Password Manager

1.0

Generated by Doxygen 1.9.8

Chapter 1

Password Manager Documentation

1.1 Introduction

Welcome to the Password Manager Documentation. This project is a tool for managing your passwords and sensitive information. It includes features such as:

- Adding and managing password entries.
- Encrypting data with OpenSSL.
- Terminal interface for database operations.

1.2 Features

- **Secure encryption**: All data is encrypted using AES-256-CBC.
- **User-friendly**: Simple command-line interface for managing entries.
- **Flexibility**: Supports various types of entries, such as passwords and PINs.

1.3 Usage

To start using the application:

1. Build the project.
2. Run the main executable.
3. Create or open existing database.
4. Follow the menu options to add, view, and manage entries.

1.3.1 Example database

Example with example_database.bin

Password: qwerty123

```
$ ./PasswordManager
Password Manager
Enter source database name (new or existing): ../example_database.bin
Enter master password: qwerty123
Loading database from file: ../example_database.bin
Database loaded successfully.

    Password Manager Menu:
    1. Add Entry
    2. Remove Entry
    3. Display Entries
    4. Save and exit
    5. Exit without saving
    Choose an option: 1
Enter title: main email
Choose type of entry:
1. Password Entry
2. PIN Entry
1
Enter username: wojtek@mail.com
Enter password: password123
Enter description:


    Password Manager Menu:
    1. Add Entry
    2. Remove Entry
    3. Display Entries
    4. Save and exit
    5. Exit without saving
    Choose an option: 3

fb
username: Wojtek
password: strongpassword
description: meta

Credit card
PIN: 4321
description: my bank

main email
username: wojtek@mail.com
password: password123
description:

    Password Manager Menu:
    1. Add Entry
    2. Remove Entry
    3. Display Entries
    4. Save and exit
    5. Exit without saving
    Choose an option: 4
Saving database to file: ../example_database.bin
Database saved successfully.
```

1.3.2 Author

This project was developed by Wojciech Kozub.

1.3.3 Copyright

GNU General Public License.

Chapter 2

Building

Application requires the OpenSSL library.

2.1 Linux

2.1.1 Install the OpenSSL library

On Ubuntu, install the OpenSSL development package by running:

```
sudo apt update
sudo apt-get install libssl-dev
```

2.1.2 Clone the repository

```
git clone https://github.com/wokozub/PasswordManager.git
cd PasswordManager
```

2.1.3 Build the project using CMake

A simple way to build this project is with cmake.

```
mkdir build
cd build
cmake ..
make
./PasswordManager
```

2.1.4 Additional Notes

Ensure you have cmake installed. If not, you can install it using:

```
sudo apt-get install cmake
```

Chapter 3

Topic Index

3.1 Topics

Here is a list of all topics with brief descriptions:

Chapter 4

Namespace Index

4.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Chapter 5

Hierarchical Index

5.1 Class Hierarchy

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

Chapter 6

Class Index

6.1 Class List

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

Chapter 7

File Index

7.1 File List

Here is a list of all files with brief descriptions:

Chapter 8

Topic Documentation

8.1 Core Classes

Classes

- class wk::Entry

  Base class for an entry (PasswordEntry, PinEntry).
- class wk::PasswordManager

  Handles the main menu for user interaction.

8.1.1 Detailed Description

Fundamental classes of the application.

8.2 Entries

Classes

- class wk::PasswordEntry

  Represents a password entry in the password manager.
- class wk::PinEntry

  Represents a PIN code entry in the password manager.

8.2.1 Detailed Description

Classes representing different types of entries.

8.3 Utilities

Classes

- class wk::Database

  Manages a collection of password and PIN entries.
- class wk::Encryptor

  A class responsible for encrypting and decrypting data using AES-256-CBC.

8.3.1 Detailed Description

Tools supporting the application functionality.

Chapter 9

Namespace Documentation

9.1 wk Namespace Reference

Classes

- class Database

  Manages a collection of password and PIN entries.
- class Encryptor

  A class responsible for encrypting and decrypting data using AES-256-CBC.
- class Entry

  Base class for an entry (PasswordEntry, PinEntry).
- class PasswordEntry

  Represents a password entry in the password manager.
- class PasswordManager

  Handles the main menu for user interaction.
- class PinEntry

  Represents a PIN code entry in the password manager.

Chapter 10

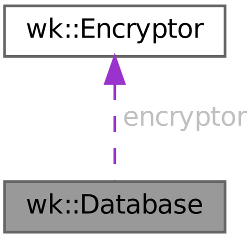
Class Documentation

10.1 wk::Database Class Reference

Manages a collection of password and PIN entries.

```
#include <Database.hpp>
```

Collaboration diagram for wk::Database:



Public Member Functions

- Database (const std::string &password)

  Constructs a Database with a master password.
- bool titleExists (const std::string &title) const

  Checking for existing titles. Using in 'PasswordManager::newEntry' to ensure that each new entry added to the database has a unique title.
- void addEntry (const std::variant< PasswordEntry, PinEntry > &entry)

  Adds a new entry to the database.
- void removeEntry (const std::string &title)

  Removes an entry from the database by title.
- void displayEntries () const

  Displays all entries in the database.
- void saveToFile (const std::string &filename)

  Saves all entries to a file.
- void loadFromFile (const std::string &filename)

  Loads entries from a file.

Private Attributes

- std::vector< std::variant< PasswordEntry, PinEntry > > entries
- Encryptor encryptor

10.1.1 Detailed Description

Manages a collection of password and PIN entries.

The Database class provides functionalities to add, remove, display, and persist entries. It supports both password and PIN entries using variants.

10.1.2 Constructor & Destructor Documentation

10.1.2.1 Database()

```
wk::Database::Database (
            const std::string & password )
```

Constructs a Database with a master password.

Parameters

password	The master password to encrypt/decrypt file.

10.1.3 Member Function Documentation

10.1.3.1 addEntry()

```
void wk::Database::addEntry (
            const std::variant< PasswordEntry, PinEntry > & entry )
```

Adds a new entry to the database.

Parameters

entry	A variant containing either a PasswordEntry or PinEntry.

References entries.

10.1.3.2 displayEntries()

```
void wk::Database::displayEntries ( ) const
```

Displays all entries in the database.

References entries.

10.1.3.3 loadFromFile()

```
void wk::Database::loadFromFile (
             const std::string & filename )
```

Loads entries from a file.

Parameters

filename	The file to load entries from.

References wk::Encryptor::decryptData(), encryptor, entries, wk::PasswordEntry::load(), and wk::PinEntry::load().

10.1.3.4 removeEntry()

```
void wk::Database::removeEntry (
             const std::string & title )
```

Removes an entry from the database by title.

Parameters

title	The title of the entry to remove.

References entries.

10.1.3.5 saveToFile()

```
void wk::Database::saveToFile (
             const std::string & filename )
```

Saves all entries to a file.

Parameters

filename	The file to save entries into.

References wk::Encryptor::encryptData(), encryptor, and entries.

10.1.3.6 titleExists()

```
bool wk::Database::titleExists (
             const std::string & title ) const
```

Checking for existing titles. Using in 'PasswordManager::newEntry' to ensure that each new entry added to the database has a unique title.

References entries.

10.1.4 Member Data Documentation

10.1.4.1 encryptor

```
Encryptor wk::Database::encryptor  [private]
```

Handles encryption and decryption.

10.1.4.2 entries

```
std::vector<std::variant<PasswordEntry, PinEntry> > wk::Database::entries  [private]
```

Container for storing database entries.

The documentation for this class was generated from the following files:

- include/Database.hpp
- src/Database.cpp

10.2 wk::Encryptor Class Reference

A class responsible for encrypting and decrypting data using AES-256-CBC.

```
#include <Encryptor.hpp>
```

Public Member Functions

- Encryptor (const std::string &password)

  Parametrized constructor.
- ∼Encryptor ()

  Destructor.
- bool encryptData (const std::string &data, const std::string &outputFile)

  Encrypts the provided data and saves it to a binary file.
- bool decryptData (const std::string &inputFile, std::string &output)

  Decrypts data from an input binary file.

Private Attributes

- unsigned char key [EVP_MAX_KEY_LENGTH]
- unsigned char iv [EVP_MAX_IV_LENGTH]

10.2.1 Detailed Description

A class responsible for encrypting and decrypting data using AES-256-CBC.

The `Encryptor` class uses the OpenSSL library to encrypt and decrypt data with the AES-256-CBC algorithm. It generates an encryption key and an initialization vector (IV) from the provided password using SHA-512. This class requires the OpenSSL library.

10.2.2 Constructor & Destructor Documentation

10.2.2.1 Encryptor()

```
wk::Encryptor::Encryptor (
             const std::string & password )
```

Parametrized constructor.

This constructor generates the encryption key and IV using SHA-512 based on the provided password.

Parameters

password	The password used to generate the encryption key and IV.

References iv, and key.

10.2.2.2 ∼Encryptor()

```
wk::Encryptor::∼Encryptor ( )
```

Destructor.

Cleans up any allocated resources.

10.2.3 Member Function Documentation

10.2.3.1 decryptData()

```
bool wk::Encryptor::decryptData (
             const std::string & inputFile,
             std::string & output )
```

Decrypts data from an input binary file.

Using in 'Database::loadFromFile' This method reads the encrypted data from the input file and decrypts it using AES-256-CBC.

Parameters

inputFile	The binary file containing the encrypted data.
output	The string to store the decrypted data.

Returns

True if decryption is successful, false otherwise.

References iv, and key.

10.2.3.2 encryptData()

```
bool wk::Encryptor::encryptData (
             const std::string & data,
             const std::string & outputFile )
```

Encrypts the provided data and saves it to a binary file.

Using in 'Database::saveToFile' This method encrypts the input data using AES-256-CBC and writes the encrypted content to an output file.

Parameters

data	The input data to be encrypted.
outputFile	The binary file to save the encrypted data.

Returns

True if encryption is successful, false otherwise.

References iv, and key.

10.2.4 Member Data Documentation

10.2.4.1 iv

```
unsigned char wk::Encryptor::iv[EVP_MAX_IV_LENGTH]  [private]
```

The initialization vector (IV)

10.2.4.2 key

```
unsigned char wk::Encryptor::key[EVP_MAX_KEY_LENGTH]  [private]
```

The encryption key generated from the password

The documentation for this class was generated from the following files:

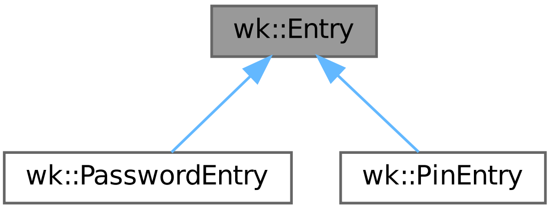
- include/Encryptor.hpp
- src/Encryptor.cpp

10.3 wk::Entry Class Reference

Base class for an entry (PasswordEntry, PinEntry).

```
#include <Entry.hpp>
```

Inheritance diagram for wk::Entry:



Public Member Functions

- Entry ()

  Default constructor.
- Entry (const std::string &title, const std::string &description)

  Parametrized constructor.
- virtual ∼Entry ()

  Destructor.
- virtual char getType () const =0

  Returns the type of the entry.
- virtual void display () const

  Displays the entry details.
- virtual void save (std::ostream &stream) const

  To serialized the entry to a stream.
- virtual void load (std::istream &stream)

  To deserialized entry data from a stream.
- std::string getTitle () const

  Returns the title of the entry.
- std::string getDescription () const

  Returns the description of the entry.
- void setTitle (std::string title)

  Sets the title of the entry.
- void setDescription (std::string description)

  Sets the description of the entry.

Protected Attributes

- std::string title
- std::string description

10.3.1 Detailed Description

Base class for an entry (PasswordEntry, PinEntry).

This class represents a generic entry with a title and description. It serves as a base class for specialized entry types: PasswordEntry and PinEntry.

10.3.2 Constructor & Destructor Documentation

10.3.2.1 Entry() [1/2]

```
wk::Entry::Entry ( )
```

Default constructor.

Initializes an empty entry with an empty title and description.

10.3.2.2 Entry() [2/2]

```
wk::Entry::Entry (
             const std::string & title,
             const std::string & description )
```

Parametrized constructor.

Initializes the entry with a specified title and description.

Parameters

title	The title of the entry.
description	The description of the entry.

10.3.2.3 ∼Entry()

```
wk::Entry::∼Entry ( )  [virtual]
```

Destructor.

Virtual destructor to allow proper cleanup in derived classes.

10.3.3 Member Function Documentation

10.3.3.1 display()

```
void wk::Entry::display ( ) const  [virtual]
```

Displays the entry details.

Reimplemented in wk::PasswordEntry, and wk::PinEntry.

10.3.3.2 getDescription()

```
std::string wk::Entry::getDescription ( ) const
```

Returns the description of the entry.

References description.

10.3.3.3 getTitle()

```
std::string wk::Entry::getTitle ( ) const
```

Returns the title of the entry.

References title.

10.3.3.4 getType()

```
virtual char wk::Entry::getType ( ) const  [pure virtual]
```

Returns the type of the entry.

Returns

A character: 'w' for password, 'n' for pin.

Implemented in wk::PasswordEntry, and wk::PinEntry.

10.3.3.5 load()

```
void wk::Entry::load (
            std::istream & stream )  [virtual]
```

To deserialized entry data from a stream.

Parameters

stream	The input stream.

Reimplemented in wk::PasswordEntry, and wk::PinEntry.

10.3.3.6 save()

```
void wk::Entry::save (
             std::ostream & stream ) const  [virtual]
```

To serialized the entry to a stream.

Parameters

stream	The output stream.

Reimplemented in wk::PasswordEntry, and wk::PinEntry.

10.3.3.7 setDescription()

```
void wk::Entry::setDescription (
             std::string description )
```

Sets the description of the entry.

Parameters

description	The description to set.

References description.

10.3.3.8 setTitle()

```
void wk::Entry::setTitle (
             std::string title )
```

Sets the title of the entry.

Parameters

title	The title to set.

References title.

10.3.4 Member Data Documentation

10.3.4.1 description

```
std::string wk::Entry::description [protected]
```

The description of the entry.

10.3.4.2 title

```
std::string wk::Entry::title [protected]
```

The title of the entry.

The documentation for this class was generated from the following files:

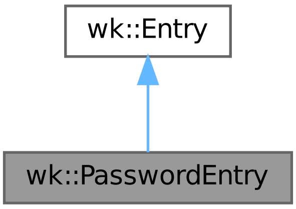
- include/Entry.hpp
- src/Entry.cpp

10.4 wk::PasswordEntry Class Reference

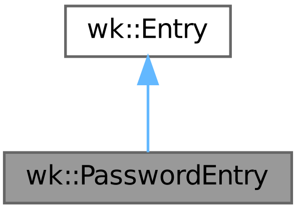
Represents a password entry in the password manager.

```
#include <PasswordEntry.hpp>
```

Inheritance diagram for wk::PasswordEntry:



Collaboration diagram for wk::PasswordEntry:



Public Member Functions

- PasswordEntry ()

  Default constructor.
- PasswordEntry (const std::string &title, const std::string &username, const std::string &password, const std::string &description)

  Parametrized constructor.
- char getType () const override

  Returns the type of the entry.
- void display () const override

  Displays the entry details.
- void save (std::ostream &stream) const

  To serialized the entry to a stream.
- void load (std::istream &stream)

  To deserialized entry data from a stream.
- std::string getUsername () const

  Gets the username.
- std::string getPassword () const

  Gets the password.
- void setUsername (std::string username)

  Sets the username.
- void setPassword (std::string password)

  Sets the password.

Public Member Functions inherited from wk::Entry

- Entry ()

  Default constructor.
- Entry (const std::string &title, const std::string &description)

  Parametrized constructor.
- virtual ∼Entry ()

  Destructor.
- std::string getTitle () const

  Returns the title of the entry.
- std::string getDescription () const

  Returns the description of the entry.
- void setTitle (std::string title)

  Sets the title of the entry.
- void setDescription (std::string description)

  Sets the description of the entry.

Private Attributes

- std::string m_username
- std::string m_password

Additional Inherited Members

Protected Attributes inherited from wk::Entry

- std::string title
- std::string description

10.4.1 Detailed Description

Represents a password entry in the password manager.

This class extends the `Entry` class and represents a password entry with additional fields such as username and password.

10.4.2 Constructor & Destructor Documentation

10.4.2.1 PasswordEntry() [1/2]

```
wk::PasswordEntry::PasswordEntry ( )
```

Default constructor.

10.4.2.2 PasswordEntry() [2/2]

```
wk::PasswordEntry::PasswordEntry (
            const std::string & title,
            const std::string & username,
            const std::string & password,
            const std::string & description )
```

Parametrized constructor.

Parameters

title	The title of the entry.
username	The username.
password	The password.
description	Additional information.

10.4.3 Member Function Documentation

10.4.3.1 display()

```
void wk::PasswordEntry::display ( ) const  [override], [virtual]
```

Displays the entry details.

Reimplemented from wk::Entry.

References wk::Entry::description, m_password, m_username, and wk::Entry::title.

10.4.3.2 getPassword()

```
std::string wk::PasswordEntry::getPassword ( ) const
```

Gets the password.

Returns

The password of the entry.

References m_password.

10.4.3.3 getType()

```
char wk::PasswordEntry::getType ( ) const  [override], [virtual]
```

Returns the type of the entry.

Returns

A character: 'w' for password, 'n' for pin.

Implements wk::Entry.

10.4.3.4 getUsername()

```
std::string wk::PasswordEntry::getUsername ( ) const
```

Gets the username.

Returns

The username of the entry.

References m_username.

10.4.3.5 load()

```
void wk::PasswordEntry::load (
             std::istream & stream )  [virtual]
```

To deserialized entry data from a stream.

Parameters

stream	The input stream.

Reimplemented from wk::Entry.

References wk::Entry::description, m_password, m_username, and wk::Entry::title.

10.4.3.6 save()

```
void wk::PasswordEntry::save (
             std::ostream & stream ) const  [virtual]
```

To serialized the entry to a stream.

Parameters

stream	The output stream.

Reimplemented from wk::Entry.

References wk::Entry::description, getType(), m_password, m_username, and wk::Entry::title.

10.4.3.7 setPassword()

```
void wk::PasswordEntry::setPassword (
             std::string password )
```

Sets the password.

Parameters

password	The new password to set.

References m_password.

10.4.3.8 setUsername()

```
void wk::PasswordEntry::setUsername (
             std::string username )
```

Sets the username.

Parameters

username	The new username to set.

References m_username.

10.4.4 Member Data Documentation

10.4.4.1 m_password

```
std::string wk::PasswordEntry::m_password  [private]
```

The password of the entry.

10.4.4.2 m_username

```
std::string wk::PasswordEntry::m_username  [private]
```

The username of the entry.

The documentation for this class was generated from the following files:

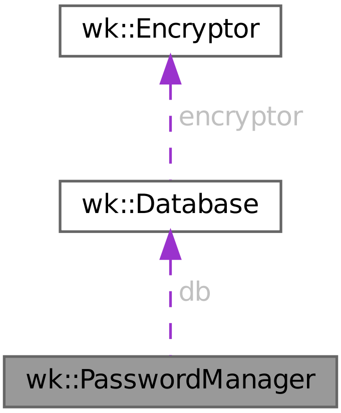
- include/PasswordEntry.hpp
- src/PasswordEntry.cpp

10.5 wk::PasswordManager Class Reference

Handles the main menu for user interaction.

```
#include <PasswordManager.hpp>
```

Collaboration diagram for wk::PasswordManager:



Public Member Functions

- PasswordManager (const std::string &dbName, const std::string &masterPassword)

    Parametrized constructor.
- void run ()

    Runs the password manager interface. ∗ This method presents a menu to the user and processes the user's choices.

Private Member Functions

- void displayMenu ()

  Displays the main menu options. ∗ This method displays the list of actions the user can choose from, such as adding an entry, removing an entry, displaying entries, saving the database, or exiting.
- void newEntry ()

  Handle the 'Database::addEntry' Adds a new entry to the database.
- void deleteEntry ()

  Handle the 'Database::removeEntry' Removes an entry from the database.
- void saveDB ()

  Handle the 'Database::saveToFile' Saves the current database to a file.
- void loadDB ()

  Handle the 'Database::loadFromFile' Loads the database from a file.

Private Attributes

- Database db
- std::string filename

10.5.1 Detailed Description

Handles the main menu for user interaction.

The `PasswordManager` class is responsible for interacting with the `Database` class to manage password entries. It provides logic of load and saving the database from a file and also provides functionality to add, remove, and display password entries.

10.5.2 Constructor & Destructor Documentation

10.5.2.1 PasswordManager()

```
wk::PasswordManager::PasswordManager (
            const std::string & dbName,
            const std::string & masterPassword )
```

Parametrized constructor.

This constructor initializes the password manager and sets up the database.

Parameters

dbName	The relative path and name of the database file (in UNIX-style path format)
masterPassword	The master password used to access the database.

10.5.3 Member Function Documentation

10.5.3.1 deleteEntry()

```
void wk::PasswordManager::deleteEntry ( )  [private]
```

Handle the 'Database::removeEntry' Removes an entry from the database.

References db, and wk::Database::removeEntry().

10.5.3.2 displayMenu()

```
void wk::PasswordManager::displayMenu ( )  [private]
```

Displays the main menu options. ∗ This method displays the list of actions the user can choose from, such as adding an entry, removing an entry, displaying entries, saving the database, or exiting.

10.5.3.3 loadDB()

```
void wk::PasswordManager::loadDB ( )  [private]
```

Handle the 'Database::loadFromFile' Loads the database from a file.

References db, filename, and wk::Database::loadFromFile().

10.5.3.4 newEntry()

```
void wk::PasswordManager::newEntry ( )  [private]
```

Handle the 'Database::addEntry' Adds a new entry to the database.

References wk::Database::addEntry(), db, and wk::Database::titleExists().

10.5.3.5 run()

```
void wk::PasswordManager::run ( )
```

Runs the password manager interface. ∗ This method presents a menu to the user and processes the user's choices.

References db, deleteEntry(), wk::Database::displayEntries(), displayMenu(), loadDB(), newEntry(), and saveDB().

10.5.3.6 saveDB()

```
void wk::PasswordManager::saveDB ( )  [private]
```

Handle the 'Database::saveToFile' Saves the current database to a file.

References db, filename, and wk::Database::saveToFile().

10.5.4 Member Data Documentation

10.5.4.1 db

`Database wk::PasswordManager::db [private]`

The `Database` object.

10.5.4.2 filename

`std::string wk::PasswordManager::filename [private]`

UNIX-style path format for database

The documentation for this class was generated from the following files:

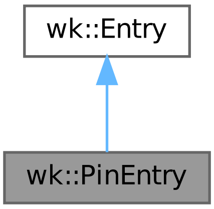
- include/PasswordManager.hpp
- src/PasswordManager.cpp

10.6 wk::PinEntry Class Reference

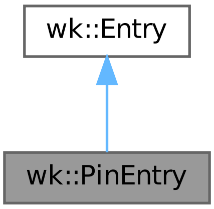
Represents a PIN code entry in the password manager.

`#include <PinEntry.hpp>`

Inheritance diagram for wk::PinEntry:



Collaboration diagram for wk::PinEntry:



Public Member Functions

- PinEntry ()

  Default constructor.
- PinEntry (const std::string &title, const int &pin, const std::string &description)

  Parametrized constructor.
- char getType () const override

  Returns the type of the entry.
- void display () const override

  Displays the entry details.
- void save (std::ostream &stream) const

  To serialized the entry to a stream.
- void load (std::istream &stream)

  To deserialized entry data from a stream.
- int getPin () const

  Gets the PIN code.
- void setPin (int pin)

  Sets the PIN code.

Public Member Functions inherited from wk::Entry

- Entry ()

  Default constructor.
- Entry (const std::string &title, const std::string &description)

  Parametrized constructor.
- virtual ∼Entry ()

  Destructor.
- std::string getTitle () const

  Returns the title of the entry.
- std::string getDescription () const

  Returns the description of the entry.
- void setTitle (std::string title)

  Sets the title of the entry.
- void setDescription (std::string description)

  Sets the description of the entry.

Private Attributes

- int m_pin

Additional Inherited Members

Protected Attributes inherited from wk::Entry

- std::string title
- std::string description

10.6.1 Detailed Description

Represents a PIN code entry in the password manager.

This class extends the `Entry` class and represents a PIN entry with additional field intiger type.

10.6.2 Constructor & Destructor Documentation

10.6.2.1 PinEntry() [1/2]

```
wk::PinEntry::PinEntry ( )
```

Default constructor.

10.6.2.2 PinEntry() [2/2]

```
wk::PinEntry::PinEntry (
             const std::string & title,
             const int & pin,
             const std::string & description )
```

Parametrized constructor.

Parameters

title	The title of the entry.
pin	The personal identification number.
description	Additional information.

10.6.3 Member Function Documentation

10.6.3.1 display()

```
void wk::PinEntry::display ( ) const  [override], [virtual]
```

Displays the entry details.

Reimplemented from wk::Entry.

References wk::Entry::description, m_pin, and wk::Entry::title.

10.6.3.2 getPin()

```
int wk::PinEntry::getPin ( ) const
```

Gets the PIN code.

Returns

The PIN code of the entry.

References m_pin.

10.6.3.3 getType()

```
char wk::PinEntry::getType ( ) const  [override], [virtual]
```

Returns the type of the entry.

Returns

A character: 'w' for password, 'n' for pin.

Implements wk::Entry.

10.6.3.4 load()

```
void wk::PinEntry::load (
             std::istream & stream )  [virtual]
```

To deserialized entry data from a stream.

Parameters

stream	The input stream.

Reimplemented from wk::Entry.

References wk::Entry::description, m_pin, and wk::Entry::title.

10.6.3.5 save()

```
void wk::PinEntry::save (
             std::ostream & stream ) const  [virtual]
```

To serialized the entry to a stream.

Parameters

stream	The output stream.

Reimplemented from wk::Entry.

References wk::Entry::description, getType(), m_pin, and wk::Entry::title.

10.6.3.6 setPin()

```
void wk::PinEntry::setPin (
             int pin )
```

Sets the PIN code.

Parameters

pin	The new PIN code to set.

References m_pin.

10.6.4 Member Data Documentation

10.6.4.1 m_pin

```
int wk::PinEntry::m_pin  [private]
```

The PIN code of the entry.

The documentation for this class was generated from the following files:

- include/PinEntry.hpp
- src/PinEntry.cpp

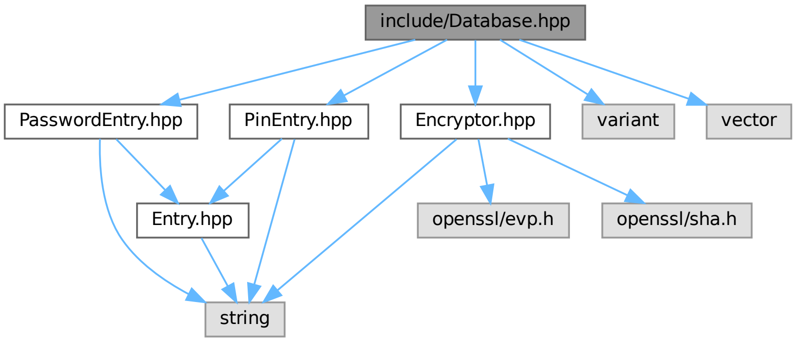
Chapter 11

File Documentation

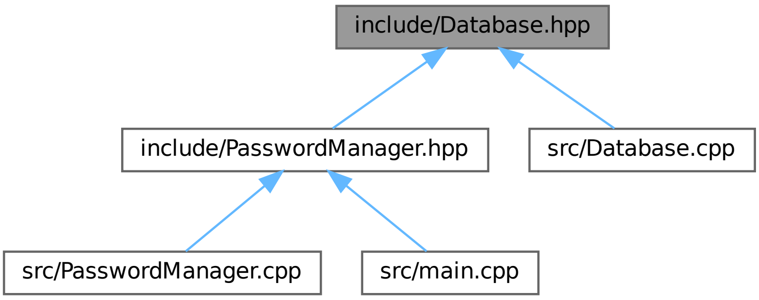
11.1 include/Database.hpp File Reference

```
#include "PasswordEntry.hpp"
#include "PinEntry.hpp"
#include "Encryptor.hpp"
#include <variant>
#include <vector>
```

Include dependency graph for Database.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class wk::Database

  Manages a collection of password and PIN entries.

Namespaces

- namespace wk

11.2 Database.hpp

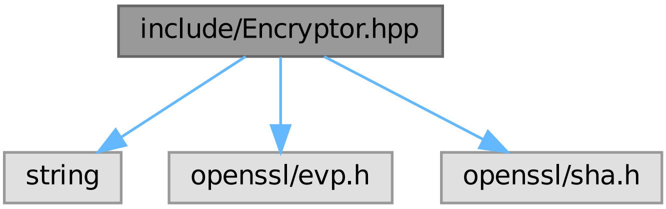
Go to the documentation of this file.

```
#ifndef DATABASE_HPP
#define DATABASE_HPP

#include "PasswordEntry.hpp"
#include "PinEntry.hpp"
#include "Encryptor.hpp"
#include <variant>
#include <vector>

namespace wk {

    class Database {
    public:
        Database(const std::string& password);

        bool titleExists(const std::string& title) const;

        void addEntry(const std::variant<PasswordEntry, PinEntry>& entry);

        void removeEntry(const std::string& title);

        void displayEntries() const;

        void saveToFile(const std::string& filename);

        void loadFromFile(const std::string& filename);
    private:
        std::vector<std::variant<PasswordEntry, PinEntry» entries;
        Encryptor encryptor;
    };

}

#endif
```

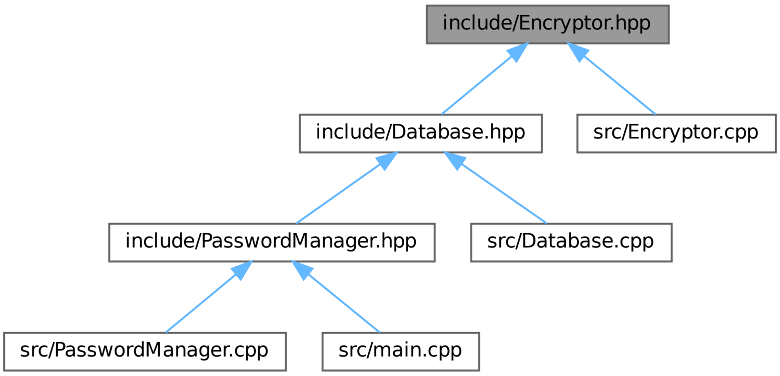
11.3 include/Encryptor.hpp File Reference

```
#include <string>
#include <openssl/evp.h>
#include <openssl/sha.h>
```

Include dependency graph for Encryptor.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class wk::Encryptor

  A class responsible for encrypting and decrypting data using AES-256-CBC.

Namespaces

- namespace wk

11.4 Encryptor.hpp

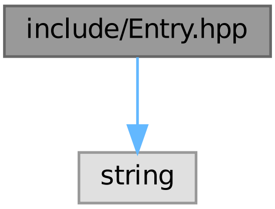
Go to the documentation of this file.

```
#ifndef ENCRYPTOR_HPP
#define ENCRYPTOR_HPP

#include <string>
#include <openssl/evp.h>
#include <openssl/sha.h>

namespace wk {

    class Encryptor {
    public:
        Encryptor(const std::string& password);

        ~Encryptor();

        bool encryptData(const std::string& data, const std::string& outputFile);

        bool decryptData(const std::string& inputFile, std::string& output);

    private:
        unsigned char key[EVP_MAX_KEY_LENGTH];
        unsigned char iv[EVP_MAX_IV_LENGTH];
    };

}

#endif
```

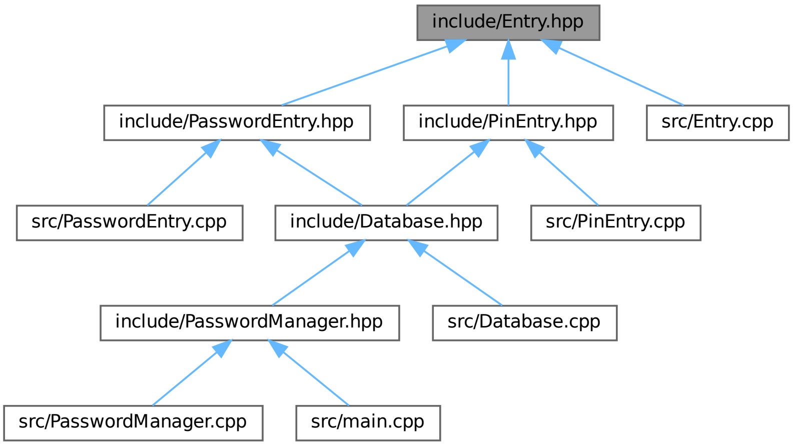
11.5 include/Entry.hpp File Reference

```
#include <string>
```

Include dependency graph for Entry.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class wk::Entry

  Base class for an entry (PasswordEntry, PinEntry).

Namespaces

- namespace wk

11.6 Entry.hpp

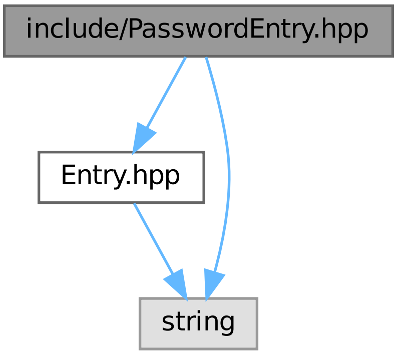
Go to the documentation of this file.

```
#ifndef ENTRY_HPP
#define ENTRY_HPP

#include <string>

namespace wk {

    class Entry {
    public:
        Entry();

        Entry(const std::string& title, const std::string& description);

        virtual ~Entry();

        virtual char getType() const = 0;

        virtual void display() const;

        virtual void save(std::ostream& stream) const;

        virtual void load(std::istream& stream);

        std::string getTitle() const;

        std::string getDescription() const;

        void setTitle(std::string title);

        void setDescription(std::string description);
    protected:
        std::string title;
        std::string description;
    };

}

#endif
```

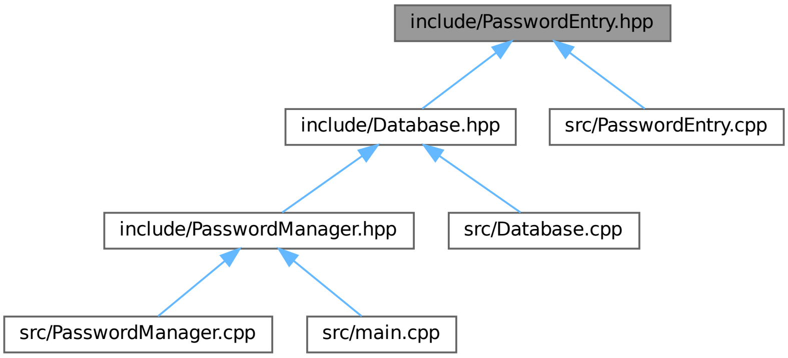
11.7 include/PasswordEntry.hpp File Reference

```
#include "Entry.hpp"
#include <string>
```

Include dependency graph for PasswordEntry.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class wk::PasswordEntry

  Represents a password entry in the password manager.

Namespaces

- namespace wk

11.8 PasswordEntry.hpp

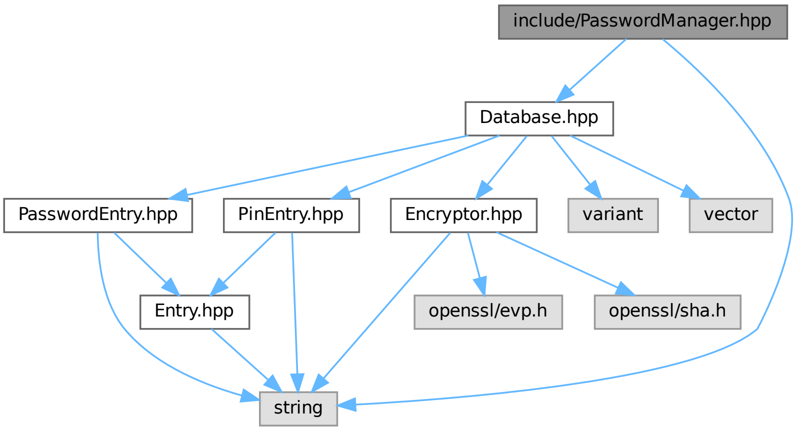
Go to the documentation of this file.

```
#ifndef PASSWORD_ENTRY_HPP
#define PASSWORD_ENTRY_HPP

#include "Entry.hpp"
#include <string>

namespace wk {

    class PasswordEntry : public Entry {
    public:
        PasswordEntry();

        PasswordEntry(const std::string& title, const std::string& username, const std::string& password, const std::string& description);

        char getType() const override;
        void display() const override;

        void save(std::ostream& stream) const;
        void load(std::istream& stream);

        std::string getUsername() const;

        std::string getPassword() const;

        void setUsername(std::string username);

        void setPassword(std::string password);
    private:
        std::string m_username;
        std::string m_password;
    };

}

#endif
```

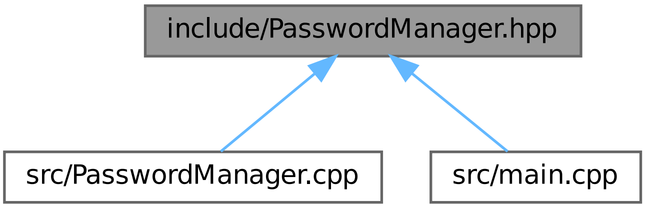
11.9 include/PasswordManager.hpp File Reference

```
#include "Database.hpp"
#include <string>
```

Include dependency graph for PasswordManager.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class wk::PasswordManager

  Handles the main menu for user interaction.

Namespaces

- namespace wk

11.10 PasswordManager.hpp

Go to the documentation of this file.

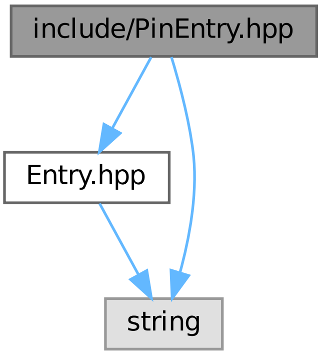
```
00001 #ifndef PASSWORD_MANAGER_HPP
00002 #define PASSWORD_MANAGER_HPP
00003
```

```
#include "Database.hpp"
#include <string>

namespace wk {

    class PasswordManager {
    public:
        PasswordManager(const std::string& dbName, const std::string& masterPassword);

        void run();

    private:
        Database db;
        std::string filename;
        void displayMenu();

        void newEntry();

        void deleteEntry();

        void saveDB();

        void loadDB();
    };

}

#endif
```

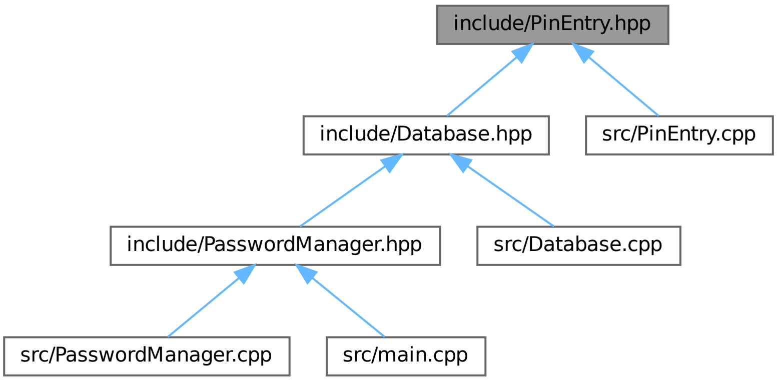
11.11 include/PinEntry.hpp File Reference

```
#include "Entry.hpp"
#include <string>
```

Include dependency graph for PinEntry.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class wk::PinEntry

  Represents a PIN code entry in the password manager.

Namespaces

- namespace wk

11.12 PinEntry.hpp

Go to the documentation of this file.

```
#ifndef PIN_ENTRY_HPP
#define PIN_ENTRY_HPP

#include "Entry.hpp"
#include <string>

namespace wk {

    class PinEntry : public Entry {
    public:
        PinEntry();

        PinEntry(const std::string& title, const int& pin, const std::string& description);

        char getType() const override;
        void display() const override;

        void save(std::ostream& stream) const;
        void load(std::istream& stream);

        int getPin() const;

        void setPin(int pin);
    private:
        int m_pin;
    };

}

#endif
```

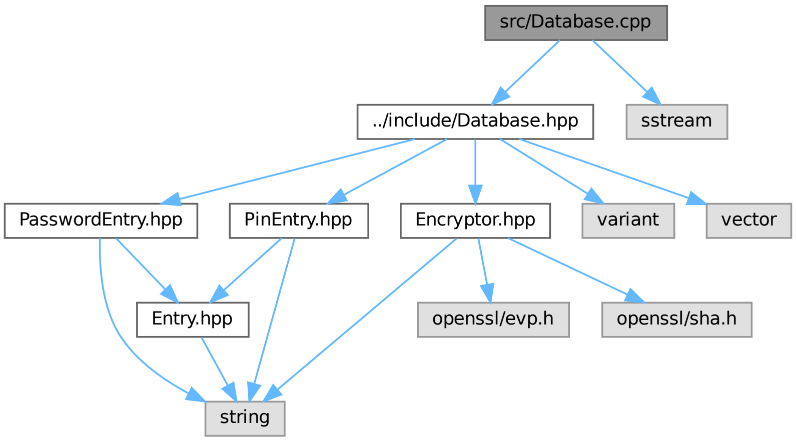
11.13 manual/building.md File Reference

11.14 manual/mainpage.md File Reference

11.15 src/Database.cpp File Reference

```
#include "../include/Database.hpp"
#include <sstream>
```

Include dependency graph for Database.cpp:



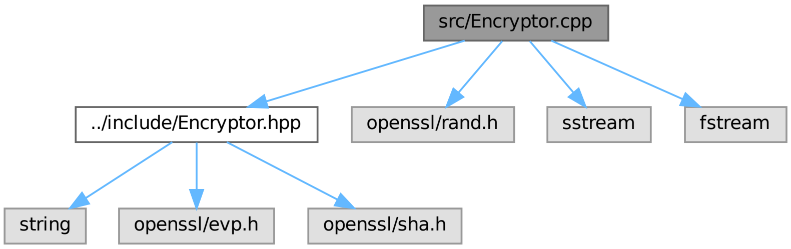
Namespaces

- namespace wk

11.16 src/Encryptor.cpp File Reference

```
#include "../include/Encryptor.hpp"
#include <openssl/rand.h>
#include <sstream>
#include <fstream>
```

Include dependency graph for Encryptor.cpp:



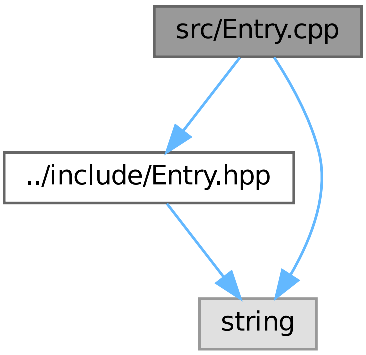
Namespaces

- namespace wk

11.17 src/Entry.cpp File Reference

```
#include "../include/Entry.hpp"
#include <string>
```

Include dependency graph for Entry.cpp:



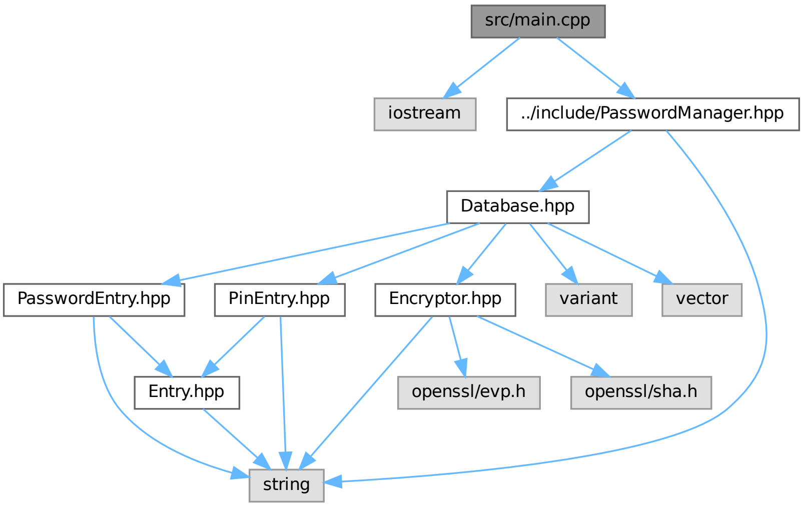
Namespaces

- namespace wk

11.18 src/main.cpp File Reference

```
#include <iostream>
#include "../include/PasswordManager.hpp"
```

Include dependency graph for main.cpp:



Functions

- int main ()

11.18.1 Function Documentation

11.18.1.1 main()

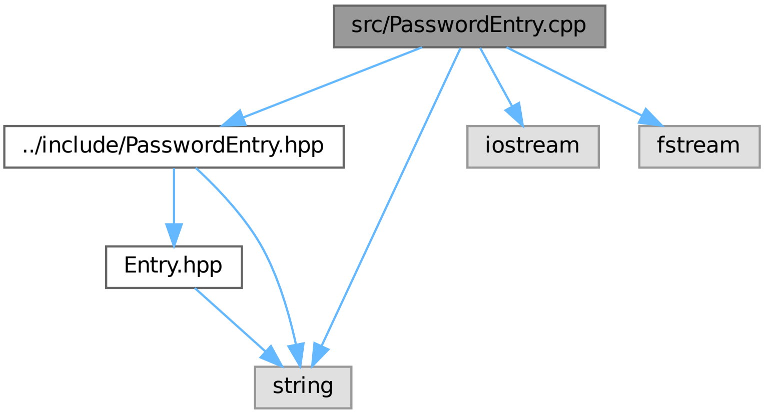
```
int main ( )
```

References wk::PasswordManager::run().

11.19 src/PasswordEntry.cpp File Reference

```
#include "../include/PasswordEntry.hpp"
#include <string>
#include <iostream>
```

```
#include <fstream>
```
Include dependency graph for PasswordEntry.cpp:

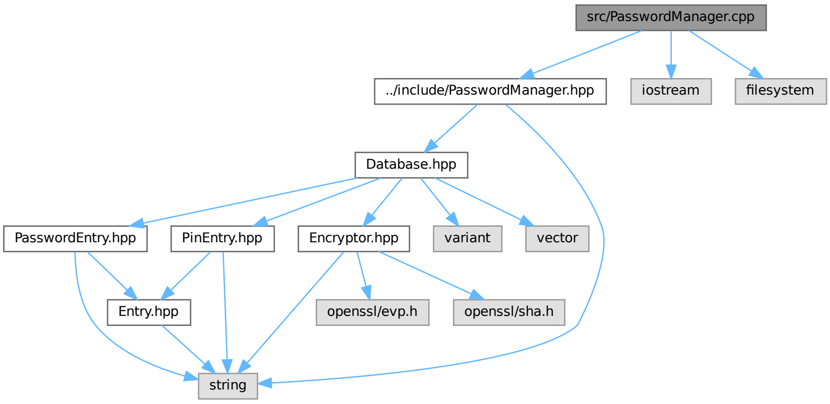


Namespaces

- namespace wk

11.20 src/PasswordManager.cpp File Reference

```
#include "../include/PasswordManager.hpp"
#include <iostream>
#include <filesystem>
```
Include dependency graph for PasswordManager.cpp:



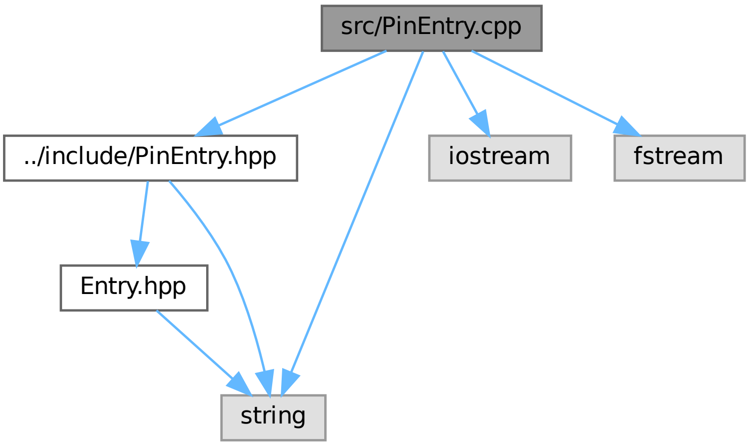
Namespaces

- namespace wk

11.21 src/PinEntry.cpp File Reference

```
#include "../include/PinEntry.hpp"
#include <string>
#include <iostream>
#include <fstream>
```

Include dependency graph for PinEntry.cpp:



Namespaces

- namespace wk

Index