MySQL with Python - Project

Lifechoices Online CLI

Communication Plan

Project Name: Lifechoices Online	**Beginning Date:** 04/05/2020
Project Managers: Thapelo & Godwin	**Completion Date:** 13/05/2020
Plan Owner: Thapelo	**Alternative:** Godwin

Planning

Project objective and key message points (high level):

- People visiting lifechoices premises need to be able to register their details on Life Choices online before entering the premises.
- People need to be able to sign-in using the Lifechoices online management system.
- People need to be able to sign-out (when leaving premises) using the Lifechoices online management system.
- An admin should at a minimum be able to check who has signed-in at any point in time.
- An admin should at a minimum be able to check who has signed-out at any point in time.
- The admin should be able to add and remove users at any point in time.

Functional Requirements

The requirements are ordered in priority, from (5 - 1) with 5 being the highest and 1 the lowest.

Requirement:	Priority Ranking
1. Register user details.	**5**
2. Ability to sign-in and sign-out.	**5**
3. Upgrade/downgrade user privileges.	**2**
4. Show people who have signed-in for the day.	**2**
5. Show people who have signed-out for the day.	**2**

6. Admin add and remove users in the system.	**1**

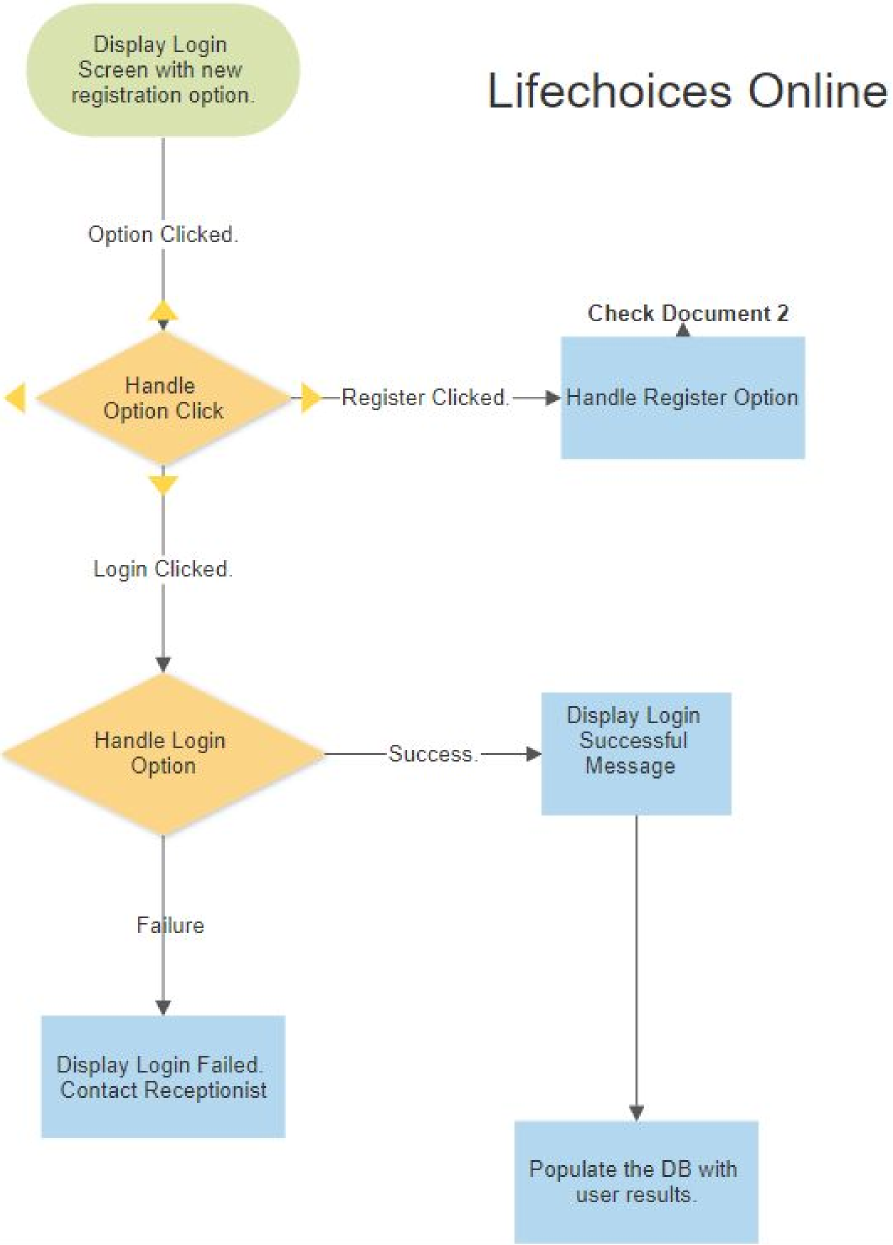
Introduction

Purpose of this document

This document formally describes the user requirements for the conceptual model of an application called **lifechoices online**. It is intended to be used as a mock-up solution to get rid of the paperwork and files that are used at the reception. Please note this is the minimal work that can be done quickly for the sake of the Python module assessment task. The full application is beyond the scope of this document.

System Context

Provided below is a simplified flow-diagram for **lifechoices online**.

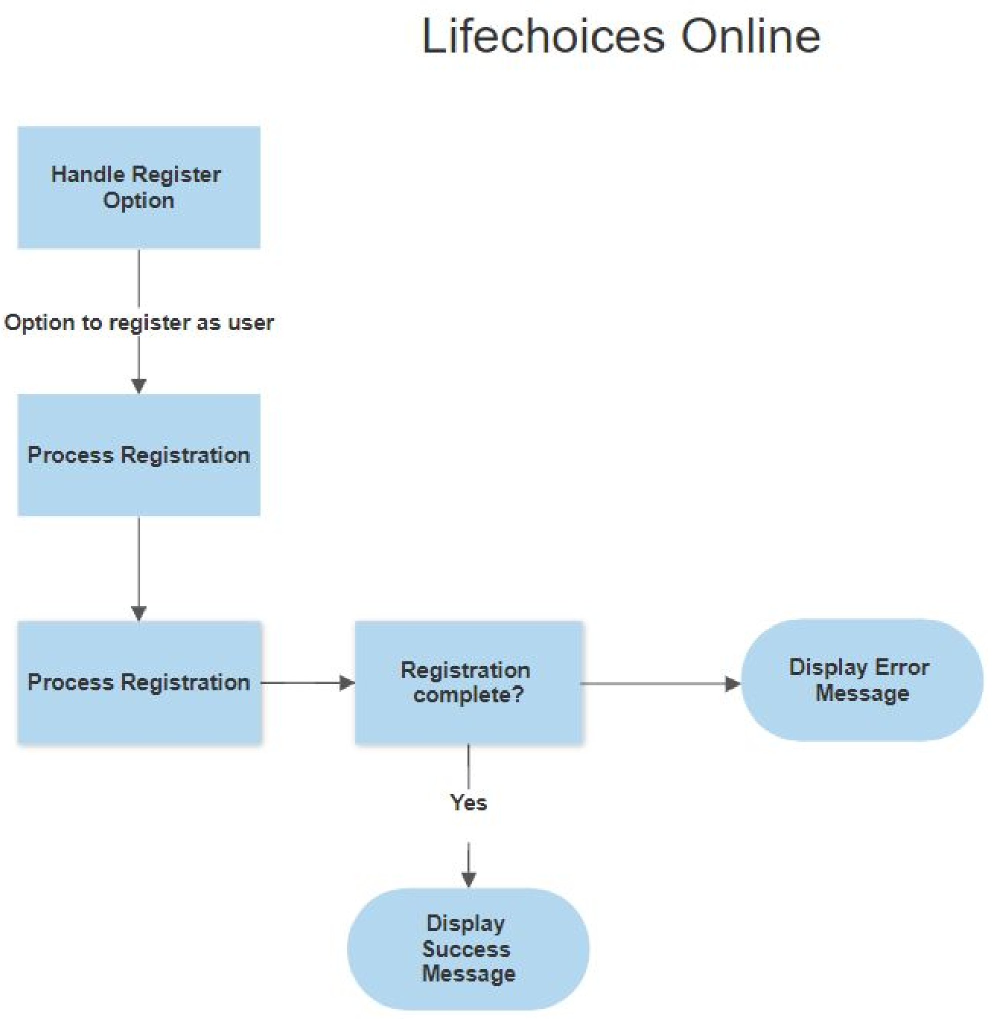

Lifechoices Online
Display Login Screen with new registration option.
Option Clicked.
Handle Option Click
Register Clicked.
Handle Register Option
Check Document 2
Login Clicked.
Handle Login Option
Success.
Display Login Successful Message
Failure
Display Login Failed. Contact Receptionist
Populate the DB with user results.

To summarise the above flow chart(Algorithm summary):

- When your program starts, it should display the login screen with the option for new registration. This is the case when new employees or visitors arrive at the premises and would like to enter the building.(Consider this the main screen)
- If the login option is clicked (selected), the user should be presented with the option asking them to enter their full_name, username and password.
- If the login is successful, the user should be presented with the “Login Successful! Enjoy Your Day” message.
- The database table to keep track of who has logged-in or logged-out needs to be populated each time a user logs-in or out of the system. Remember there are fire drills at lifechoices, fire-marshals need to know who is or not inside the building.

- If the login is unsuccessful, the user should be presented with a “Login Unsuccessful.” message and give the user an option to return to the main menu.

Document 2 option as shown in the flow-chart above.



To summarise the above flow chart(Algorithm summary):

- When you are on the main screen, by pressing the “**a**” key, for admin, you should be presented with a new screen to login.
- After the successful login of the admin user, they should be presented with a screen to add a new user or delete a user.
- If the user was successfully added in the system the message “Successfully Added User” should be added to the screen and re-display the same screen.
- If the user was not successfully added to the system, the message “Failed To Add The User” should be should to the user and re-display the same menu screen again.

- Add the **quit** option to the current screen which will take you back to the **main** screen.
- Any deleted/removed user in the system can know longer login to the system again. They will have to be registered again using the new registration option on the main screen. Deleting users from the system is beyond the scope of this project.

Sample Database Design Diagram

Below is a sample database structure that you could have in your system. You can add extra features for some extra marks.

But for you to start, you need to create the database using whatever is your preference, **workbench** or **mysql command client**. Find attached the sample lifechoices online database mysql dump.

But for illustration, this is what you need to minimally create your **lifechoices** online database application.

```
CREATE DATABASE  IF NOT EXISTS `lifechoicesonline`;
USE `lifechoicesonline`;

DROP TABLE IF EXISTS `users`;
CREATE TABLE `users` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `full_name` varchar(60) DEFAULT NULL,
  `username` varchar(50) DEFAULT NULL,
  `password` varchar(20) DEFAULT NULL,
  PRIMARY KEY (`id);
```

- Create a database **table** which will link normal users(as the above table) who have logged-in or logout in the system. Make sure you get the **date-time** for when the user log's in and logs's out correctly. This can be used by the admin to check who is **inside the building or not**.

- You can create any extra tables that you feel will be ideal for your application to make it efficient.

What you need to submit for this project.

- Make an sql dump of your sql database project and save it in your directory of your choice.
- Create a zip folder of your PyCharm project with all the files needed inside the zip folder.
- Insert your mysql dump sql file inside your project above.
- Submit the project on the google classroom as before.

End of Project