PasswordProtectionProgram

Generated by Doxygen 1.8.13

Contents

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

Chapter 2

Class Index

2.1 Class List

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

Chapter 3

File Index

3.1 File List

Here is a list of all documented files with brief descriptions:

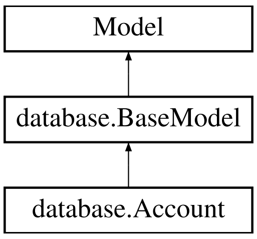
Chapter 4

Class Documentation

4.1 database.Account Class Reference

SQLite table to store passwords.

Inheritance diagram for database.Account:



Static Public Attributes

- **ID** = peewee.PrimaryKeyField()
- **AccName** = peewee.CharField()
- **AccType** = peewee.CharField()
- **UserName** = peewee.CharField(null=True)

4.1.1 Detailed Description

SQLite table to store passwords.

Use peewee orm library to create a table class that stores accounts

Parameters

AccID	Account ID and Primary Key
AccType	Type of Account used
UserName	Account Username

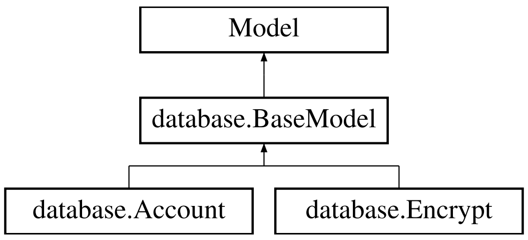
The documentation for this class was generated from the following file:

- database.py

4.2 database.BaseModel Class Reference

Base Model for database connection All other Tables will connect automatically to our database.

Inheritance diagram for database.BaseModel:



Classes

- class Meta

4.2.1 Detailed Description

Base Model for database connection All other Tables will connect automatically to our database.

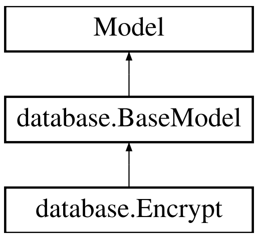
The documentation for this class was generated from the following file:

- database.py

4.3 database.Encrypt Class Reference

SQLite table to store hash keys and hash values.

Inheritance diagram for database.Encrypt:



Static Public Attributes

- **ID** = peewee.ForeignKeyField(Account, to_field='ID', primary_key=True, on_delete='CASCADE')
- **HashVal** = peewee.CharField()
- **HashKey** = peewee.FixedCharField(10)

4.3.1 Detailed Description

SQLite table to store hash keys and hash values.

use peewee orm library to create a table class that stores hash values and hash keys in a database. This table is not accessabile via the application.

Parameters

Eid	Encrypted Password ID and Foreign key from Account ID
HashVal	Hashed value of Password
HashKey	Key to Decrypt Password

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

- database.py

4.4 database.BaseModel.Meta Class Reference

Static Public Attributes

- **database** = db

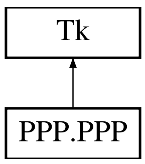
The documentation for this class was generated from the following file:

- database.py

4.5 PPP.PPP Class Reference

An ADT that represents the GUI.

Inheritance diagram for PPP.PPP:



Public Member Functions

- def __init__ (self, args)

  PPP constructor.
- def showLogIn (self, kwargs)

  Log In Screen.
- def matchPassword (self, frame, kwargs)

  Match Password.
- def showCreateMP (self, args)

  Master Password Creation Screen.
- def checkMP (self, frame, kwargs)

  Check Master Password.
- def showEntry (self, frame, detailFrame)

  Show entry.
- def showPWPage (self, args)

  Password Management Screen.
- def addEntry (self, scrollFrame, canvas, detailFrame, pwd)

  Add entry to database and display in scrollbar frame.
- def viewDetails (self, idnum, frame)

  Displays details of entry.

Public Attributes

- **accounts**
- **img**

4.5.1 Detailed Description

An ADT that represents the GUI.

4.5.2 Constructor & Destructor Documentation

4.5.2.1 __init__()

```
def PPP.PPP.__init__ (
              self,
              args )
```

PPP constructor.

Initializes a PPP GUI object using a variable argument list

Parameters

$*$*args*	A variable argument list that contains information that should be dsplayed in GUI

4.5.3 Member Function Documentation

4.5.3.1 addEntry()

```
def PPP.PPP.addEntry (
              self,
              scrollFrame,
              canvas,
              detailFrame,
              pwd )
```

Add entry to database and display in scrollbar frame.

Adds entry to the database and the display

Parameters

scrollFrame	The frame on the left which displays the entries as buttons
detailFrame	The frame on the right which displays details of each entry
∗*pwd*	Variable list of entries from the user when he/she adds an entry

4.5.3.2 checkMP()

```
def PPP.PPP.checkMP (
              self,
              frame,
              kwargs )
```

Check Master Password.

Checks if password meets criteria for master password creation

Parameters

frame	The frame which called the method, so it can be updated to show something else upon entering a satisfatory password
∗∗*kwargs*	A variable argument list, in this case takes the label that tells you if incorrect password and entry from user

4.5.3.3 matchPassword()

```
def PPP.PPP.matchPassword (
              self,
              frame,
              kwargs )
```

Match Password.

Checks if password matches master password stored in database

Parameters

frame	The frame which called the method, so it can be updated to show something else upon entering the correct password
∗∗*kwargs*	A variable argument list, in this case takes the label that tells you if incorrect password and entry from user

4.5.3.4 showCreateMP()

```
def PPP.PPP.showCreateMP (
              self,
              args )
```

Master Password Creation Screen.

Displays the master password creation frame

Parameters

∗*kwargs*	A variable argument list to provide extra functionality to the frame

4.5.3.5 showEntry()

```
def PPP.PPP.showEntry (
              self,
              frame,
              detailFrame )
```

Show entry.

Shows entries that already exists as a button on the left scrolling frame

Parameters

frame	The frame that displays button
detailFrame	The frame that will show further details if button on left frame is clicked

4.5.3.6 showLogIn()

```
def PPP.PPP.showLogIn (
              self,
              kwargs )
```

Log In Screen.

Displays the log in frame

Parameters

∗kwargs	A variable argument list to provide extra functionality to the frame

4.5.3.7 showPWPage()

```
def PPP.PPP.showPWPage (
              self,
              args )
```

Password Management Screen.

Where user can add and view account information

Parameters

∗args	A variable argument list

4.5.3.8 viewDetails()

```
def PPP.PPP.viewDetails (
              self,
              idnum,
              frame )
```

Displays details of entry.

Displays details of entry (type, name, username, password), called when button for entry is clicked

Parameters

idnum	The id number of the entry that was clicked
frame	The frame in which to display the details on

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

- PPP.py

Chapter 5

File Documentation

5.1 database.py File Reference

PPP_database

Classes

- class database.BaseModel

  Base Model for database connection All other Tables will connect automatically to our database.
- class database.BaseModel.Meta
- class database.Account

  SQLite table to store passwords.
- class database.Encrypt

  SQLite table to store hash keys and hash values.

Functions

- def database.CreateTables ()

  Instantiate new empty tables.
- def database.DropTables ()

  Delete tables.
- def database.Insert (Id, N, T, U, Hv, Hk)

  Insert new Account Instance and Encrypt Instance.
- def database.GetId (id_)

  search tables with AccId
- def database.GetT (Atype)

  Get Table Rows with Account Type.
- def database.GetN (Aname)

  Get Table Row with Account name.
- def database.Delete (id_)

  Delete Table Row with ID.
- def database.UpdateU (Aid, U)

  Update Table Row with ID.
- def database.UpdateP (Aid, Hv)

  Update Password with ID.

Variables

- **database.db** = peewee.SqliteDatabase('pppDatabase.db')

5.1.1 Detailed Description

PPP_database

Author

Joseph Lu, luy89

Date

20/10/2017

5.1.2 Function Documentation

5.1.2.1 CreateTables()

```
def database.CreateTables ( )
```

Instantiate new empty tables.

Encrypt Table should also be reset when Account is reset

5.1.2.2 Delete()

```
def database.Delete (
              id_ )
```

Delete Table Row with ID.

Parameters

id	Account ID

5.1.2.3 DropTables()

```
def database.DropTables ( )
```

Delete tables.

Encrypt Table should also be reset when Account is reset

5.1.2.4 GetId()

```
def database.GetId (
              id_ )
```

search tables with AccId

Parameters

Id	Account Id

5.1.2.5 GetN()

```
def database.GetN (
              Aname )
```

Get Table Row with Account name.

Parameters

Aname	Account Name

5.1.2.6 GetT()

```
def database.GetT (
              Atype )
```

Get Table Rows with Account Type.

Parameters

Atype	Account type

5.1.2.7 Insert()

```
def database.Insert (
              Id,
              N,
              T,
              U,
              Hv,
              Hk )
```

Insert new Account Instance and Encrypt Instance.

Parameters

N	Account Name
T	Account Type
U	username
Hv	Hash Value
Hk	Hash Key

5.1.2.8 UpdateP()

```
def database.UpdateP (
              Aid,
              Hv )
```

Update Password with ID.

Parameters

Aid	Account Id (Encrypted ID)
Hv	new Hash Value

5.1.2.9 UpdateU()

```
def database.UpdateU (
              Aid,
              U )
```

Update Table Row with ID.

Parameters

Aid	Account Id
U	new Username
Hv	new Hash Value

5.2 Encrypt.py File Reference

Password Encryption

Functions

- def Encrypt.generKey ()

This function generates a unique) key (for encoding user passwords) using Python's Fernet library.

- def Encrypt.cryptEncode (key, passw)

  This function uses a key to encrypt an input string.
- def Encrypt.cryptDecode (key, encrypted)

  This function uses the saved key to decrypt the encrypted user password stored in the database.

5.2.1 Detailed Description

Password Encryption

Author

Shabana Dhayananth

Date

October 15, 2017

5.2.2 Function Documentation

5.2.2.1 cryptDecode()

```
def Encrypt.cryptDecode (
              key,
              encrypted )
```

This function uses the saved key to decrypt the encrypted user password stored in the database.

Parameters

key,encrypted	refer to the key that was used to encrpt the password and the encrypted password

Returns

decrypted password

5.2.2.2 cryptEncode()

```
def Encrypt.cryptEncode (
              key,
              passw )
```

This function uses a key to encrypt an input string.

Fernet uses symmetric encryption on the input key

Parameters

key,passw	refer to the key to be used to encypt and the password to be encrypted

Returns

encrypted password

5.2.2.3 generKey()

```
def Encrypt.generKey ( )
```

This function generates a unique) key (for encoding user passwords) using Python's Fernet library.

key derived from a string that is run through the kdf (key derivation function)

Returns

key that will be used to encode the user passwords (32 bytes)

5.3 GenPassword.py File Reference

Generate Random Password

Functions

- def GenPassword.genPass ()
  This function generates a random password consisting of upper case, lower case alphanumeric characters.
- def GenPassword.genPassCrypt ()
  This function generates a random password consisting of upper case, lower case alphanumeric characters.

5.3.1 Detailed Description

Generate Random Password

Author

Shabana Dhayananth

Date

October 27, 2017

5.3.2 Function Documentation

5.3.2.1 genPass()

```
def GenPassword.genPass ( )
```

This function generates a random password consisting of upper case, lower case alphanumeric characters.

default random number generator's sequences can be reproduced, in case SystemRandom() is not available on user system

Returns

random password consisting of 8 characters

5.3.2.2 genPassCrypt()

```
def GenPassword.genPassCrypt ( )
```

This function generates a random password consisting of upper case, lower case alphanumeric characters.

Same as genPass() but uses SystemRandom() to generate random numbers so sequences are not reproducible

Returns

random password consisting of 8 characters

5.4 PPP.py File Reference

The graphical user interface for a password manager.

Classes

- class PPP.PPP

  An ADT that represents the GUI.

Variables

- string **PPP.BGC** = "#cccccc"
- tuple **PPP.LARGE** = ("Helvetica", 16)
- string **PPP.BG** = "#383A39"
- string **PPP.FG** = "#A1DBCD"
- **PPP.app** = PPP()

5.4.1 Detailed Description

The graphical user interface for a password manager.

Author

Suhavi Sandhu

Date

November 10, 2017

5.5 PWChecking.py File Reference

Check Passwords

Functions

- def PWChecking.checkMP (password)

  Checks master password at creation.
- def PWChecking.checkLogIn (entered, actual)

  Checks master password at login.

5.5.1 Detailed Description

Check Passwords

Author

Suhavi Sandhu

Date

November 10, 2017

5.5.2 Function Documentation

5.5.2.1 checkLogIn()

```
def PWChecking.checkLogIn (
              entered,
              actual )
```

Checks master password at login.

Verifies that the entered password matches the actual

Parameters

entered	The password entered by the user
actual	The real master password

5.5.2.2 checkMP()

```
def PWChecking.checkMP (
              password )
```

Checks master password at creation.

Verifies that the password meets criteria

Parameters

password	The password that is being checked

Index