# My password manager
# Kamil Misiak

## Dokumentacja techniczna

Spis treści:

## 1. Opis aplikacji

My Password Manager powstał aby pomóc nam zarządzać swoimi hasłami do różnych serwisów. Pozwala łatwo odnaleźć nasze dostępy składające się z pary login i hasło. Dzięki naszej aplikacji możemy w łatwy sposób wygenerować hasło do nowego konta w serwisie. Co zapewnia dużo większe bezpieczeństwo niż używanie tego samego hasło powiązanego z wieloma kontami. Konta przechowywane w password managera znajdują się lokalnie na naszym komputerze, dodatkowo są specjalnie zaszyfrowane dzięki czemu ich odczytanie bez podania głównego hasła powiązanego z plikiem jest prawie niemożliwe.

Aplikacja została przygotowana do funkcjonowania wieloplatformowego na takich platformach jak min. Windows, linux, Mac os. Oraz platformy mobile android oraz ios. Została uruchomiona pod kontrolą Mac os 10.14 oraz Windows 10 bez napotkania żadnych błędów. Aplikacja działa na większości systemów operacyjnych.

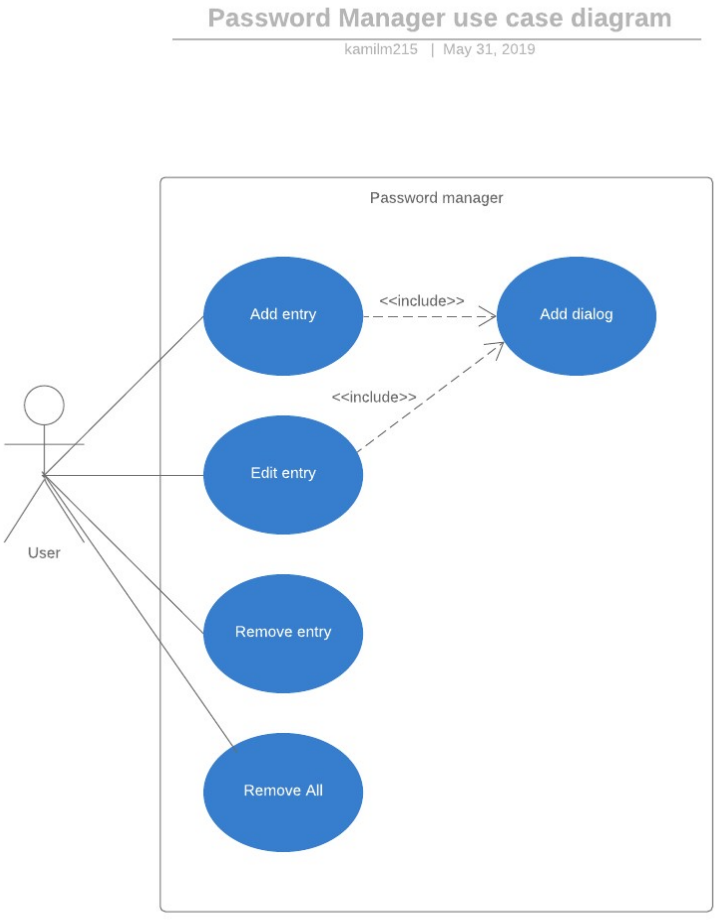
Aplikacja jest przeznaczona do wszelakiego rodzaju użytkowników których chcą zapewnić pełne bezpieczeństwo podczas korzystanie z internetu, bez potrzeby pamiętania dużej ilości skomplikowanych haseł.

Aplikacja została napisana w środowisku programistycznym Qt Creator 4.8.1, z wykorzystaniem takich komponentów jak QDialog, QWidget, QAbstractTableModel.

Haszowanie haseł odbywa się wykorzystaniem biblioteki SimpleCrypter napisanej przez Andre Somers'a.

## 2.Diagram przypadków użycia

Użytkownik po uruchumieniu systemu ma kilka podstawowych opcji do wykonania. Takich jak dodanie wpisu i edycja co uruchamia następnie nowe okno w który uzupełnia dane. Pozostałe opcje to usuwanie i usunięcie wszystkiego.



**3.Opis zastosowanych algorytmów i Diagramy klas**

Aplikacja password manager składa się z kilku Klas. Program została wykonana na podstawie wzorca projektowego MVC. W projekcie zachowano podstawowe wytyczne projektowania obiektowego takie jak abstrakcją, hermetyzacją, dziedziczenie i polimorfizm.

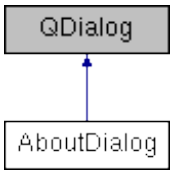
## Class List

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

| | |
|---|---|
| **AboutDialog** | |
| **AddDialog** | |
| **MyPasswordManager** | |
| **NewPasswordTab** | |
| **Pass** | |
| **PasswordDialog** | |
| **PasswordWidget** | |
| **SimpleCrypt** | Simple encryption and decryption of strings and byte arrays |
| TableModel | |

## AboutDialog Class Reference

Inheritance diagram for AboutDialog:

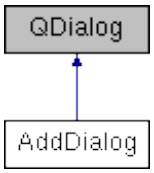



### Public Member Functions

**AboutDialog**(QWidget *parent=nullptr)

## AddDialog Class Reference

Inheritance diagram for AddDialog:




### Public Member Functions

**AddDialog**(QWidget *parent=nullptr)

void **generatePassword**(int length)

## Public Attributes

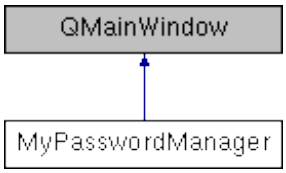
QLineEdit * **nameText**

QLineEdit * **loginText**

QLineEdit * **passwordText**

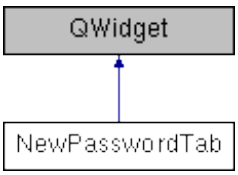
### MyPasswordManager Class Reference

Inheritance diagram for MyPasswordManager:




# NewPasswordTab Class Reference

Inheritance diagram for NewPasswordTab:




## Public Slots

void **addEntry**()

## Signals

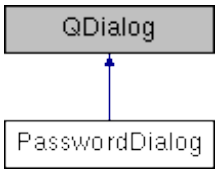
void **sendDetails**(QString name, QString login, QString password)

## Public Member Functions

**NewPasswordTab**(QWidget *parent=nullptr)

# PasswordDialog Class Reference

Inheritance diagram for PasswordDialog:




## Public Member Functions

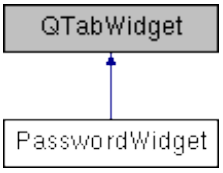
**PasswordDialog**(QWidget *parent=nullptr)

## Public Attributes

QLineEdit * **passwordText**

# PasswordWidget Class Reference

Inheritance diagram for PasswordWidget:




## Public Slots

void **showAddEntryDialog**()

QString **showPasswordDialog**()

void **showInfoDialog**()

void **addEntry**(QString name, QString login, QString password)

void **editEntry**()

void **removeEntry**()

void **hideEntry**()

## Signals

void **selectionChanged**(const QItemSelection &selected)

## Public Member Functions

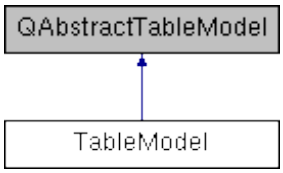
**PasswordWidget**(QWidget *parent=nullptr)

void **readFromFile**(const QString &fileName, const QString password)

void **writeToFile**(const QString &fileName, const QString password)

# TableModel Class Reference

Inheritance diagram for TableModel:




## Public Member Functions

**TableModel**(QObject *parent=nullptr)

**TableModel**(QList<**Pass**> passes, QObject *parent=nullptr)

int **rowCount**(const QModelIndex &parent) const override

int **columnCount**(const QModelIndex &parent) const override

QVariant **data**(const QModelIndex &index, int role) const override

QVariant **headerData**(int section, Qt::Orientation orientation, int role) const override

Qt::ItemFlags **flags**(const QModelIndex &index) const override

bool **setData**(const QModelIndex &index, const QVariant &value, int role=Qt::EditRole) override

bool **insertRows**(int position, int rows, const QModelIndex &index=QModelIndex()) override

bool **removeRows**(int position, int rows, const QModelIndex &index=QModelIndex()) override

bool **hidePassword**(const QModelIndex &index, int role)

QList<**Pass**> **getPasses**() const

# Pass Struct Reference

## Public Member Functions

bool **operator==**(const**Pass**&other) const

## Public Attributes

QString **name**

QString **login**

QString **password**

QString **hidePassword**

**4.Instrukcja obsługi**

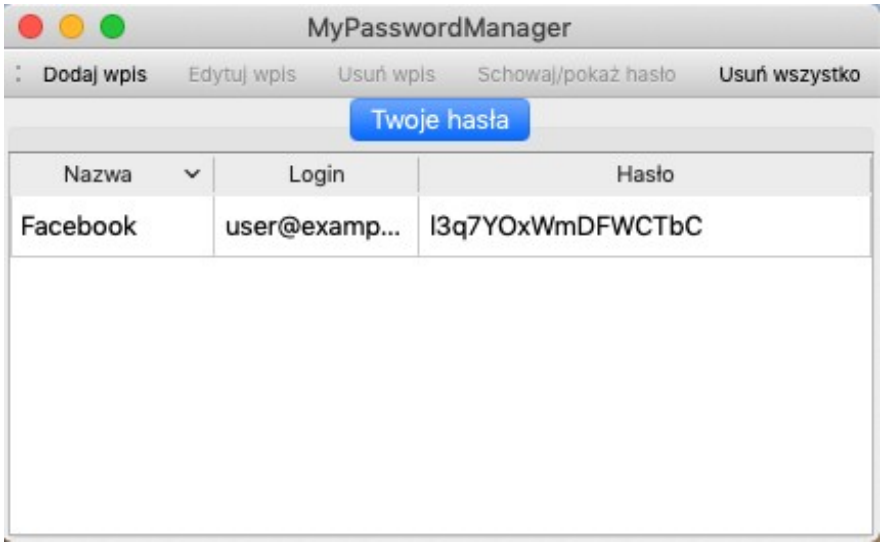
Uruchamiając aplikacje napotykamy ekran początkowy, na którym w głównej części znajduje się przycisk dodaj który pozwala dodać nowy wpis.. Oraz w górnej części aplikacji znajduje się toolbar w którym mamy możliwość wykonania podstawowych operacji zarządzania treścią haseł , na pierwszym zdjęciu ukazany jest ekran startowy. Po wybraniu opcji dodaj naszym oczom ukazuje się ekran dodawania nowego wpisu ukazany na zdjęciu 2.

Zdjęcie 1

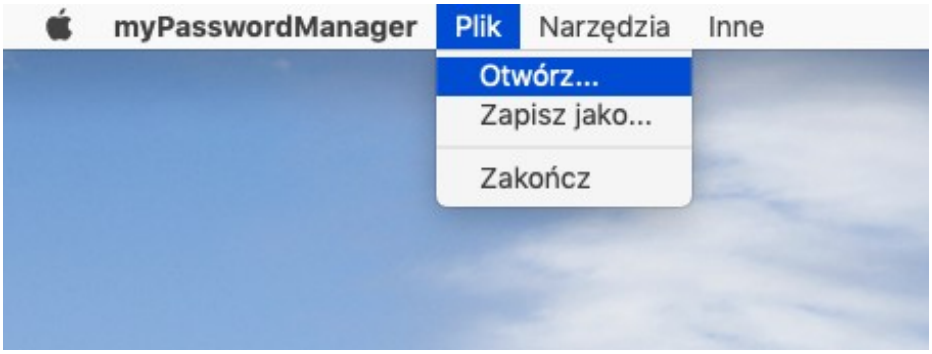

Zdjęcie 2


Po dodaniu nowego wpisu na naszym ekranie ukazuje się tabela z zapisanymi danymi ukazana na zdjęciu 3. Po wybraniu kursorem myszy jednego z wpisów uaktywniają się opcje edycji wpisów. Mamy również kilka dodatkowych opcji związanych z zapisem i odczytem do pliku znajdujących się menu bar ukazanym na zdjęciu 4. Odczyt i zapis realizowany jest przez narzędzia domyślne dla naszego systemu operacyjnego.

Zdjęcie 3



Zdjęcie 4


myPasswordManager
Plik
Narzędzia
Inne
Otwórz...
Zapisz jako...
Zakończ