INDEX

* Documentation…
* Algorithms………..
* Flowcharts………..
* Code………………….
* Output……………….

ACKNOWLEDGEMENT

I would like to express my gratitude towards my teacher Mrs. Rashmi who has given us this wonderful opportunity and in the process a wonderful experience, by making me work on a project based on the topic “Personalized Music Library”, which has not only helped me gain knowledge but also helped boost my confidence in this subject.

I would also like to thank my parents and friends who helped me in the due course of this project, each in their own way.

OBJECTIVE

This program is written for users who would like to create and administrate their own personalized music library. It would allow users to add or delete music. Each library of the users are securely separated from that of the others. The program is simple and user friendly and will guide users through the program without any hassle.

THE

DOCUMENTATION

* Modules

1. Os:- This module provides a unified interface to a number of operating system functions.
2. Pickle:- This module implements a fundamental but powerful algorithm for serializing and de-serializing a Python object structure.
3. Tabulate:- This module is used to represent data in the form of tables.
4. Pyglet:- This module is used to play music by forming a queue of the songs.
5. Tkinter:- This module can be used to form buttons or any graphical user interface object.

* CLASSES

1. Registration:- This class is used for registration and for logging in.
2. Music:- This is the important class as it is used for entering data about the songs to form the library.

* FILES

1. User.dat:- Binary files for different users along with their details.
2. Users.txt:- Text file containing the different users who have registered.
3. P1.dat:- Binary file containing data in regard with the music library.
4. Temp.dat:-

* FUNCTIONS

1. Register():- Enter your registration details to create an account.
2. Login:- Enter your registration details to login into your account.
3. Getdata:- Enter the details about the song.
4. Getname:- Enter new name of the song
5. Getartist:- Enter new name of the artist.
6. Getalbum:- Enter new name of the album.
7. Outdata():- Returns the data.
8. Append:- Add music to your library.
9. Search:- Search for a particular song.
10. Display:-Displays the library you have made so far.
11. Delete:-Used to delete a song.
12. Update:-Used to update the details of a song added.
13. Sort: Arrangement of song in an order of desire.
14. Play:- Used to play a song.

ALGORITHMS

* Register()

1. Start
2. Take input of username and password
3. Open ‘Users.txt’ and write the contents of the user in file.
4. Print “Registration Successful”.
5. Stop.

* Login()

1. Start
2. Open ‘Users.txt’.
3. Take input of username and password
4. If username not present, take contents and print “Login successful.”
5. If username already present, print “Login incorrect.”
6. Stop.

* Getdata()

1. Start
2. Take input of name of song, name of artist and name of album.
3. Stop.

* Getname()

1. Start
2. Take input of new name of song.
3. Stop

* Getartist()

1. Start
2. Take input of new name of artist.
3. Stop

* Getalbum()

1. Start
2. Take input of new name of album.
3. Stop

* Outdata()

1. Start
2. Open binary file of user
3. Return the data.
4. Stop

* Append()

1. Start
2. Open the file ‘p1.dat’.
3. Take input of the number of songs user wishes to add.
4. Dumps all the data from the user in to the file ‘p1.dat’.
5. Stop

* Update()

1. Start
2. Take the choice of the user to update song name, artist name or album name
3. Open the file ‘p1.dat’ and another file ‘temp.dat’
4. Take the input of user for updating, according to the user choice made
5. Close both the files
6. Removes the file ‘p1.dat’ and renames the file ‘temp.dat’ to ‘p1.dat’
7. Stop

* Delete()

1. Start
2. Open the file ‘p1.dat’ and another file ‘temp.dat’
3. Take input of the song to be deleted
4. Close both the files
5. Remove the file ‘p1.dat’ and rename the file ‘temp.dat’ to ‘p1.dat’.
6. Stop

* Search()

1. Start
2. Open the file ‘p1.dat’
3. Take the choice of the user to search using song name or artist name
4. If search found, exit loop & takes input from user on whether they would like to perform additional operations
5. At users request call either update or delete or play functions
6. If play(), check for song in list and take input from user on song to play
7. Creates a window in tk to play songs

* Play\_it()

1. Start
2. Button to play the Song
3. Stop

* Pause\_it()

1. Start
2. Button to pause the playing song
3. Stop
4. Close the file ‘p1.dat’
5. Stop

* Display()

1. Start
2. Open the file ‘p1.dat’
3. Display the contents of file ‘p1.dat’ in a table
4. Close the file ‘p1.dat’
5. Stop

* Sort()

1. Start
2. Open the file ‘p1.dat’
3. Take the choice of the user to sort using song name, artist name or album name
4. Sort in ascending order
5. Close the file ‘p1.dat’
6. Stop

* Play()

1. Start
2. Open the file ‘p1.dat’
3. Enter the format of the song to be played in
4. Opens a Window in ‘tk’ with relevant buttons
5. Close the file ‘p1.dat’
6. Plays the songs in playlist

* Play\_Song()

1. Start
2. Plays the Song
3. Stop

* Pause\_Song()

1. Start
2. Pauses the Song
3. Stop

* Next\_Up()

1. Start
2. Moves to the next song in the Playlist
3. Stop
4. Stop

FLOWCHART

1. Register ()

Print Registration Successful

Take input of Username and Password & write it in the file

2. Login ()

If Present

Print “Login Unsuccessful”

If Not Present

Print “Login Successful”

Check whether input is present in file

Open file “Users” Take input of Username and Password

3. Getdata ()

Input Song Name, Artist Name, Album Name

4. Getname ()

Input Song name

5. Getartist ()

Input Artist name

6. Getalbum ()

Input Artist name

7. Outdata ()

Open file and return the Data

8. Append ()

Open the file and Take input of song information and dump data in the file

9. Update ()

Open files “P1.dat” and “temp.dat”. Take input for updating according to users choice. Close files and remove “p1.dat” and rename “temp.dat” to “p1.dat”

Take choice of user to update song name, artist name or album name

10. Delete ()

A

Open “P1.dat” and “temp.dat”. Take input of song to be deleted. Close both files, remove “P1.dat” and rename “temp.dat” to “P1.dat”

A