***Documentación Técnica de la aplicación de ITESO PARKING***

**Modulo Log IN :**

**Contexto de nuestra base de datos:**

**Graphical user interface, text, application, email

Description automatically generated**

Para obtener el usuario utilizamos el key de cada uno de los childs de Usuarios y para las contraseñas utilizamos el valor de password.

**Función Log In()**

En este pantallazo podemos observar una de las funciones más importantes dentro de nuestro módulo de Log IN, es la función de LogIn(). En esta función nosotros lo que hacemos es evaluar las entradas tanto de Usuario como contraseña. Estas las comparamos con las credenciales dentro del apartado de Usuarios en la base de Datos. En caso de que las comparaciones de los IF’s sean correctas imprimimos un log exitoso y actualizamos el valor de active.usuario, es decir que ahora asignamos que el usuario activo es el que se acaba de logear.

Text

Description automatically generated

Aquí esta otro función entry() que sirve para el establecimiento de la parte visual por medio de Tk Inter. Solo es para evitar la repetición de Código

Text

Description automatically generated

**Modulo Sign Up :**

La primera función que revisaremos del Módulo de Sign Up, la primera función que revisaremos será la función de Validuser(), en esta función la funcionalidad es que compara las características del log que recibe de Tk Inter y lo compara con las condiciones que tiene en sus ifs. Una de las funciones más importantes es que compara el hecho de que un nombre de usuario no exista ya previamente.

Text

Description automatically generated

Text

Description automatically generatedEn esta función lo que hacemos es comprobar que la contraseña que se proveyó por parte del log de contraseña tenga las características necesarias para poder ser subida. En caso de resultar insatisfactoria la comprobación lo que hace es modificar la variable de un label para que aparezca según que error hace falta.

En esta funciones lo que buscamos es dividir partes del código que necesitábamos recurrentemente con el propósito de facilitar su lectura y su implementación. La función más importante de estas 3 es la de Register() donde hace una actualización a la base de datos y sube las credenciales recién subidas, en caso de ser llamada por una de las funciones que arriba consultamos.

Text

Description automatically generated

**Modulo Estacionamiento:**

**Contexto de la estructura de la base de datos:**

Nuestra base de datos lo que tiene como valores de estado son los paths a los archivos respectivamente de su estado

Sin información = bgris.ppm

Ocupado = brojo.ppm

Reservado = bamarillo.ppm

Libre = bverde.ppm

Esto nos facilito la manipulación de las variables de fotografía ya que solo necesitábamos obtener el valor del json de nuestra base de datos para poder obtener el path necesario del display de la foto según en que estado queramos que se encuentre.

Graphical user interface, application

Description automatically generated

**Funcionamiento de las asignación de imágenes según la variable de path encontrada en la base de datos**

Lo primero que hacemos es definir un Frame() individual para cada botón. Dentro de estos frames vamos a montar cada uno de los botones respectivamente.



Después lo que definimos es las variables imga1-amga15 estas variables de tipo PhotoImage lo que hacen es montar a una variable de Tkinter PhotoImage la imagen respectiva según el path que reciban de la base de datos, esto según cada uno.

A picture containing text

Description automatically generated

Una vez que nuestros PhotoImage se encuentran ya definidos lo que tenemos que hacer es invocar los botones a1-a15. Estos botones hacen referencia a su respectivo frame para montar el botón de cada uno. Se montan con su Imga respectivo y se le atribuye la etiqueta command en la que se le da su función respectiva por botón.

Graphical user interface, text

Description automatically generated

**Esta es la explicación de todas las funciones fbtn1 hasta fbtn15:**

Como utilizamos 15 botones diferentes y no pudimos agregarle alguna referencia a los botones que daban click para poder obtener cual era el botón seleccionado, utilizamos 15 funciones diferentes que se asignan a cada botón individualmente y respectivamente

En estas funciones que se activan al darle click a los botones, primero realizan el establecimiento de 15 variables globales que son booleanos que determinan cual es el botón que fue presionado por última vez. En cada uno de estas funciones lo que hace es asignar como true el booleano asignado a la función 1 es decir boola1 y negar todos los demás esto lo que nos genera es que independientemente la cantidad de botones que presione el usuario antes de presionar el botón de reservar, siempre tendrá como true únicamente el ultimo botón presionado.

Otra funcionabilidad de estas funciones es que asignan el valor de “A1” - “A15” a la variable dinámica que está asociada al lugar seleccionado, de esta forma si usted hace click en los botones le dice cuál es el botón que acaba de presionar, siendo este desde A1 hasta A15.

Por último realiza un chequeo con la variable atribuida al valor del Cajón de estacionamiento si esta es diferente a “bgris.ppm” manipula la variable state del botón respectivamente para deshabilitarlo.

Esta utilización de las 15 variables booleanas nos sirven para nuestra siguiente función checkbool()

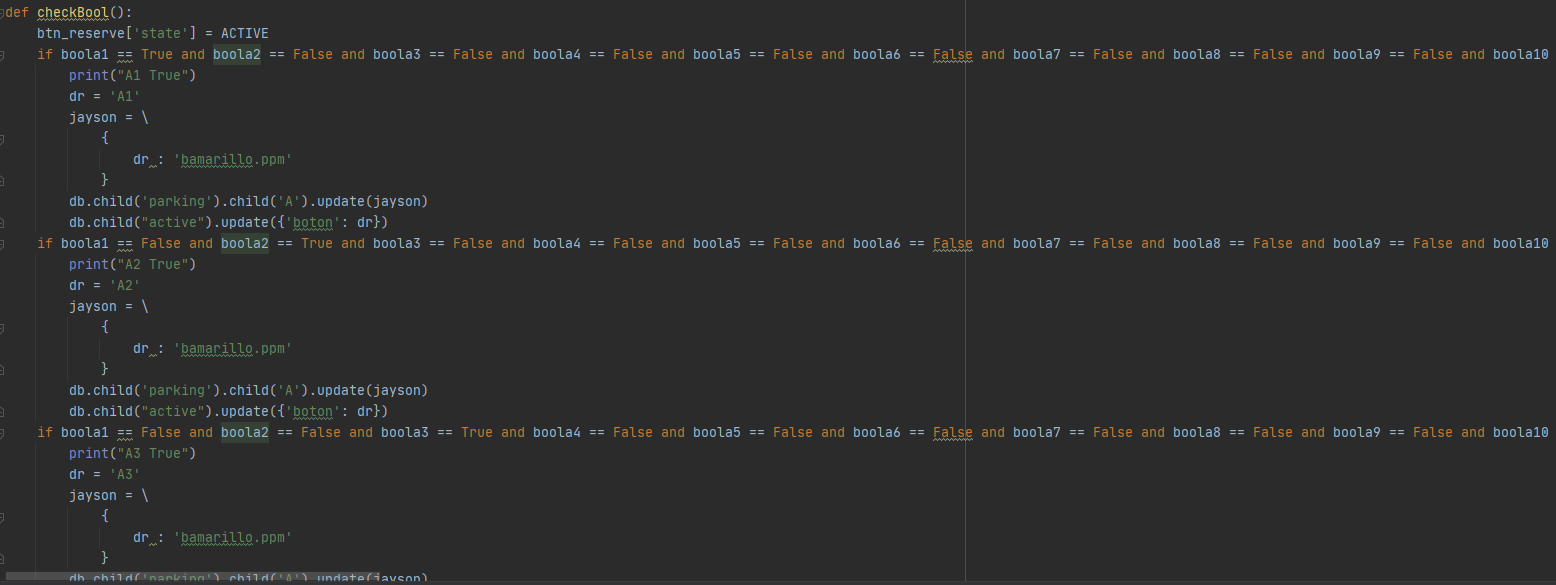
A picture containing graphical user interface

Description automatically generated

Text

Description automatically generated

**Función checkbool()**

La función de checkbool() lo que hace es que se dedica a comprobar cual es la variable de tipo boola1-boola15 que se encuentra en condición true. Esto es para saber cual fue el último botón que presionó nuestro usuario. Una vez que hace esto actualiza a la base de datos para cambiar al estado de reservado el lugar seleccionado, es decir cambia el valor a “bamarillo.ppm”. A parte de esto sube a “active.boton” el botón que se seleccionó para pasar a sala de espera el valor que después de hacer log out necesita colocar como ocupado.

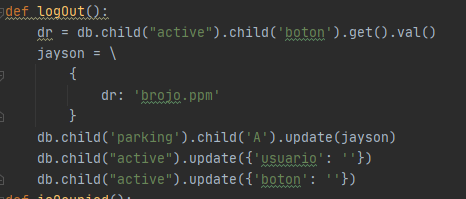
Se realizan 15 sentencias donde se compara que una y solo una de las variables booleanas esta activa.

Modulo: Sala de espera

Aquí en sala de espera utilizamos las mismas funciones y configuraciones que estacionamiento para también poder retribuir de la base de datos el estado de los lugares de estacionamiento.

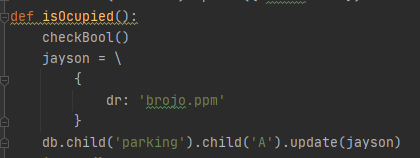
Hicimos modificaciones para adaptarla a las necesidades de la pagina así como la integración de los botones de isOcupied() y isFree().

Función logOut()



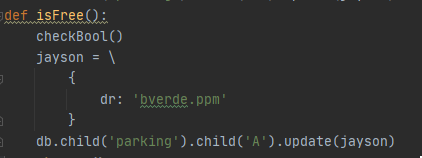
La función de logOut simplemente accede al valor del usuario activo dentro de la base de datos y cuando se invoca la función el usuario activo queda vacío y se actualiza el valor del Cajón que fue reservado a ocupado

Función isOcupied()



La función isOcupied cambia el estado del Cajón seleccionado a ocupado cambia la imagen que se muestra a un botón color rojo que indica que el Cajón está ocupado

Función isFree()



La función isFree cambia el estado del Cajón seleccionado a disponible cambia la imagen que se muestra a un botón color verde que indica que el Cajón está disponible.