so easily accessible information about a task or tool is necessary to prevent frustration to different types of users. An application would be able to cater to both types of user. |

## Stakeholders

For my application, I have two main groups of shareholders, young people and old people. This is because they both share issues with password management, but also have interesting differences.

### Young People

Young people are avid users of social media websites, which all have individual accounts which must be managed. This group tends to use identical or similar passwords over multiple websites. This could be because they find it inconvenient to remember many strong and unique passwords.

#### Persona Attributes:

* Generally dislike wasting time on slow processes at the expense of almost anything else
* Generally dislike overly complex and unintuitive applications
* Generally enjoy efficient useful learning
* Generally prefer to learn by doing rather than watching

#### Example:

Name: Ethan, Age: 17



Occupation: Works as a student in an academy. He struggles with memory issues, forgetting where he left his house keys or what his next lesson is. He is the only son in his family, and he aspires for a job.

Likes: Handouts, in class

Dislikes: Remembering mundane facts, such as whether or not he has P.E in the afternoon or not.

Typical day: Ethan walks to school, attends classes sometimes and records homework in his diary if he has it. At the end of the day, or earlier, he walks home and plays games until his bed time. Sometimes he needs to remember his password himself for the school computers. He typically has most of his passwords automatically saved for him on his home desktop system.

Real User: Ethan Connolly

#### How will they make use of my proposed solution?

|  |  |  |
| --- | --- | --- |
| Need | Solution | Why the solution is appropriate to my stakeholders. |
| Secure passwords | One or more master keys which would be entered to temporarily decrypt a set of encrypted passwords. | This would prevent password theft and further account breaches. |
| Easy access to multiple secure passwords | Each field of a record would be selectable and could be copied from. This includes passwords. | This will allow the user to have to remember significantly less amount of information. |
| Good password management habits | A table of passwords could contain identifiable information for each password, such as the website it is used for. Each password could be generated using any one of many techniques explained or performed in the program, resulting in heightened cyber-security. | This would set good habits for when the cost of having an insecure password is greater. |
| Teaching about how to use a password manager | The implementation of tooltips would provide on demand information about how to use specific tools. | This would help by reminding the user how to use a tool, or introducing how to use a specific tool. |

### Senior Citizens

Older people may struggle with memory issues due to old age, or from not using or doing particular tasks for a long time. They may also have outdated methods of doing tasks, which may not be completely wrong, but are suboptimal now.

#### Persona Attributes:

* May not find modern ‘intuitive’ features as intuitive as younger people
* Generally prefer clear and concise explanations over learning by doing
* May have sickness or disabilities which may hinder use of poorly designed interfaces

#### Example:

Name: Harold, Age: 54



Occupation: Retired, but he sometimes invests in stocks and gambles.

Likes: Being organised, kittens and spending his time wisely.

Dislikes: Disorder and chaos.

Typical day: Harold usually sleeps until midday, then has his breakfast at 2 o’clock in the afternoon. He regularly participates in high-stakes poker matches on and off the net. He prefers cashless payment systems, but is beginning to struggle with remembering his account details and passwords from using this method so often.

Real User: My Uncle

#### How will they make use of my proposed solution?

|  |  |  |
| --- | --- | --- |
| Need | Solution | Why the solution is appropriate to my stakeholders. |
| Easier access to multiple passwords | One or a set of master keys will be used to access multiple passwords which can be directly copied to the clipboard of the user. | This will allow the user to have to remember significantly less amount of information. |
| Teaching to learn about how to use password managers | Access to additional helpful information will be available throughout the application. | This would help the user to understand how to use the application through guides and further reading. |
| Secure password storage | The table will be encrypted when saved or locked, and decrypted when loaded or unlocked. | This would prevent password theft and further account breaches. |

Additionally, information and tools will be provided to generate secure passwords.

### Stakeholder Requirements

So that the needs of my stakeholders are met, my application must meet the following stakeholder requirements:

1. Records and password can be accessed quickly
2. Records have direct links to the relevant websites which can be clicked
3. Password strength has a visual indication for each record
4. Similar passwords cannot be used
5. If the master password is incorrectly entered too many times, a hint or security question can be provided

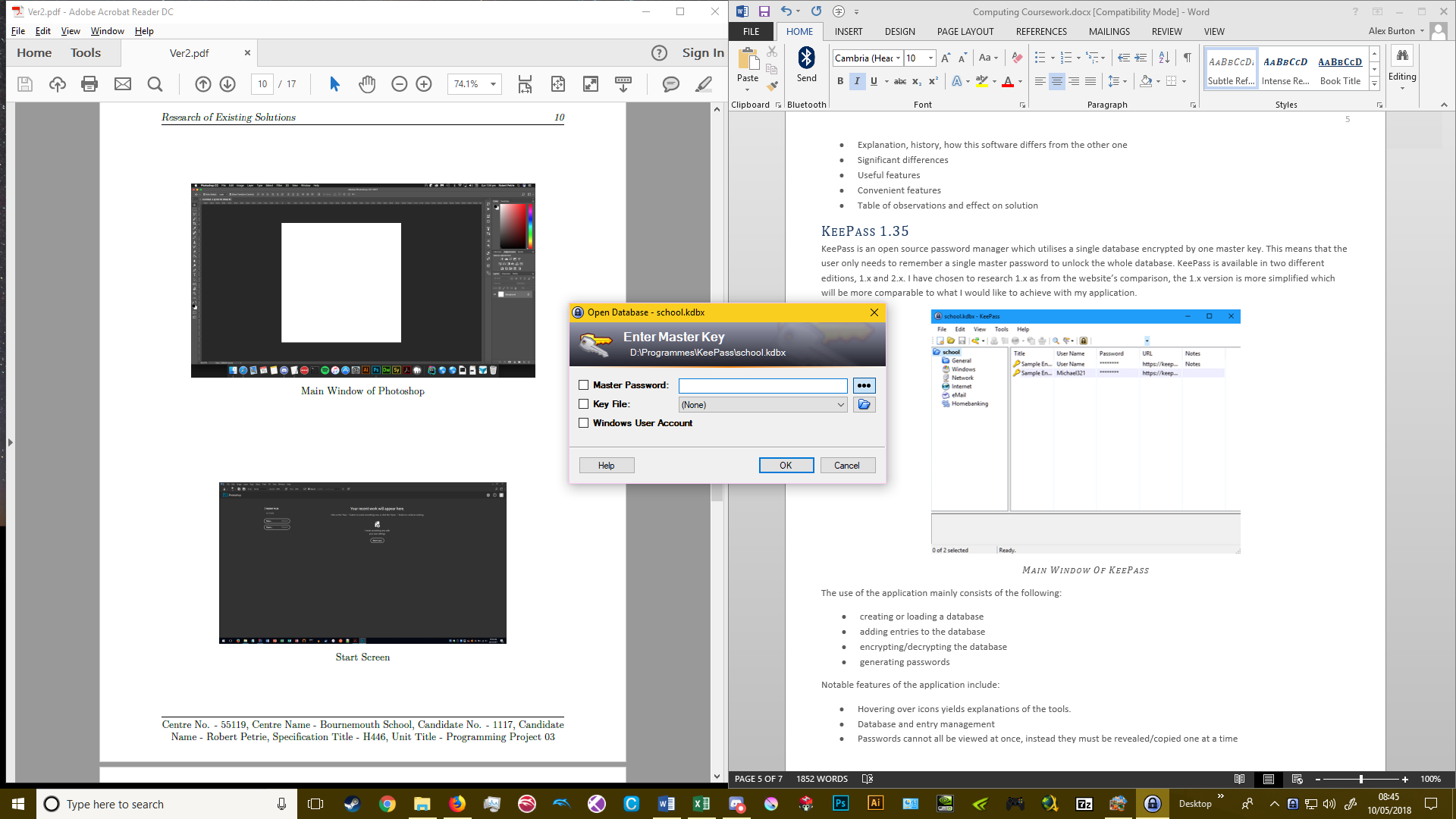
## Research of existing solutions

The solutions which I plan to research include a password manager, and one password generator, and a passwords strength detector. These solutions are:

1. KeePass 1.35
2. Password Generator PlusBeta
3. The Password Meter

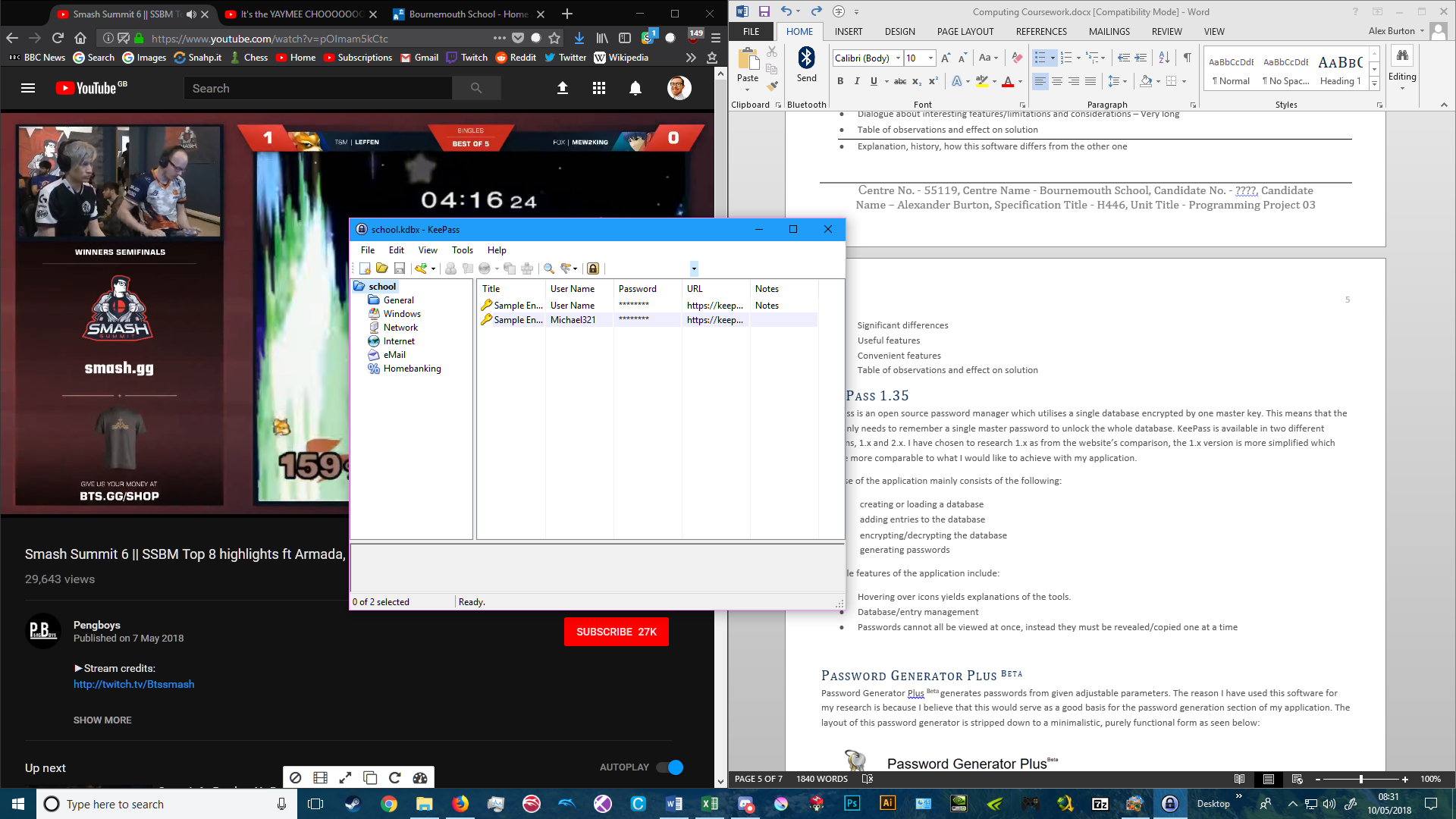
## KeePass 1.35

KeePass is an open source password manager which utilises a single database encrypted by one master key. This means that the user only needs to remember a single master password to unlock the whole database. KeePass is available in two different editions, 1.x and 2.x. I have chosen to research 1.x as from the website’s comparison, the 1.x version is more simplified which will be more comparable to what I would like to achieve with my application.



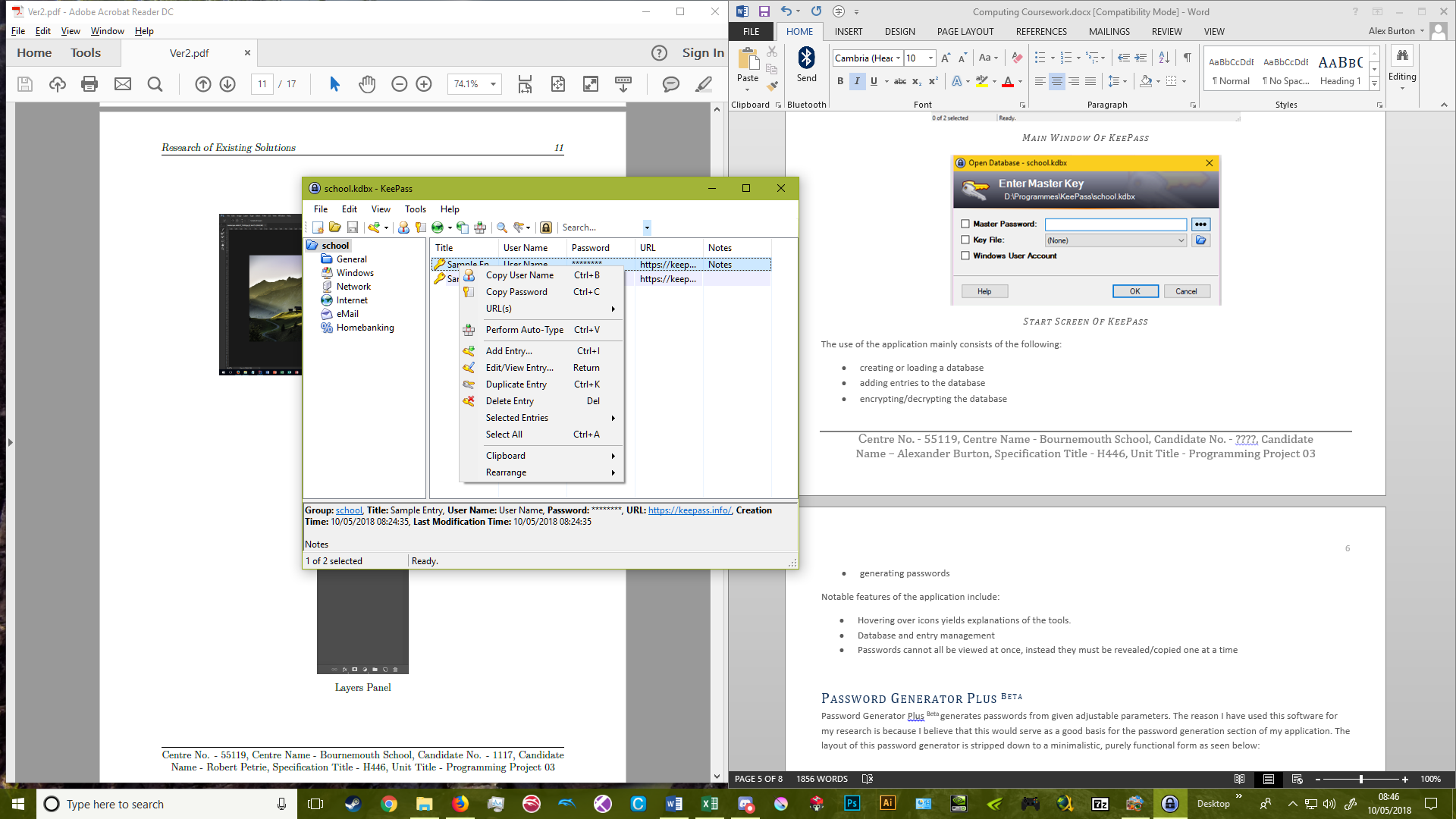
Start Screen

This is where the user loads and decrypts a database by entering a master password or loading a key file. The user also has the option of loading a different database. There are a couple of input options such as checkboxes, buttons and a textbox. The application makes use of labels, text and graphical icons to help the user distinguish the function of these input methods.



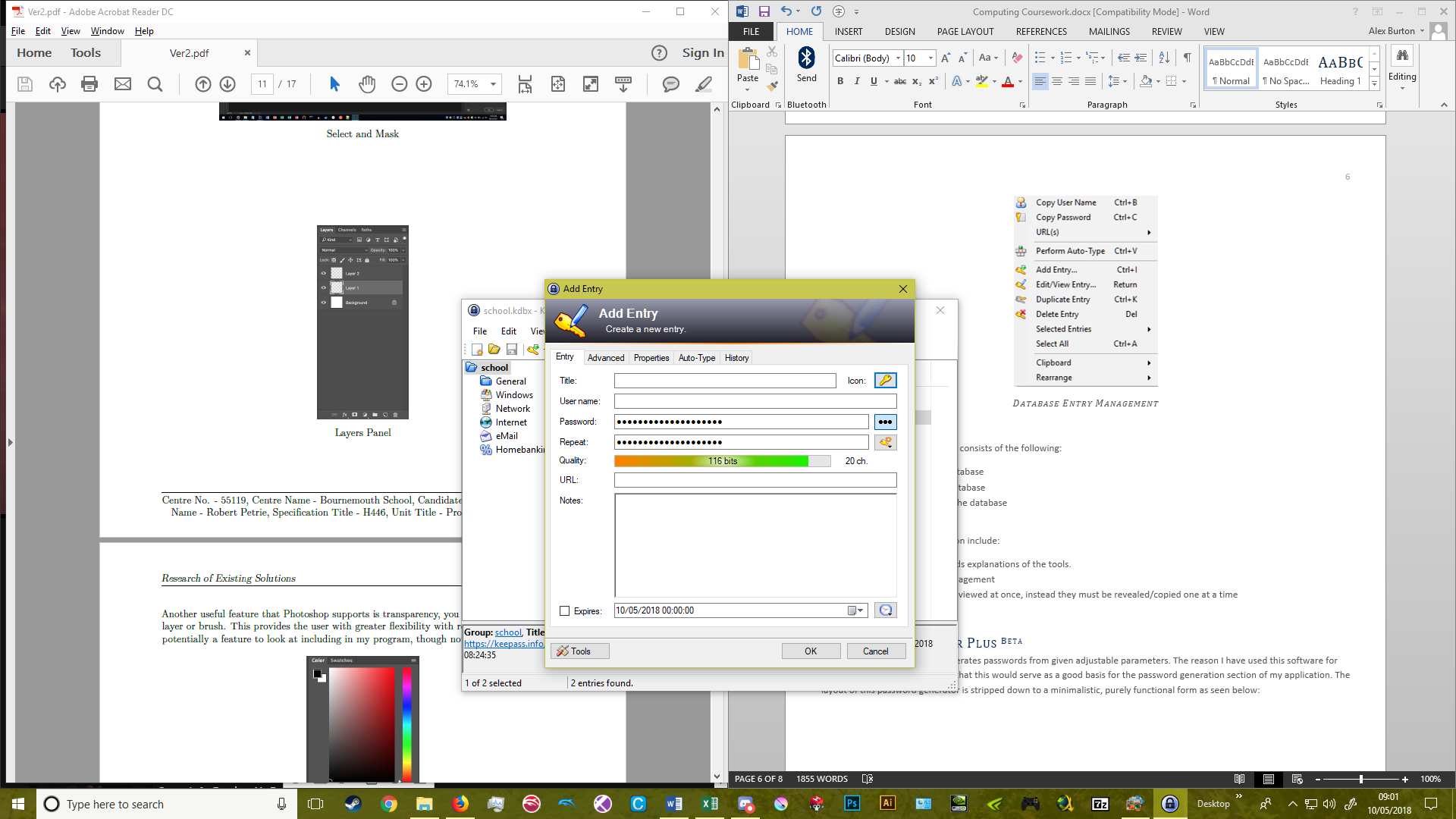
Main Window

This is where the user will have a range of management options and tools to choose from. If the user hovers over the icons, tooltips with helpful information appears.



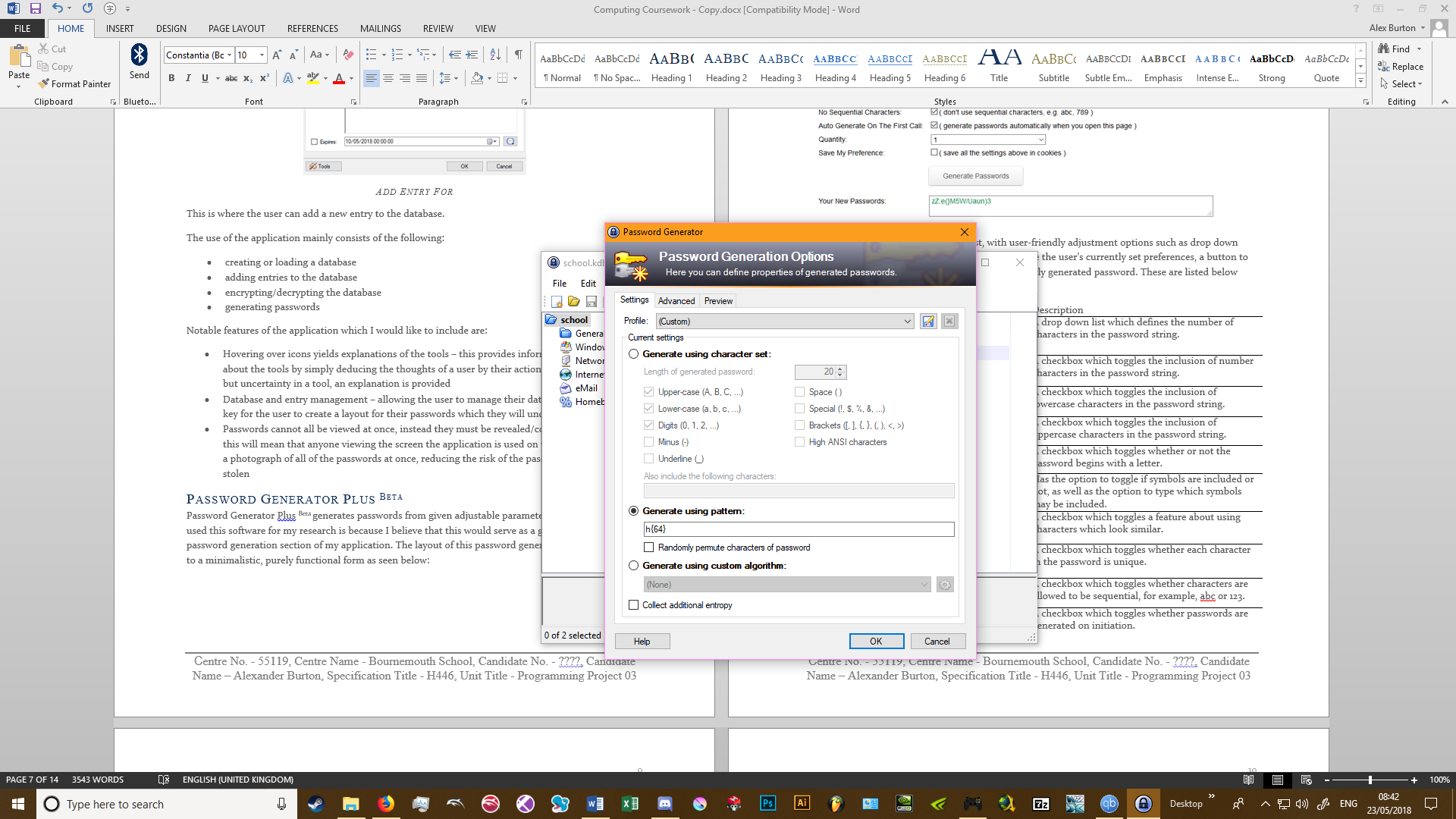
Database Entry Management

This is where the user can select an action to manage entries in the database. The typical actions for entry management are, copying data, adding, editing, or removing entries and selecting entries.



Add Entry Form

This is where the user can add a new entry to the database. The user has the option of manually entering data to fields which can then be later copied to the clipboard from the database. There is a visual representation of the ‘quality’ of the password, as well as the option for the user to make a note about the entry for themselves and an expiration date and time for the entry.



This is where the user can generate a password with specific parameters.

The use of the application mainly consists of the following:

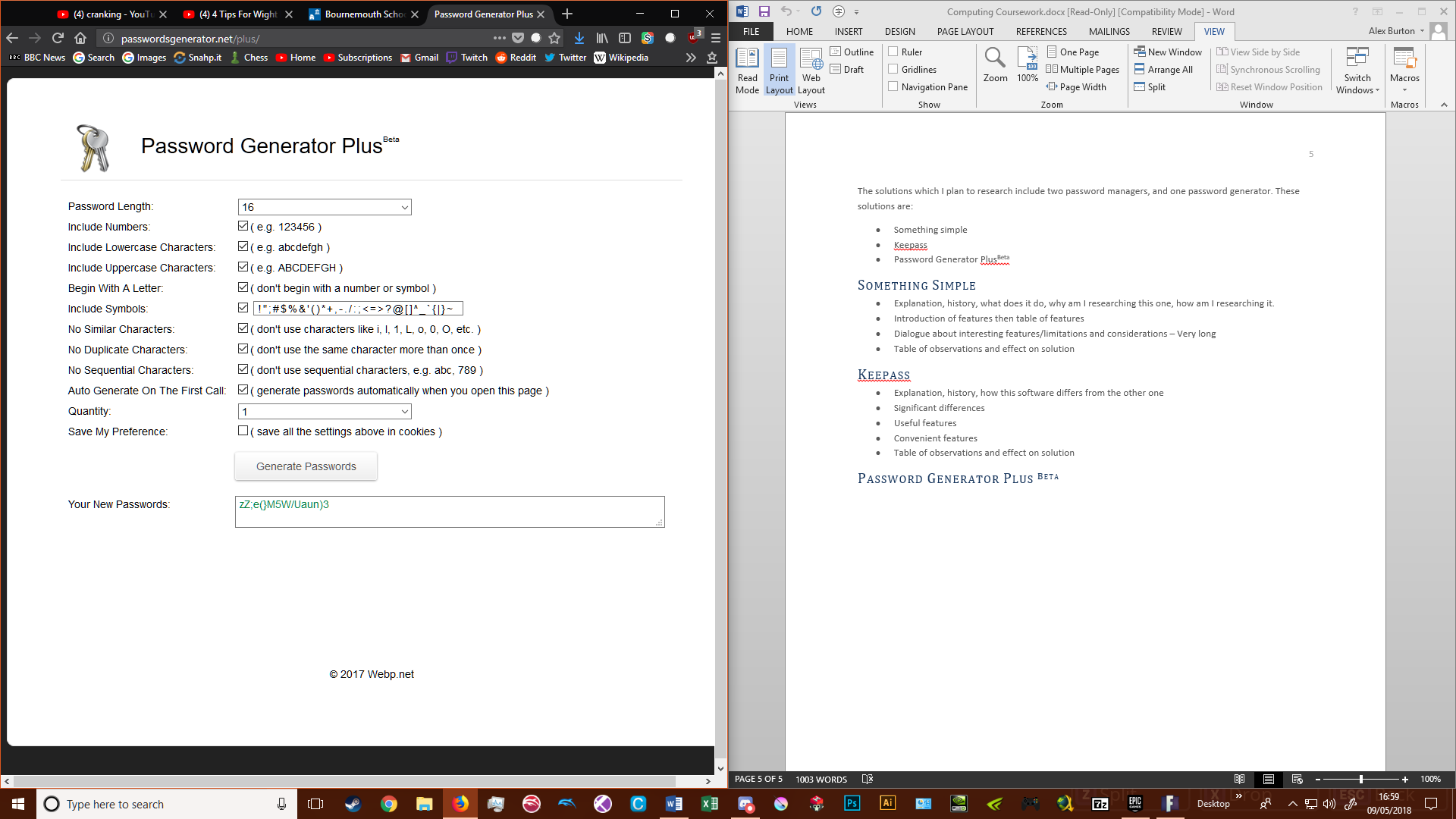
* creating or loading a database
* adding entries to the database
* encrypting/decrypting the database
* generating passwords

Notable features of the application which I will either include or are:

|  |  |  |
| --- | --- | --- |
| Feature/s | Inclusion | Justification |
| Hovering over icons yields explanations of the tools | Include | This provides information to the user about the tools by simply deducing the thoughts of a user by their actions, if they show interest but uncertainty in a tool, an explanation is provided. |
| Database and entry management | Include | Allowing the user to manage their database and entries is key for the user to create a layout for their passwords which they will understand. |
| Passwords cannot all be viewed at once, instead they must be revealed/copied one at a time | Include | This will mean that anyone viewing the screen the application is used on will not be able to take a photograph of all of the passwords at once, reducing the risk of the passwords being seen and stolen. |
| Searching through the database | Include | This allows the user to type in a string which is used to find a record containing a similar string in one of its fields. |
| Grouping records of the database | Include | This helps the user manage a large list of records by the user self-curating the groupings, this will reduce the time the user spends looking for particular records. |
| Sorting the database by field | Include | This will aid the user in finding records if they are able to remember where in a sorted list a record may come based on the sorting parameter. |
| Change the master key | Include | This is important so that the user can keep the database secure by giving them the option to change the master password in the case that it is compromised. |
| Scheduling | Include | The master password and records should have optional reminders to change the password at regular specified time intervals, this is important so that if a password is stolen, it is no longer useful after it is changed. |
| Lock Workspace | Include | This keeps the application open, but encrypts all of the files and brings back the start screen, this prevents data theft while the user is not at their desktop machine and prevents the user being inconvenienced. |
| Clear clipboard | Include | When the application is closed, or a user-specified amount of time has elapsed, the clip-board is cleared. This will prevent the user from accidently entering their decrypted password in to an insecure application if they use the paste function. |
| Compression | Include | As an educational application, it is unlikely that the user will use any substantial file sizes, however the application may be used on hardware with a low storage capacity such as a raspberry pi. Therefore it is important that my application does use compression. |
| Print | Exclude | I think that this is poor for password security as all of the passwords are in one physical place and without encryption, therefore this does not work towards my aim to promote good password security habits. |
| Import/Export | Exclude | There are a large number of formats that records would need to be accounted for, I believe that including this feature would be time consuming with a poor outcome, therefore I have chosen to exclude this feature. |

## Password Generator Plus Beta

Password Generator Plus Beta generates passwords from given adjustable parameters. The reason I have used this software for my research is because I believe that this would serve as a good basis for the password generation section of my application. The layout of this password generator is stripped down to a minimalistic, purely functional form as seen below:



As seen, the parameters are given in a list, with user-friendly adjustment options such as drop down lists tick boxes. Additionally, there is an option to save the user’s currently set preferences, a button to generate passwords, and a textbox containing the newly generated password. These are listed below with a description of their function:

|  |  |
| --- | --- |
| Feature | Description |
|  | A drop down list which defines the number of characters in the password string. |
|  | A checkbox which toggles the inclusion of number characters in the password string. |
|  | A checkbox which toggles the inclusion of lowercase characters in the password string. |
|  | A checkbox which toggles the inclusion of uppercase characters in the password string. |
|  | A checkbox which toggles whether or not the password begins with a letter. |
|  | Has the option to toggle if symbols are included or not, as well as the option to type which symbols may be included. |
|  | A checkbox which toggles a feature about using characters which look similar. |
|  | A checkbox which toggles whether each character in the password is unique. |
|  | A checkbox which toggles whether characters are allowed to be sequential, for example, abc or 123. |
|  | A checkbox which toggles whether passwords are generated on initiation. |
|  | A dropdown list which defines how many passwords will be generated. |
|  | A checkbox which toggles whether the preferences set are saved between uses. |
|  | A button which generates the passwords. |
|  | A textbox containing the string of the password or passwords generated. |
|  | This string is generated below the created password to provide a memory aid. |

From this research I had observed that this website tool was made with simple text, checkboxes, dropdown lists, and textboxes. These act as a form of data validation. I think it would be beneficial to include restricted parameters in my own application. This would allow the application to ensure that the user has a limited scope of choice so that the data entered is sensible for use within the program. Also, the use of colour representation is applicable to my application as this would aid the user by displaying choices in a visually intuitive manner, for example using a colour representation to represent security when selected password length.

I also observed that no key information was hidden in the program; I think that this improves the readability of the application to show as many options as possible rather than hiding them behind menus. I aim to make my application similarly clear, however, this should not come at the expense of cluttering the window. By doing this I will work towards making my application more intuitive but still powerful.

One issue I have is that the roles of the tools are not immediately apparent. In order to know what each checkbox and option corresponds to, the user must read the text about them. I believe that the application, although memory efficient, is poorly designed. For my application, I will include graphical representations to enhance the structure of my application by making information more effectively expressed.

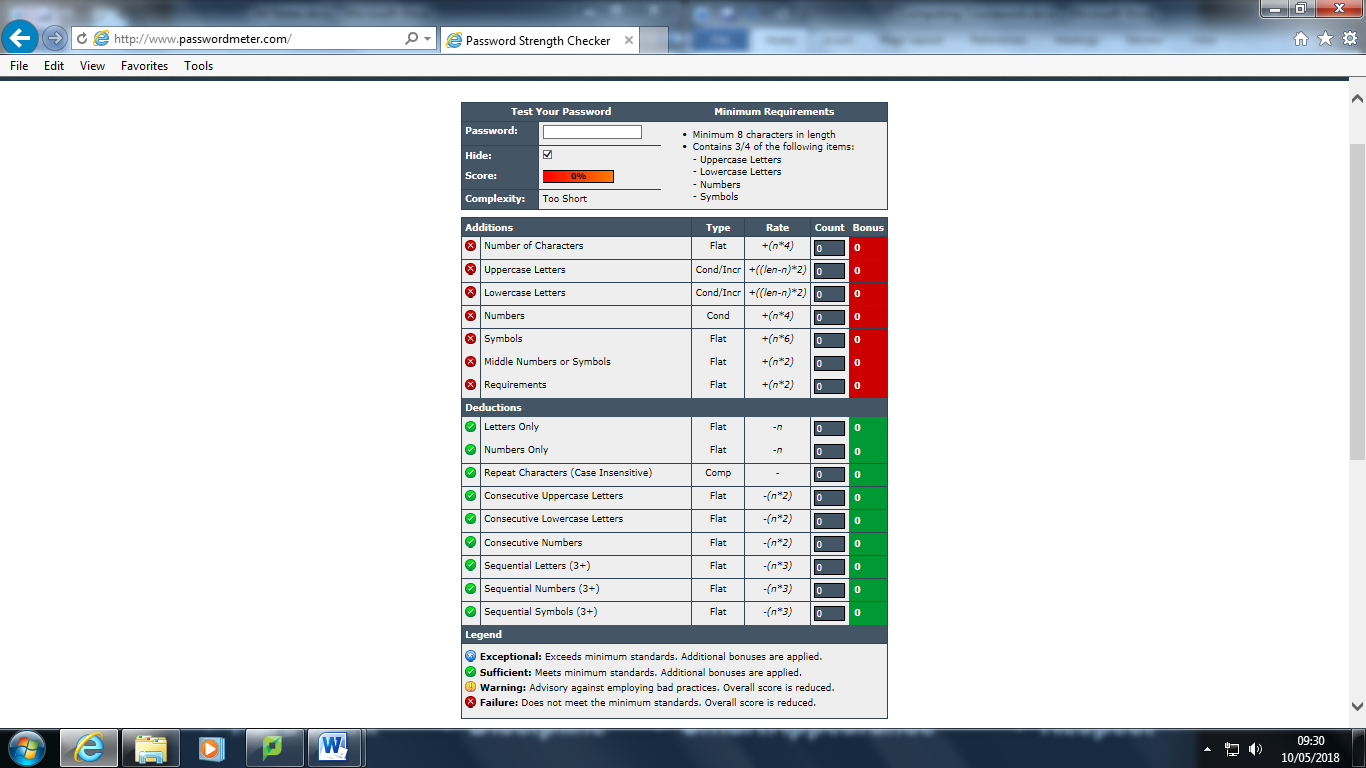
Another Issue I have is the string of the generated password being shown. I believe that the application should not show the user the password once they’ve generated it unless they choose to display it, as the password would be compromised if their screen is being watched.

In table format:

|  |  |  |
| --- | --- | --- |
| Feature/s | Inclusion | Justification |
| Restricted parameters | Include | This would allow the application to ensure that the user has a limited scope of choice so that the data entered is sensible for use within the program |
| Colour representation | Include | This would aid the user by displaying choices in a visually intuitive manner, for example using a colour representation to represent security when selected password length. |
| Concise interface | Include | This would help in making my application more intuitive but still powerful. |
| Graphical representations | Include | This would enhance the structure of my application by making information more effectively expressed. |
| Password String | Partially Exclude | I believe that the application should not show the user the password once they’ve generated it unless they choose to display it, as the password would be compromised if their screen is being watched. |

## The Password Meter

The Password Meter is a simple tool on a website used to enter passwords and determine their strength. At the top of the website, there is a textbox where the user can enter their, or a template password. Next to this is a strength meter, which could be used to determine the strength of the password. Below this is a table of additions and deductions, with columns about how that addition or deduction affects the score. This can be used to make alterations to the password to improve its strength. The purpose of the application is to provide instantaneous visual feedback to allow the user to improve their password strength by focusing on typical bad habits of creating passwords.



Start Layout

The Password Meter is short and to the point. Despite this, the method and options for generating the password are bare bones, and while the interface is informative, is unappealing and unsuitable for my stakeholders. I think that the principle of showing the user typical password creation mistakes, and having a transparent password strength system is a good idea which I will try and incorporate. The use of symbols to represent the suitability of the password for each category is also a strength which aids the principle I mentioned, therefore this would be good to incorporate also.

## Essential Features

From my research into similar software and my stakeholders, I have created a list of essential features which I would like to include in my application:

|  |  |  |
| --- | --- | --- |
| Feature | Description Of Feature | Explanation of Choice |
| A database to store the passwords | A table consisting of rows of records, each record has its own primary key and each field holds some data. | this allows records to have multiple data fields, which is important to identify how and where individual passwords are used, as well as any additional information about them |
| A database saving and loading feature | A set of routines that will save a database file or load a database file. | this is important for the database to remain identical between sessions of using the application, as well as in-case of an unforeseen crash or error |
| A tool with multiple methods to generate passwords | A set of password generation methods which will have optional parameters. | allows the user to have choice and information about generating passwords in order to make a stronger password, as well as a password which may be memorable |
| An integrated tool to determine and display the strength of passwords | A password strength meter an strength algorithm | gives the user a visual representation of their password strength so that they can make changes to their password to make it more secure |
| Data validation | An algorithm or group of algorithms which checks the type of data entered is correct. | ensures that the data entered by the user is sensible and will work within the application as expected, rather than cause a crash, as well as immediately providing feedback to the user about incorrect data entry |
| The most important tools should be immediately accessible | An interface layout scheme where any important tools can be seen when the application starts. | this improves the ability of the user to immediately recognise the purpose of tools at a glance, reducing the learning curve of the program |
| Database Encryption and Decryption | Functions which will take the database as an input, and will output a decrypted or encrypted version. | If the database is not able to be encrypted and decrypted accurately, data will be lost, the passwords will not function, and the data will not be stored securely. |
| Password Generation Functions/Explanations | One or more functions with password parameters to automatically generate passwords for the user. | This is important as the user may not be aware of better password generation methods and procedures. This would help them to generate more secure passwords. |
| Entry Management Form | A form which will be a more user friendly way to manage everything about entries of a database. | This is important so that the user has the freedom to manage all elements of an entry in one form. |
| Clear Clipboard | A simple algorithm that will clear the user’s clipboard. | This is to prevent malicious applications from looking at the user’s clipboard to obtain their password, as well as preventing the user from accidently pasting their password somewhere visible. |
| Restricted parameters | One or more algorithms that checks if the type of data entered is the correct type of data. | This is to prevent unexpected errors by ensuring the data entered is always of an expected type. This will also prevent crashes. |
| Records can be accessed quickly | The algorithm to access records is made more efficient, perhaps asynchronous, to load records faster. | This is to reduce password access time so that the user can access their accounts when they are short of time. |
| Records include a links/URLs field which can be clicked | A field is included which will contain clickable URLs and links to websites | This is to act as a convenience feature for the user to reduce the number of additional actions the user must perform. |
| Similar passwords cannot be used | An algorithm will compare a newly entered password to the other passwords in the database and determine whether or not a password is too similar to it. | This is to prevent emerging patterns of passwords so that it is more difficult to access the user’s accounts by hackers. |
| Security Question / Hint | If the master password is incorrectly entered too many times, a popup with a hint or security question will appear. | This helps the user when they cannot remember their password. |

## Limitations of Proposed Solution

I have decided to create ‘all-in-one’ package, with the functionality to create and store passwords. This means that I will be including at least all of the stripped down features of a regular password-manager, and my own twist on generating passwords. I will create a user-friendly interface, using both text and graphical techniques.

I will not attempt to create any complicated encryption features myself. Instead, I will probably use libraries for this because it would allow me to spend more time creating features to educate the user whilst having more secure encryption. If I have the time, I may write some code for simpler encryption techniques.

I may struggle with making all of the important tools immediately accessible, for example the password generator tools may be more suitable only when they are required rather than all of the time, as otherwise the interface may become quite cluttered.

## System Requirements

My application will consist of text and bitmap images as well as variable processing requirements dependent on the encryption and decryption functions used. However, this is unlikely to affect the minimum hardware required.

### Minimum Hardware and Software Requirements

|  |  |
| --- | --- |
| OS: | Windows® Vista/XP |
| Processor: | 1.0 GHz Processor |
| Memory: | 1 GB RAM |
| Storage: | 500 MB available space |
| Additional notes: | Mouse, Keyboard, Monitor |

### Justification

I believe the processing, storage and memory requirements will be fairly low, perhaps even lower than I have stated. If the user uses an excessively large database file, the application would require a larger amount of storage space and memory to hold, and would take a longer amount of time to process during the encryption and decryption processes, however, this would fall outside of the scope of intended use. Therefore, these requirements should be sufficient for the intent of education.

## Requirements Specification

For my solution to best fulfil the concept, a number of criteria will need to be met:

|  |  |
| --- | --- |
| Criteria | Justification |
| Be stable | So that data is not lost during saving, and so that the user is not confused or frustrated using the application. |
| Save and load files accurately | So that data can be stored and loaded in a reproducible way. |
| Respond promptly to user input | So that the user understands that their input has had an effect in the application. |
| Have a concise and intuitive interface | So that tools and information are easily distinguished, even by someone who is not an advanced user. |
| Encrypt and decrypt files accurately | So that data fidelity is kept to a high standard. |
| Include a built-in password generation tool | So that the user is able to generate secure passwords easily and offline. |
| Use data validation | To prevent bad data entry this could cause unexpected errors or crashes in the program. |
| Only show one decrypted password at a time | So that a screenshot, picture or quick look at the screen of the user will not compromise all of their data. |
| Encrypt and Decrypt the Database | So that the data will be stored securely, and can be retrieved only by the user. |
| Use Tooltips | So that the user has additional explanatory information if they require it. |
| Include an Entry Management Form | So that the user can manage all elements of an entry in one form. |
| Include a Basic Entry Management Dropdown List | So that the user can manage some elements of an entry without having multiple forms open. |
| Record Searching | So that an individual record is more easily located from a list of many. |
| Record Grouping | So that groups of similar types of records can be accessed by the user. |
| Master Key Management | So that the master key can be changed if the user thinks their key is too weak or compromised. |
| User-Scheduled Maintenance Reminders | So that the useful-lifetime of compromised passwords is lower. |
| Lock/Unlock Workspace | So that the user can encrypt the database quickly, without having to exit the application. |
| Clear Clipboard | So that the user does not accidently reveal their password. |
| Compression | So that larger databases take up less storage space, this is important if the user decides to use the application on portable storage media. |
| Tool Icons | So that the purpose of tools is more apparent from a glance. |
| Records and passwords can be accessed quickly | To make the interface more responsive for the user |
| Links and URLs can be clicked | To provide convenience for the user |
| Hint | To help the user when they forget their master password |

## Measurable Success Criteria

It is important that the above specifications are tested against the following criteria.

### Be stable

* The program does not crash on these conditions:
  + On start
  + On saving and loading
  + On general user interaction

### Save and load files accurately

* Application does not crash if the user enters an illegal file path
* Application does not crash if saving is cancelled before completion
* Files should be able to be saved and loaded so that all information held in the database is accurate after loading to immediately before saving

### Respond promptly to user input

* Actions either cause immediate changes, or generate a loading indicator

### Have a concise and intuitive interface

* Most tools should be visible at all times
* Tools should have self-explanatory graphical representations
* The interface should not be cluttered with too many tools or small icons

### Encrypt and decrypt files accurately

* Application does not crash when encrypting files
* Application does not crash when decrypting files
* Files should be able accurate after decryption to before they were encrypted

### Include a built-in password generation tool

* Have some form of ‘strength meter’ for the passwords
* Have multiple methods to generate passwords
* Have options about the characters used in the passwords

### Use data validation

* Use form tools where possible for suitable data types
* Ensure that the data entered into textboxes is suitable

Only show one decrypted password at a time

* The database uses asterisks instead of the normal characters for the password
* A decrypted password is shown on the data entry form

### Use Tooltips

* Every tool with an icon has an explanation that appears in a textbox when hovered over after a few seconds
* Every tool icon has an explanation that appears in a textbox when hovered over after a few seconds

### Include an Entry Management Form

* A form can be opened to manage every field of an individual entry in the database

### Include a Basic Entry Management Dropdown List

* A dropdown list can be accessed to manage an individual entry of a database by right clicking
* One of the dropdown options is to open the entry management form

### Record Searching

* A textbox exists that can locate records with data in fields similar to the data entered into the textbox
* There is a management option to sort records by field

### Record Grouping

* The user can create fields that will determine whether a record is part of a particular group or not
* Records can be made hidden or visible using these new fields as filters

### Master Key Management

* The user can change their master key

### User-Scheduled Maintenance Reminders

* The user has the option to schedule a time and date to change their master key
* The user has the option to schedule a time and date to update a password for a particular record
* A notification will appear for the user to change the relevant password or key at the specified time

### Lock/Unlock Workspace

* A tool can temporarily encrypt and decrypt the user’s database without closing the application

### Clear Clipboard

* A tool can be used to clear the clipboard at any time
* There will be an option for the clipboard to be automatically cleared a set time after copying a password

### Compression

* There will be a function that will compress files before they are saved
* There will be a function that will decompress files when they are loaded

### Tool Icons

* Tools have graphical representations in the form of square icons

### Records and Passwords Can be Accessed Quickly

* The time taken from clicking the executable to accessing a password or record is possible in under 15 seconds

### Links and URLs can be clicked

* The links and URLs are highlighted in blue and can be clicked to automatically load the website in the user’s default browser

### Hint or Security Question

* If the master password is incorrectly entered 5 times, a hint opens in a pop-up

# Design

## Breaking down the application

I have broken down the project into the following problems:

##### Encryption and Decryption

* How will libraries be used for encryption and decryption?
* When will encryption and decryption be used?
* How will data be parsed for encryption and decryption?

##### Database Management

* How will databases be structured?
* How can records be changed and sorted by the user?
* How can records be automatically sorted?
* How can changes to the database be temporarily stored for the user to cycle through?
* How can the user search for specific records?
* How can the user obtain data from the database?
* How can the user keep data secure when temporary leaving the computer unattended?

##### Password Generation

* How will passwords be generated?
* How will the user choose a password generation method?
* How will generated passwords be exported?
* How will parameters for passwords be implemented?
* How will the user choose to generate individual passwords or password lists?

##### User Interface

* How will different sections of the application be integrated?
* How will the user be instructed and informed about tools?
* How will the user be able to navigate forms and tabs?
* How will password strength be displayed?
* How will tools be represented?
* How will passwords only be shown at appropriate times?

##### File Management

* How will databases be created?
* How will databases be loaded?
* How will databases be saved?

##### Data Validation

* How will user input be subject to data validation?
* How will generated data be subject to data validation?
* How will loaded/imported data be subject to data validation?