Android

Registration Class

1. set_registration_info()
   Parameters: Username: string, Password: string
   Return: bool

Description: null

2. check()
   Parameters: Username: string, Password: string
   Return: bool

Description: Interacting with the database, checking if the username already exists, if so, display an error message; checking if the password is less than 6 characters, if so, display an error message.

3. save_to_dataset()
   Parameters: Username: string, Password: string
   Return: bool

Description: Database should store this info.

Login Class

1. check()
   Parameters: Username: string, Password: string
   Return: bool

Description: Interacting with the database, first check whether the username exists (already registered), and then check whether the password matches. If not, warn. If the password is incorrect more than 10 times, the account will be frozen.

CustomerInfo Class

1.
set_username()
Parameters: OldUsername: string, NewUsername: string
Return: bool

Description: Warn if the NewUsername already exists in the database.

2.
set_password()
Parameters: OldPassword: string, NewPassword: string
Return: bool

Description: For security reasons, users need to enter the old password to confirm. New password length must also be at least six characters.

3.
set_age()
Parameters: Date: string
Return: Age: string

Description: Automatically display age by setting date of birth.

4.
set_Interests()
Parameters: Interests: string
Return: bool

Description: Add, modify, and delete interests via keyboard.

ViewMap Class

1.
show_recommanded_stores()

Parameters: StoreName: string
Return: StoreNameList: list

Description: Recommend store names based on current location and recommendation system.

2.
search_stores()
Parameters: StoreName: string
Return: bool

Description: Use the keyboard to enter the store name and search for relevant information in the database.

3.
Navigate()
Parameters: StoreName: string
Return: Destination_Route

Description: Enter the destination through the keyboard and get the route information. The closest location can be dynamically matched during input.

ViewHuntedList Class

1.
update_hunted_list()
Parameters: OldHuntedList: list
Return: NewHuntedList: list

Description: Delete/reorder.

2.
show_stores_info()
Parameters: StoreName: string
Return: StoreInfo: Class StoreInfo

Description: Return an instance of the StoresInfo Class based on the store name.

StoreInfo Class

//Utility Class
StoreName: string
StoreLoaction: string
StoreDescription: string

ItemsList: list[ItemInfo]

ItemInfo Class

//Utility Class
ItemName: string
ItemPrice: float
ItemDescription: string

ItemImage: list

ViewStore Class

1.

show_stores_info() // Same as above.

2.

MarkAsHunted()
Parameters: StoreName: string, OldHuntedList: list
Return: NewHuntedList: list

Description: Adds the specified store to the HuntedList.

Feedback Class

1.

feedback_item()
Parameters: StoreName: string, ItemName: string, Username: string, comment: string, rating: int
Return: bool

Description: Evaluate specific products through parameter information.

2.

```
feedback_notification() //Similar to above
```