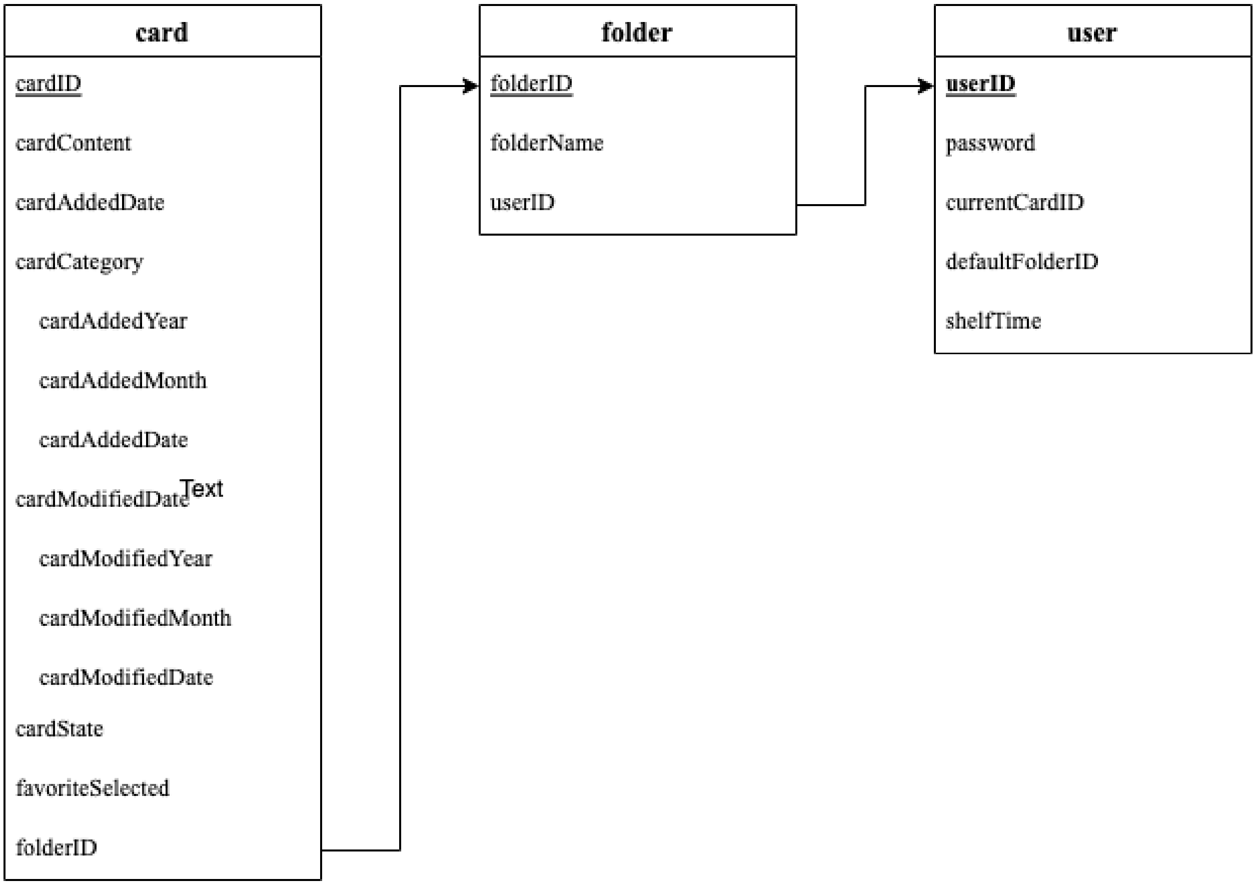
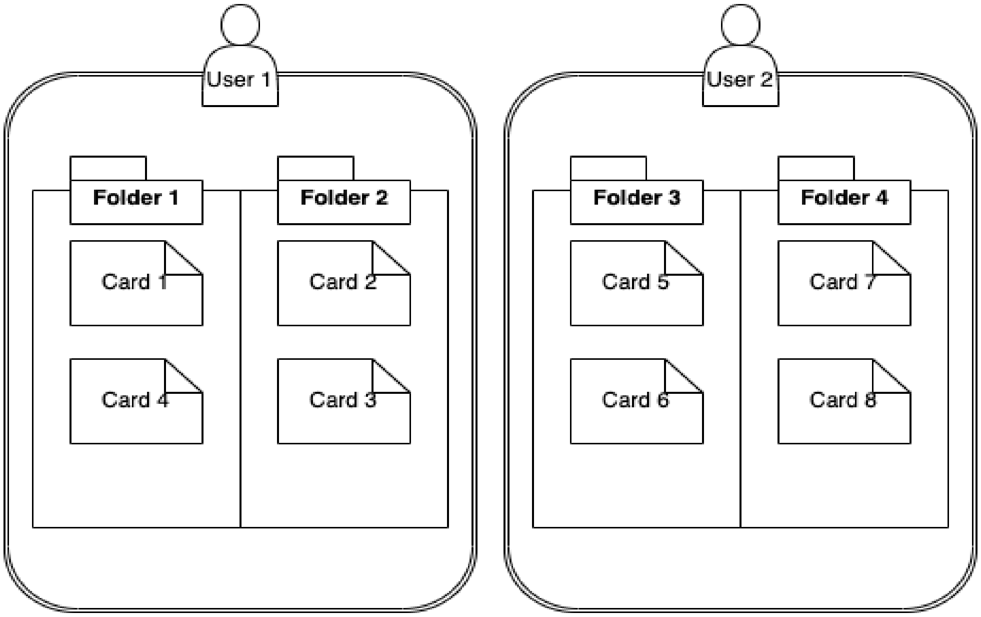
Clipboard Manager DB Access Functions

Entities:


card
cardID
cardContent
cardAddedDate
cardCategory
cardAddedYear
cardAddedMonth
cardAddedDate
cardModifiedDate
Text
cardModifiedYear
cardModifiedMonth
cardModifiedDate
cardState
favoriteSelected
folderID
folder
folderID
folderName
userID
user
userID
password
currentCardID
defaultFolderID
shelfTime

Example Datas:

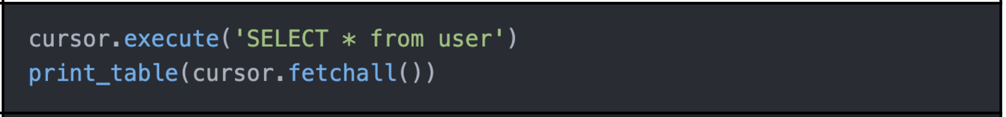


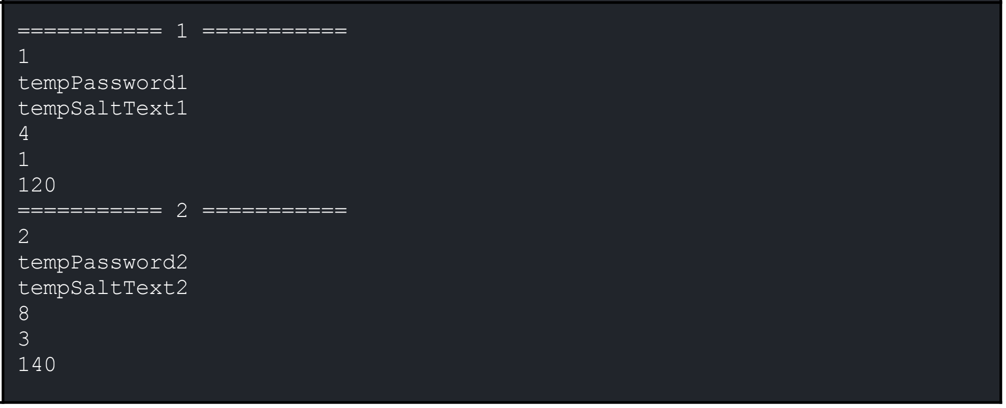
Example Cards Data Table

Functions:

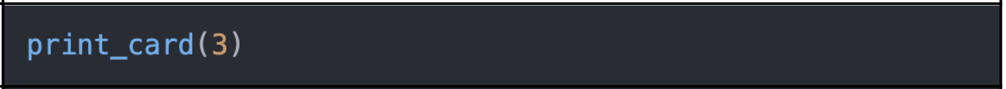
1. print_table(table_content)

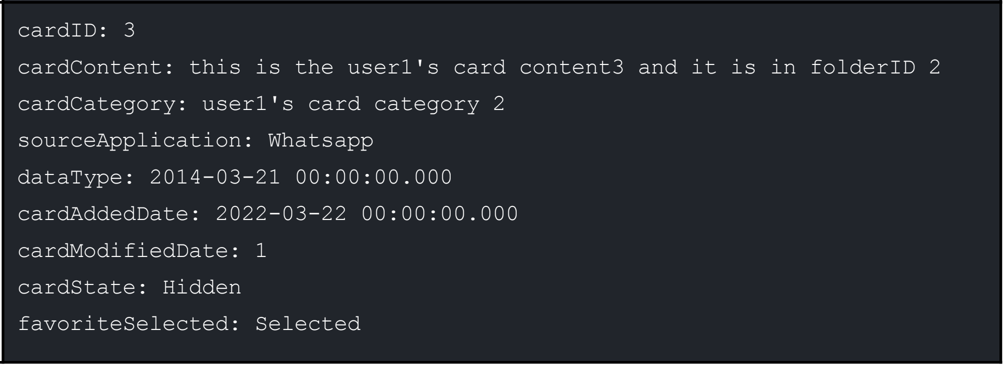
Parameters	- Table_content: iterates through the table and prints the results
Return Value	N/A
Explanation	This function is just for debugging purposes. If the programmer inserts the table value in terms of array n * n array format, the function simply *prints* the data inside that array in the console.

Example code	
	`cursor.execute('SELECT * from user')`
	`print_table(cursor.fetchall())`

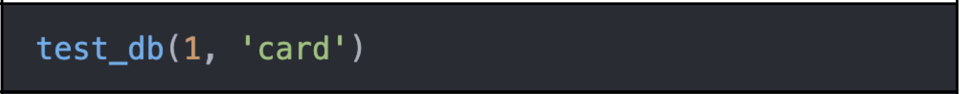
Output	
	`=========== 1 ===========`
	`1`
	`tempPassword1`
	`tempSaltText1`
	`4`
	`1`
	`120`
	`=========== 2 ===========`
	`2`
	`tempPassword2`
	`tempSaltText2`
	`8`
	`3`
	`140`

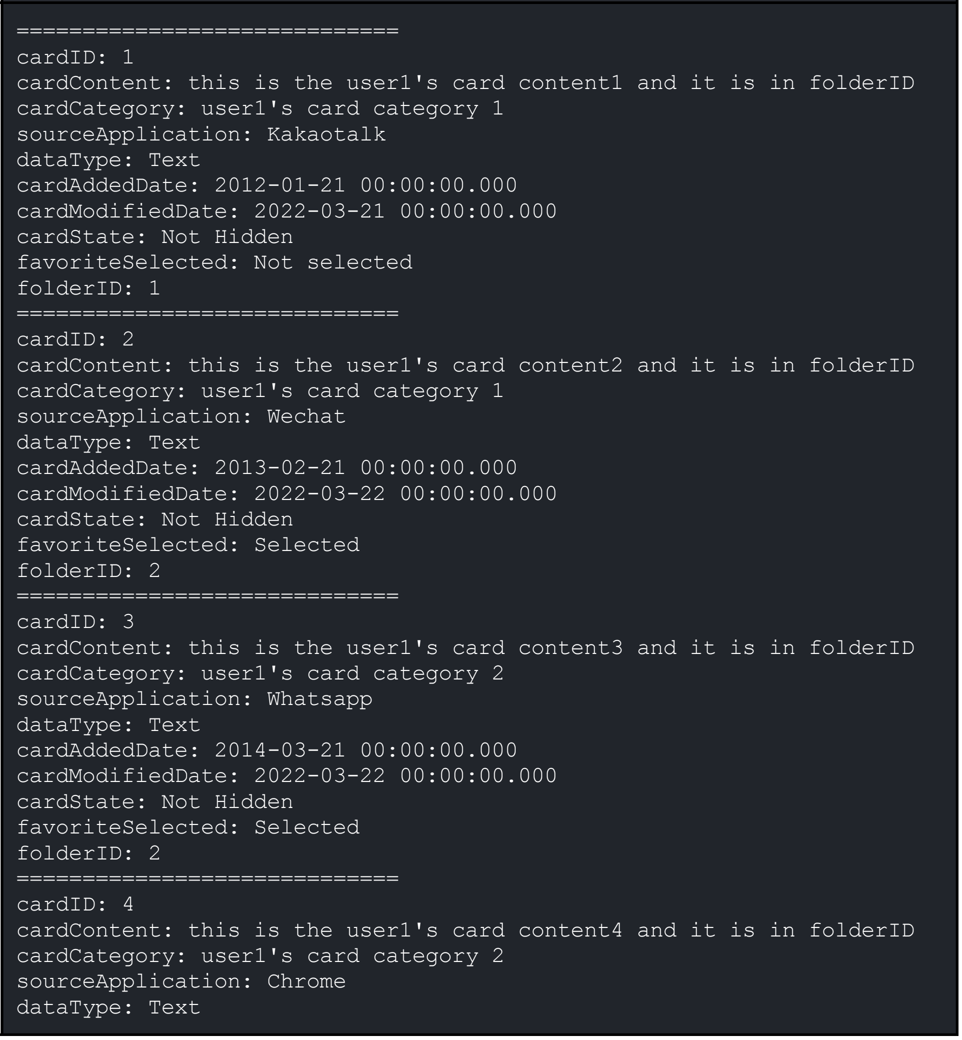
2. print_card(cardID)

Parameters	- cardID
Return Value	N/A
Explanation	This function is just for debugging purposes. If the programmer calls this function by inputting cardID, the function simply *prints* the data inside that card in the console.
Example code	 `print_card(3)`

Output	
	`cardID: 3`
	`cardContent: this is the user1's card content3 and it is in folderID 2`
	`cardCategory: user1's card category 2`
	`sourceApplication: Whatsapp`
	`dataType: 2014-03-21 00:00:00.000`
	`cardAddedDate: 2022-03-22 00:00:00.000`
	`cardModifiedDate: 1`
	`cardState: Hidden`
	`favoriteSelected: Selected`

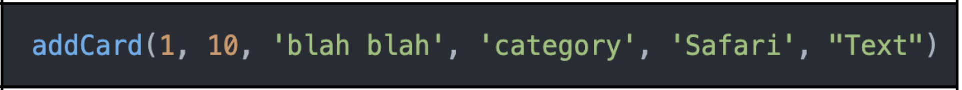
3. test_db(userID, data_type = "all")

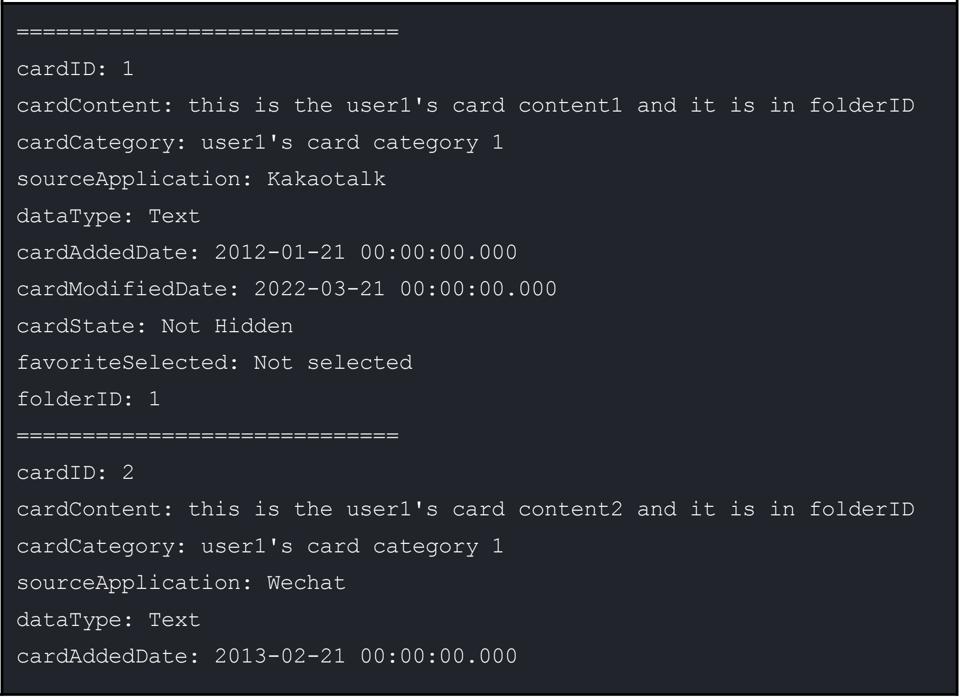
Parameters	- userID - data_type: default = “all” - “all”: prints all entities - “card”: prints all the card entities - “folder”: prints the folder entities - “user”: prints all the user entities
Return Value	N/A
Explanation	This function is just for debugging purposes. If the programmer calls the function, the function prints all the information that fits the parameter and is currently stored in the database along with the attribute name (ex. cardAddedDate: 2015-04-21 00:00:00.000)
Example code	 `test_db(1, 'card')`

Output	
	`==============================`
	`cardID: 1`
	`cardContent: this is the user1's card content1 and it is in folderID`
	`cardCategory: user1's card category 1`
	`sourceApplication: Kakaotalk`
	`dataType: Text`
	`cardAddedDate: 2012-01-21 00:00:00.000`
	`cardModifiedDate: 2022-03-21 00:00:00.000`
	`cardState: Not Hidden`
	`favoriteSelected: Not selected`
	`folderID: 1`
	`==============================`
	`cardID: 2`
	`cardContent: this is the user1's card content2 and it is in folderID`
	`cardCategory: user1's card category 1`
	`sourceApplication: Wechat`
	`dataType: Text`
	`cardAddedDate: 2013-02-21 00:00:00.000`
	`cardModifiedDate: 2022-03-22 00:00:00.000`
	`cardState: Not Hidden`
	`favoriteSelected: Selected`
	`folderID: 2`
	`==============================`
	`cardID: 3`
	`cardContent: this is the user1's card content3 and it is in folderID`
	`cardCategory: user1's card category 2`
	`sourceApplication: Whatsapp`
	`dataType: Text`
	`cardAddedDate: 2014-03-21 00:00:00.000`
	`cardModifiedDate: 2022-03-22 00:00:00.000`
	`cardState: Not Hidden`
	`favoriteSelected: Selected`
	`folderID: 2`
	`==============================`
	`cardID: 4`
	`cardContent: this is the user1's card content4 and it is in folderID`
	`cardCategory: user1's card category 2`
	`sourceApplication: Chrome`
	`dataType: Text`

	cardAddedDate: 2015-04-21 00:00:00.000 cardModifiedDate: 2022-03-21 00:00:00.000 cardState: Not Hidden favoriteSelected: Not selected folderID: 1

4. addCard(userID, cardID, content, category, sourceApplication, dataType)

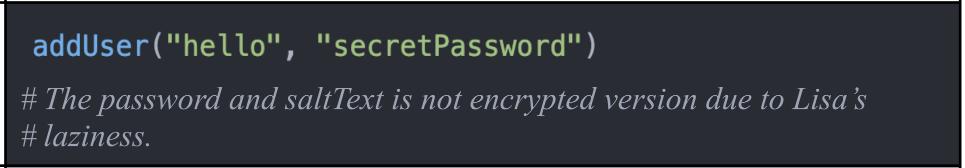
Parameters	- userID - cardID - content: card content - category: card category - sourceApplication: source application (ex. Chrome, Wechat) - dataType: (ex. Text, Image, URL)
Return Value	N/A
Explanation	This function is for adding the card to the database. Receives above information, and the function receives the local time to add the cardAddedDate and cardModifiedDate. cardState is initialized as 1 to represent that the hiding the card option is NOT selected. favoriteSelected is initialized as 0 to represent that favoriting the card option is NOT selected.
Example code	 `addCard(1, 10, 'blah blah', 'category', 'Safari', "Text")`

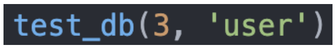
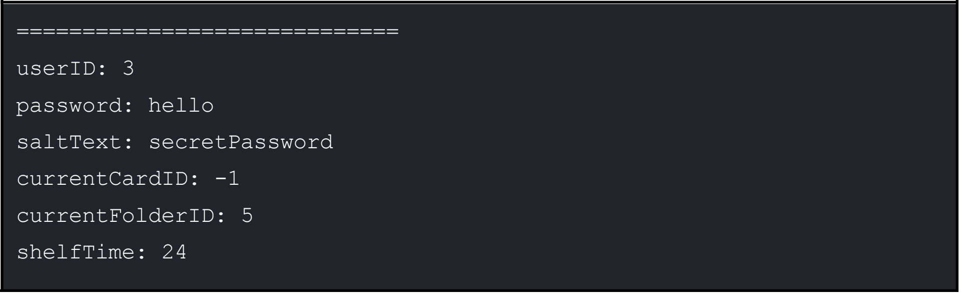
Output	N/A. But if you call that function and call `test_db(1, 'card')`, you can see that new card has been added to the database.
	New part has been **marked red** for your convenience!
	
	`==============================`
	`cardID: 1`
	`cardContent: this is the user1's card content1 and it is in folderID`
	`cardCategory: user1's card category 1`
	`sourceApplication: Kakaotalk`
	`dataType: Text`
	`cardAddedDate: 2012-01-21 00:00:00.000`
	`cardModifiedDate: 2022-03-21 00:00:00.000`
	`cardState: Not Hidden`
	`favoriteSelected: Not selected`
	`folderID: 1`
	`==============================`
	`cardID: 2`
	`cardContent: this is the user1's card content2 and it is in folderID`
	`cardCategory: user1's card category 1`
	`sourceApplication: Wechat`
	`dataType: Text`
	`cardAddedDate: 2013-02-21 00:00:00.000`

```
cardModifiedDate: 2022-03-22 00:00:00.000
cardState: Not Hidden
favoriteSelected: Selected
folderID: 2
==============================
cardID: 3
cardContent: this is the user1's card content3 and it is in folderID
cardCategory: user1's card category 2
sourceApplication: Whatsapp
dataType: Text
cardAddedDate: 2014-03-21 00:00:00.000
cardModifiedDate: 2022-03-22 00:00:00.000
cardState: Not Hidden
favoriteSelected: Selected
folderID: 2
==============================
cardID: 4
cardContent: this is the user1's card content4 and it is in folderID
cardCategory: user1's card category 2
sourceApplication: Chrome
dataType: Text
cardAddedDate: 2015-04-21 00:00:00.000
cardModifiedDate: 2022-03-21 00:00:00.000
cardState: Not Hidden
favoriteSelected: Not selected
folderID: 1
==============================
cardID: 10
cardContent: blah blah
cardCategory: category
sourceApplication: Safari
dataType: Text
cardAddedDate: 2022-05-10 10:00:25
cardModifiedDate: 2022-05-10 10:00:25
cardState: Not Hidden
favoriteSelected: Not selected
folderID: 1
```

5. addUser(password)

Parameters	- password: please enter the result from the encryptPassword function. - saltText: please enter the result from the encryptPassword function.
Return Value	N/A
Explanation	This function is for adding the user to the database. This function sets the *currentCardID* (the most recent cardID) and the *defaultFolderID* as -1 **temporarily**. Shelf time is initialized as 24 **months** which is 2 years. The currentCardID will be changed if the user adds the first card using *addCard()* function.

Example code	
	`addUser("hello", "secretPassword")`
	`# The password and saltText is not encrypted version due to Lisa's`
	`# laziness.`

Output	
	N/A. But if you call that function and call `test_db(3, 'user')`, you can see that new user has been added to the database.
	The userID part is 3 because I know that I only added 2 users previously.
	You may make use of *showAllUsers()* function that comes after this to print the data of all the users.
	
	`=============================`
	`userID: 3`
	`password: hello`
	`saltText: secretPassword`
	`currentCardID: -1`
	`currentFolderID: 5`
	`shelfTime: 24`

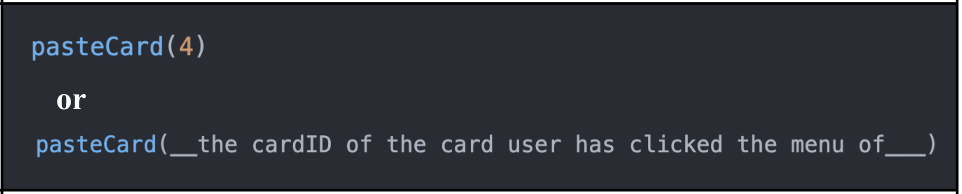
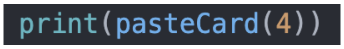
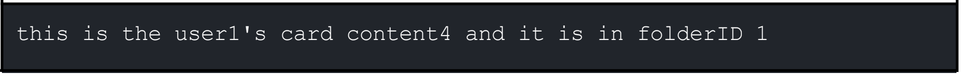
6. showAllUsers()

Parameters	N/A
Return Value	- userID list
Explanation	This function is for debugging purposes mostly.

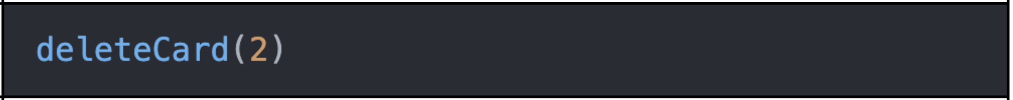
Example code	`for user in showAllUsers():`
	`    test_db(user, 'user')`

Output	By itself, it only returns the list like [1, 2, 3].
	But if you use this function like the example code section, you can see all the user information.
	`=============================`
	`userID: 1`
	`password: tempPassword1`
	`saltText: tempSaltText1`
	`currentCardID: 4`
	`currentFolderID: 1`
	`shelfTime: 120`
	`=============================`
	`userID: 2`
	`password: tempPassword2`
	`saltText: tempSaltText2`
	`currentCardID: 8`
	`currentFolderID: 3`
	`shelfTime: 140`
	`=============================`
	`userID: 3`
	`password: hello`
	`saltText: secretPassword`
	`currentCardID: -1`
	`currentFolderID: 5`
	`shelfTime: 24`

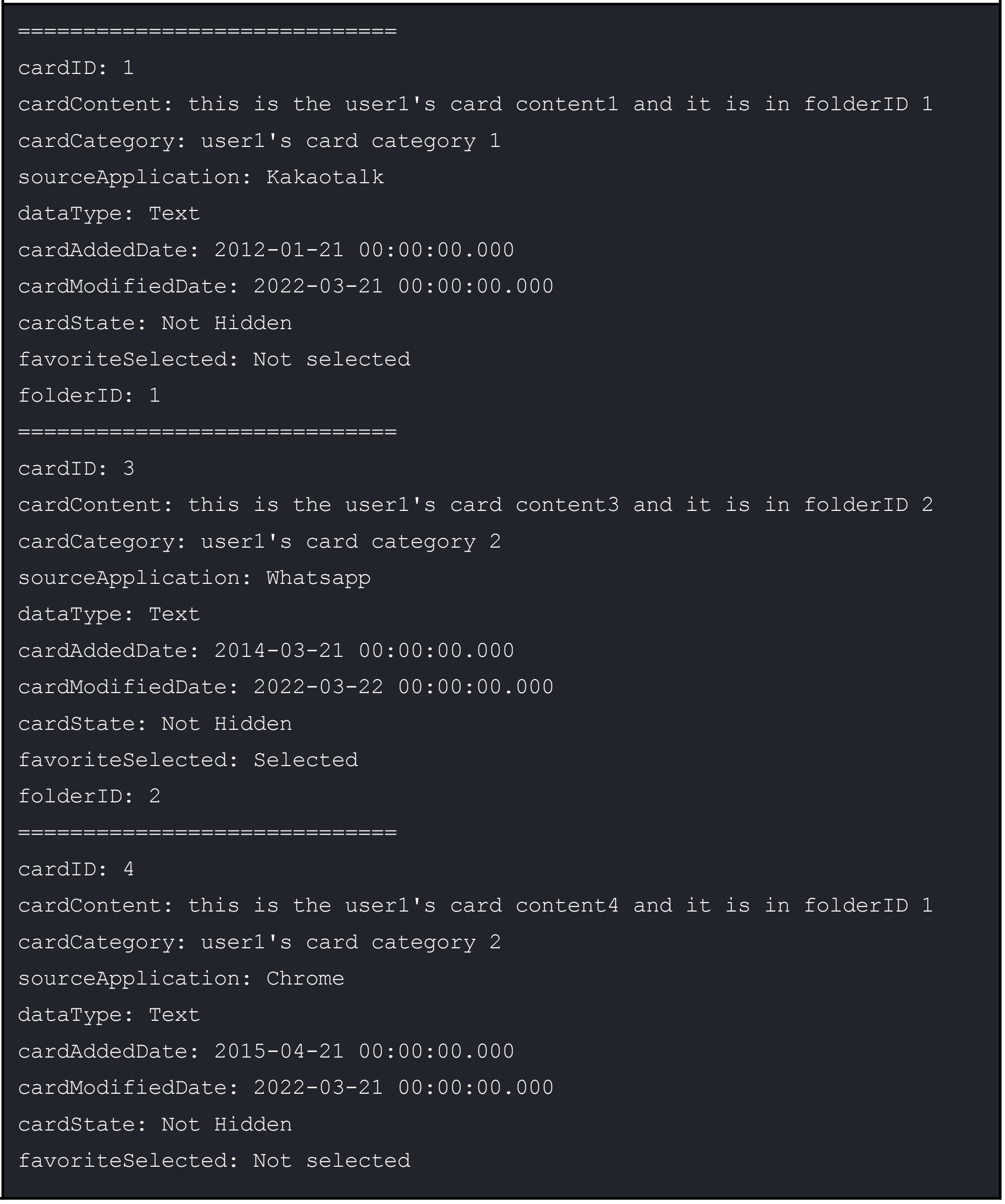
7. pasteCard(cardID)

Parameters	- cardID
Return Value	- cardContent
Explanation	This function is for pasting option. What I thought the paste function usage would be, if the user decides to paste something or press hotkey (commant + V), the available options (cards) would pop up. If the user clicks the card, the content should be pasted. Or! If the user clicks on the copy the content from the menu of the card, the information/content of the card should be pasted into the system so that the user can simply (commant + V) to paste the content.
Example code	 `pasteCard(4)` **or** `pasteCard(__the cardID of the card user has clicked the menu of___)`
Output	N/A. But if you do  `print(pasteCard(4))`, you can see that the card content of the card has been returned.  `this is the user1's card content4 and it is in folderID 1`

8. deleteCard(cardID)

<table>
<tr><td><b>Parameters</b></td><td>- cardID</td></tr>
<tr><td><b>Return Value</b></td><td>N/A</td></tr>
<tr><td><b>Explanation</b></td><td>This function is for deleting the card from the database.</td></tr>
<tr><td><b>Example code</b></td><td>


```
deleteCard(2)
```

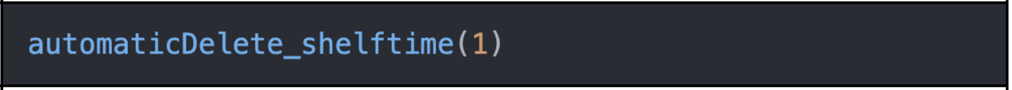
</td></tr>
<tr><td rowspan="2"><b>Output</b></td><td>N/A. But if you call that function and call `test_db(1, 'card')`, you can see that the card2 (for example above) has been deleted from the database.</td></tr>
<tr><td>


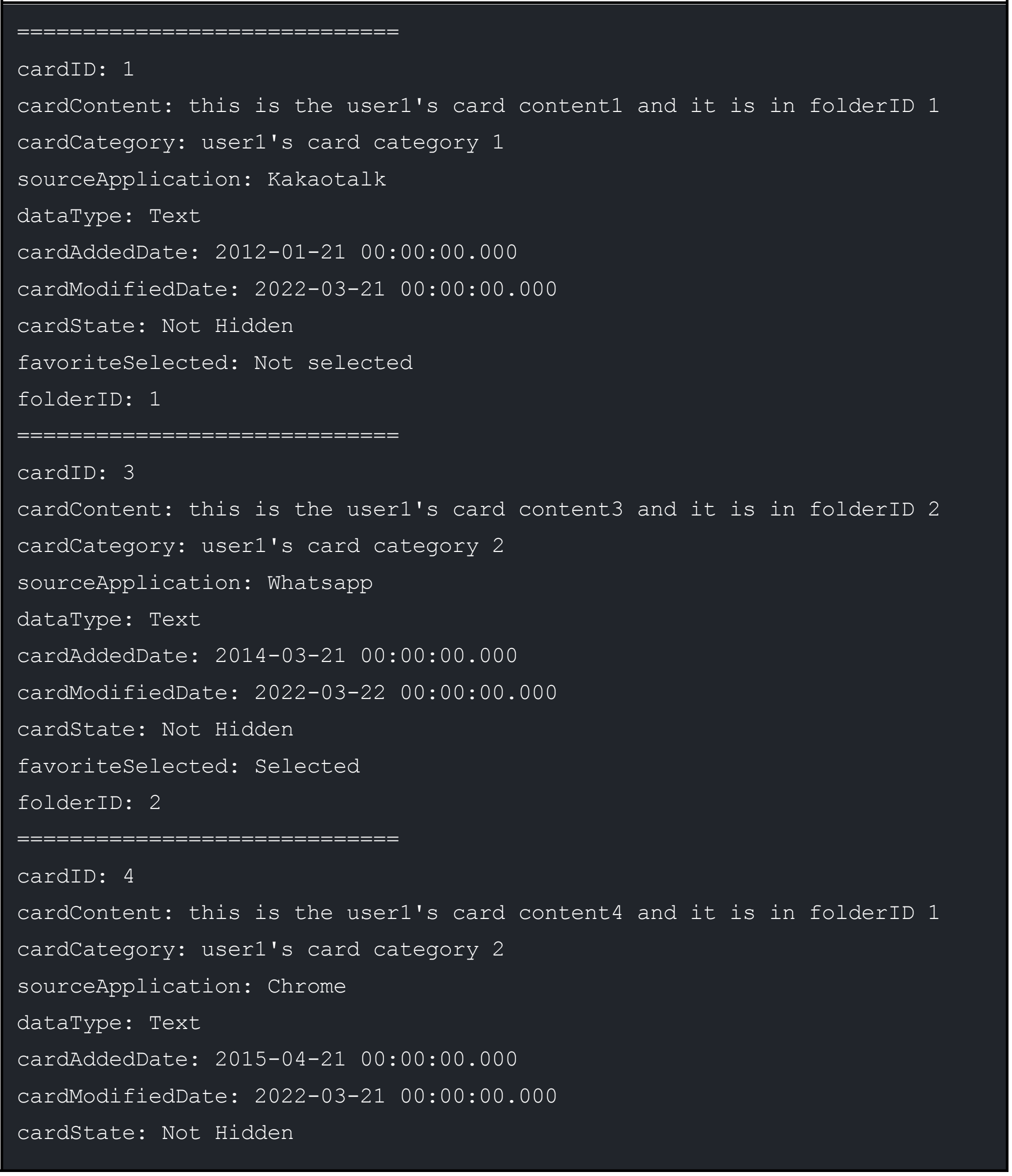
```
==============================
cardID: 1
cardContent: this is the user1's card content1 and it is in folderID 1
cardCategory: user1's card category 1
sourceApplication: Kakaotalk
dataType: Text
cardAddedDate: 2012-01-21 00:00:00.000
cardModifiedDate: 2022-03-21 00:00:00.000
cardState: Not Hidden
favoriteSelected: Not selected
folderID: 1
==============================
cardID: 3
cardContent: this is the user1's card content3 and it is in folderID 2
cardCategory: user1's card category 2
sourceApplication: Whatsapp
dataType: Text
cardAddedDate: 2014-03-21 00:00:00.000
cardModifiedDate: 2022-03-22 00:00:00.000
cardState: Not Hidden
favoriteSelected: Selected
folderID: 2
==============================
cardID: 4
cardContent: this is the user1's card content4 and it is in folderID 1
cardCategory: user1's card category 2
sourceApplication: Chrome
dataType: Text
cardAddedDate: 2015-04-21 00:00:00.000
cardModifiedDate: 2022-03-21 00:00:00.000
cardState: Not Hidden
favoriteSelected: Not selected
```

</td></tr>
</table>

	`folderID: 1`

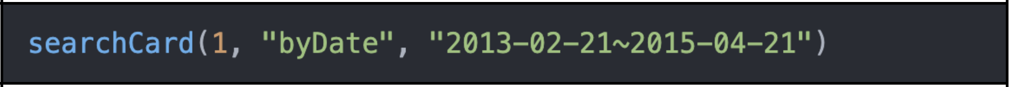
9. automaticDelete_shelftime(userID)

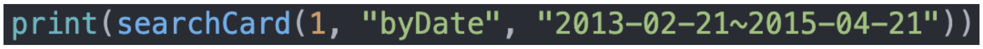
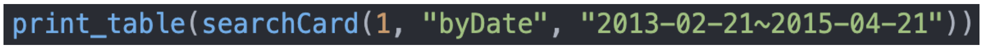
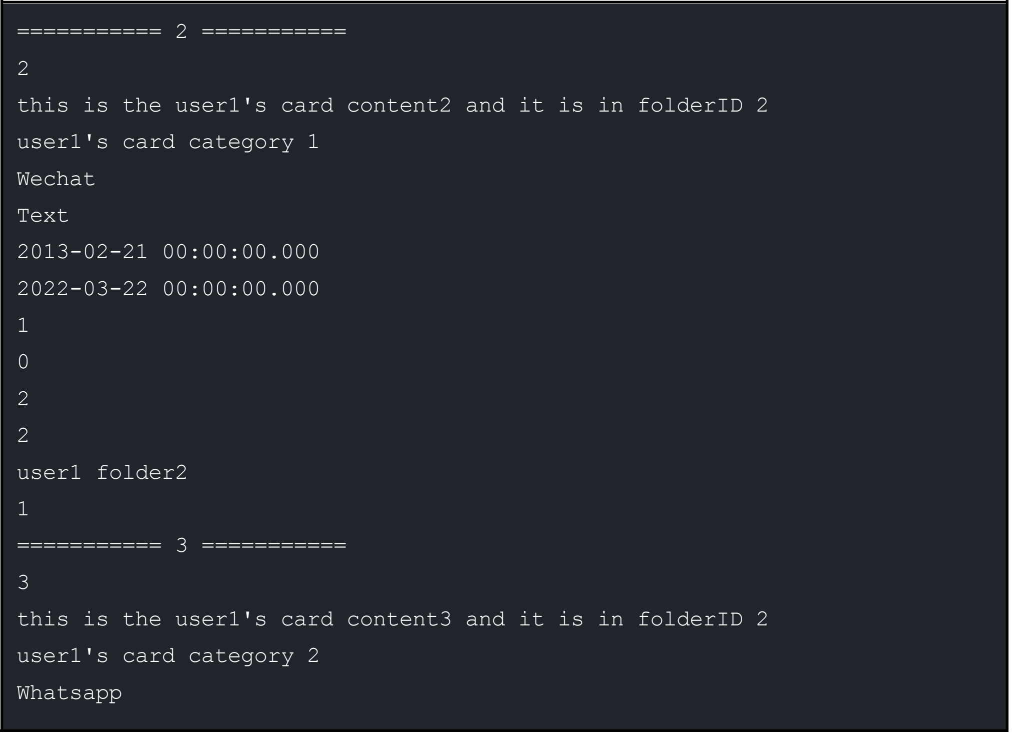
Parameters	- userID
Return Value	N/A
Explanation	This function is for deleting the card from the database, but based on the shelf time. This function should be ran everyday as the date changes.
Example code	 `automaticDelete_shelftime(1)`

Output	N/A. But if you call that function and call `test_db(1, 'card')`, you can see that the card2 (for example above) has been deleted from the database.
	
	`============================`
	`cardID: 1`
	`cardContent: this is the user1's card content1 and it is in folderID 1`
	`cardCategory: user1's card category 1`
	`sourceApplication: Kakaotalk`
	`dataType: Text`
	`cardAddedDate: 2012-01-21 00:00:00.000`
	`cardModifiedDate: 2022-03-21 00:00:00.000`
	`cardState: Not Hidden`
	`favoriteSelected: Not selected`
	`folderID: 1`
	`============================`
	`cardID: 3`
	`cardContent: this is the user1's card content3 and it is in folderID 2`
	`cardCategory: user1's card category 2`
	`sourceApplication: Whatsapp`
	`dataType: Text`
	`cardAddedDate: 2014-03-21 00:00:00.000`
	`cardModifiedDate: 2022-03-22 00:00:00.000`
	`cardState: Not Hidden`
	`favoriteSelected: Selected`
	`folderID: 2`
	`============================`
	`cardID: 4`
	`cardContent: this is the user1's card content4 and it is in folderID 1`
	`cardCategory: user1's card category 2`
	`sourceApplication: Chrome`
	`dataType: Text`
	`cardAddedDate: 2015-04-21 00:00:00.000`
	`cardModifiedDate: 2022-03-21 00:00:00.000`
	`cardState: Not Hidden`

	favoriteSelected: Not selected folderID: 1

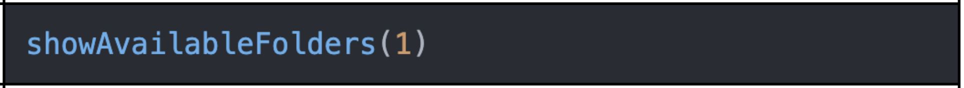
10. searchCard(userID, searchType, searchContent)

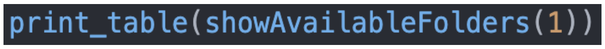
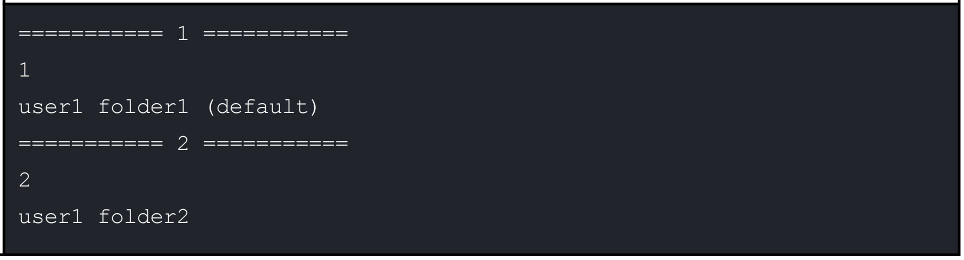
Parameters	- userID - searchType: the mode that the user selects to search in. - “byKeyword”: search by specific keyword - (ex. “Computer science”) - “byDate”: search by date parameter - (ex. 2013-02-21~2015-04-21) - searchContent: the section that is inside the search box.
Return Value	- List of **tuples** that contains all the data
Explanation	This function is for showing the content that the user is searching for.
Example code	 `searchCard(1, "byDate", "2013-02-21~2015-04-21")`

Output	N/A. But if you use print like
	 `print(searchCard(1, "byDate", "2013-02-21~2015-04-21"))`
	, you can see all the card information.
	For convenience, using print_table() function, the result has been iterated like below.
	 `print_table(searchCard(1, "byDate", "2013-02-21~2015-04-21"))`
	 `=========== 2 ===========`
	`2`
	`this is the user1's card content2 and it is in folderID 2`
	`user1's card category 1`
	`Wechat`
	`Text`
	`2013-02-21 00:00:00.000`
	`2022-03-22 00:00:00.000`
	`1`
	`0`
	`2`
	`2`
	`user1 folder2`
	`1`
	`=========== 3 ===========`
	`3`
	`this is the user1's card content3 and it is in folderID 2`
	`user1's card category 2`
	`Whatsapp`

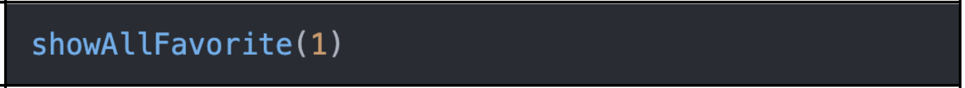
	Text 2014-03-21 00:00:00.000 2022-03-22 00:00:00.000 1 0 2 2 user1 folder2 1

11. showAvailableFolders(userID)

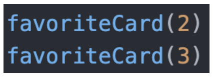
Parameters	- userID
Return Value	- List of tuples of the (folderID, folderName) that the specific user owns.
Explanation	This function is for showing all the folders that the user owns.
Example code	 `showAvailableFolders(1)`

Output	N/A since there is no print() inside the function itself. However, if you use print() or print_table(), you can see the return values.  `print_table(showAvailableFolders(1))` has been used for your convenience.
	
	`=========== 1 ===========`
	`1`
	`user1 folder1 (default)`
	`=========== 2 ===========`
	`2`
	`user1 folder2`

12. showAllFavorite(userID)

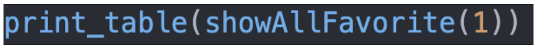
<table>
<tr><td><b>Parameters</b></td><td>- userID</td></tr>
<tr><td><b>Return Value</b></td><td>- List of tuples of all the card information that have been favorited by the user</td></tr>
<tr><td><b>Explanation</b></td><td>This function is to show all the cards which user pressed favorite/star(to represent favorite option) button.</td></tr>
<tr><td><b>Example code</b></td><td>


```
showAllFavorite(1)
```
</td></tr>
<tr><td><b>Output</b></td><td>
N/A because there is no print() inside the showAllFavorite() function.
However, if you use print() or print_table() function, you will be able to see all the result inside the console for debugging.
For this example, favoriteCard() function -- will be shown later in this document -- like below already has been used.



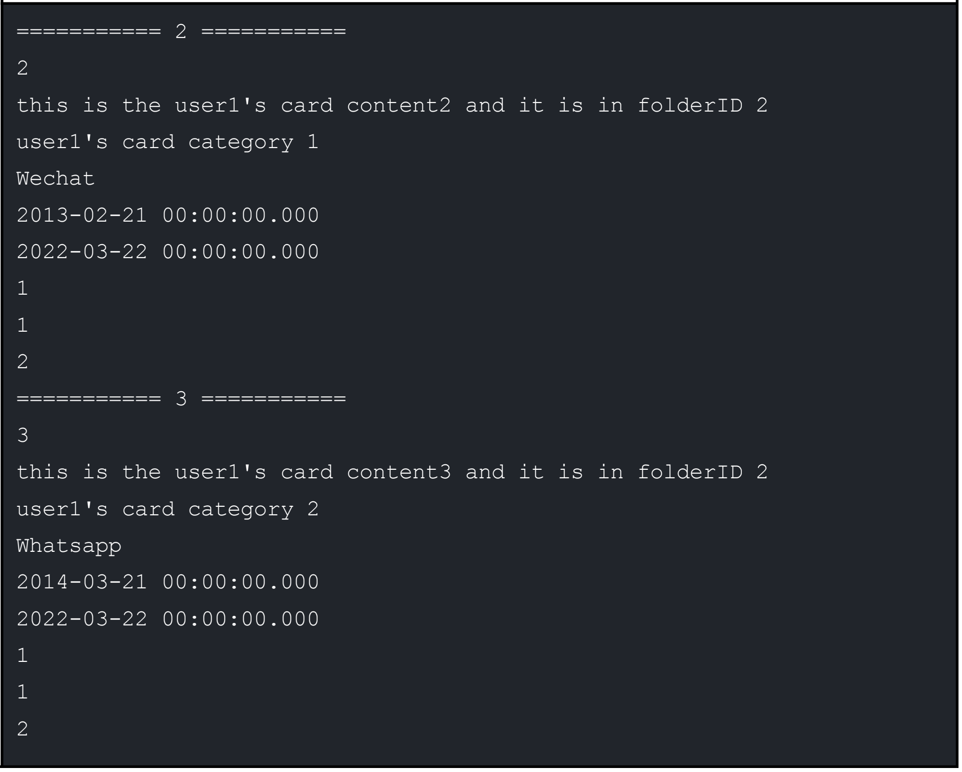
```
favoriteCard(2)
favoriteCard(3)
```

# meaning I temporarily favorited card 2 and 3.



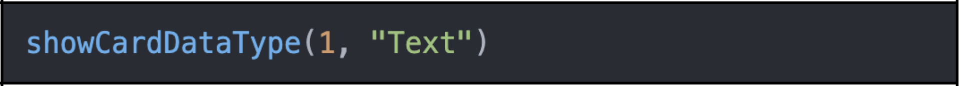
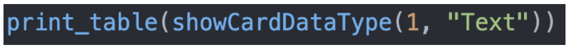
```
print_table(showAllFavorite(1))
```

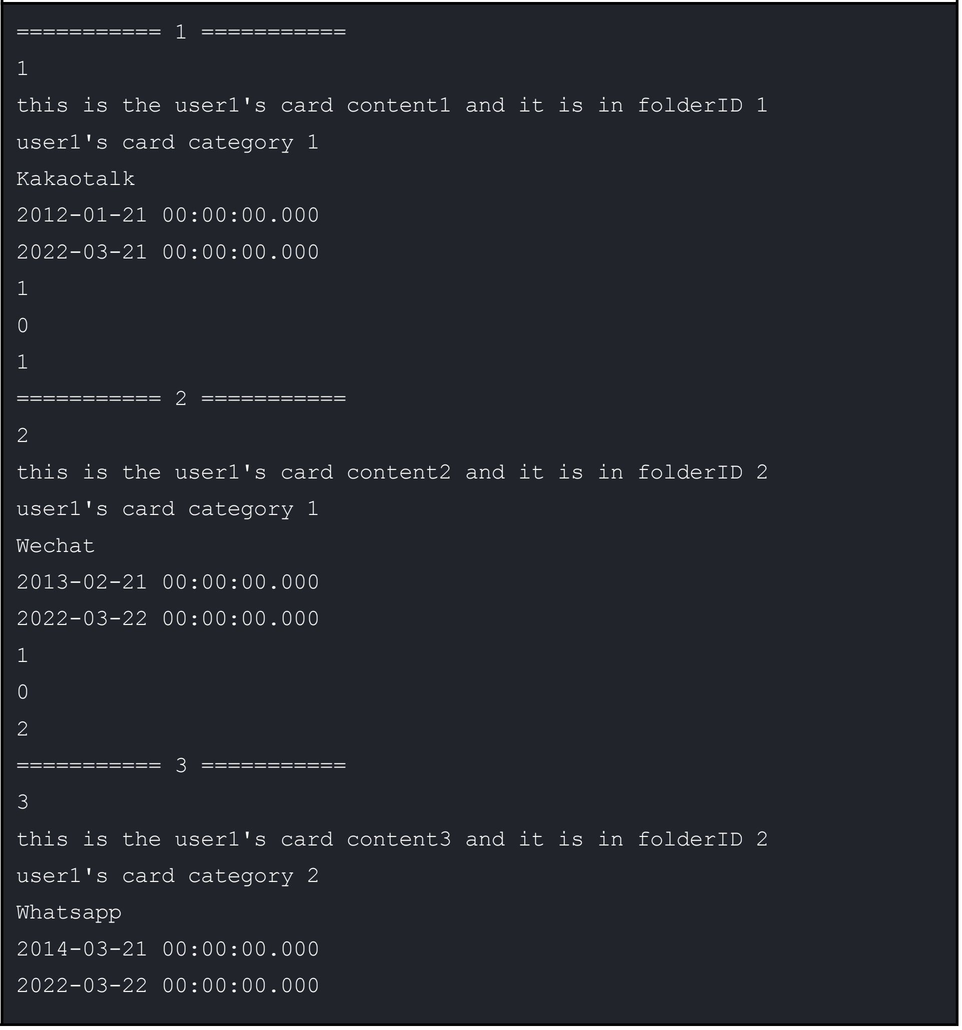
has been used for your convenience.



```
=========== 2 ===========
2
this is the user1's card content2 and it is in folderID 2
user1's card category 1
Wechat
2013-02-21 00:00:00.000
2022-03-22 00:00:00.000
1
1
2
=========== 3 ===========
3
this is the user1's card content3 and it is in folderID 2
user1's card category 2
Whatsapp
2014-03-21 00:00:00.000
2022-03-22 00:00:00.000
1
1
2
```
</td></tr>
</table>

13. showCardDataType(userID, datatype)

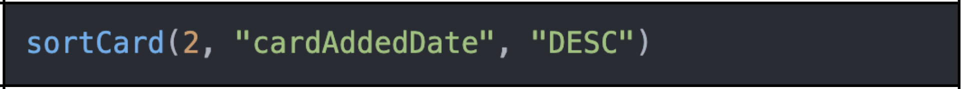
Parameters	- userID - datatype: data type like Text, Image, URL
Return Value	- List of tuples of all the inputted data type
Explanation	This function is to show all the cards which has specific data type
Example code	 `showCardDataType(1, "Text")`
Output	N/A because there is no print() inside the showCardDataType() function. However, if you use print() or print_table() function, you will be able to see all the result inside the console for debugging.  `print_table(showCardDataType(1, "Text"))` has been used for your convenience.

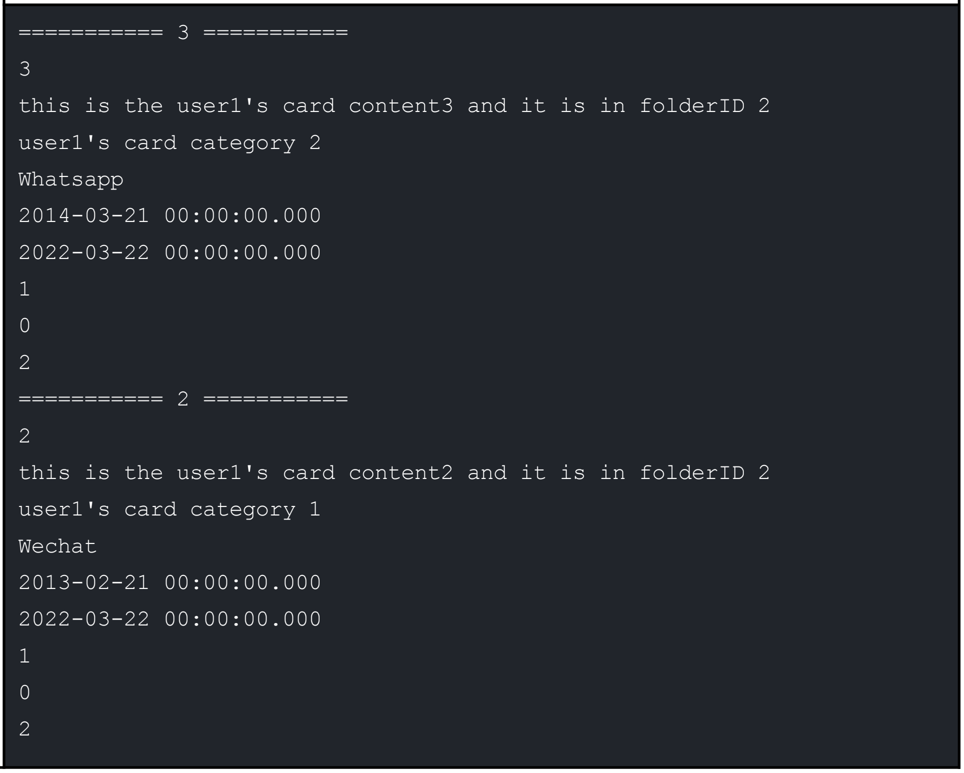


```
=========== 1 ===========
1
this is the user1's card content1 and it is in folderID 1
user1's card category 1
Kakaotalk
2012-01-21 00:00:00.000
2022-03-21 00:00:00.000
1
0
1
=========== 2 ===========
2
this is the user1's card content2 and it is in folderID 2
user1's card category 1
Wechat
2013-02-21 00:00:00.000
2022-03-22 00:00:00.000
1
0
2
=========== 3 ===========
3
this is the user1's card content3 and it is in folderID 2
user1's card category 2
Whatsapp
2014-03-21 00:00:00.000
2022-03-22 00:00:00.000
```

```
1
0
2
=========== 4 ===========
4
this is the user1's card content4 and it is in folderID 1
user1's card category 2
Chrome
2015-04-21 00:00:00.000
2022-03-21 00:00:00.000
1
0
1
```

14. sortCard(folderID, sortType, whatOrder)

Parameters	- folderID - sortType: attribute name (ex. cardAddedDate) - whatOrder - “DESC”: descending order - “ASC”: ascending order
Return Value	- List of tuples of all the card inside that folder
Explanation	Returns all the cards inside that folder but with what order
Example code	 `sortCard(2, "cardAddedDate", "DESC")`

Output	N/A because there is no print() inside the sortCard() function. However, if you use print() or print_table() function, you will be able to see all the result inside the console for debugging. `print_table(sortCard(2, "cardAddedDate", "DESC"))` has been used for your convenience.
	
	`=========== 3 ===========`
	`3`
	`this is the user1's card content3 and it is in folderID 2`
	`user1's card category 2`
	`Whatsapp`
	`2014-03-21 00:00:00.000`
	`2022-03-22 00:00:00.000`
	`1`
	`0`
	`2`
	`=========== 2 ===========`
	`2`
	`this is the user1's card content2 and it is in folderID 2`
	`user1's card category 1`
	`Wechat`
	`2013-02-21 00:00:00.000`
	`2022-03-22 00:00:00.000`
	`1`
	`0`
	`2`

15. changeShelfTime(userID, newShelfTime)

Parameters	- userID - newShelfTime: the shelftime that the user wants to change to.
Return Value	N/A
Explanation	This function updates the shelf time to the duration that the user wants to keep the cards. The duration must be inputted as the number of months.

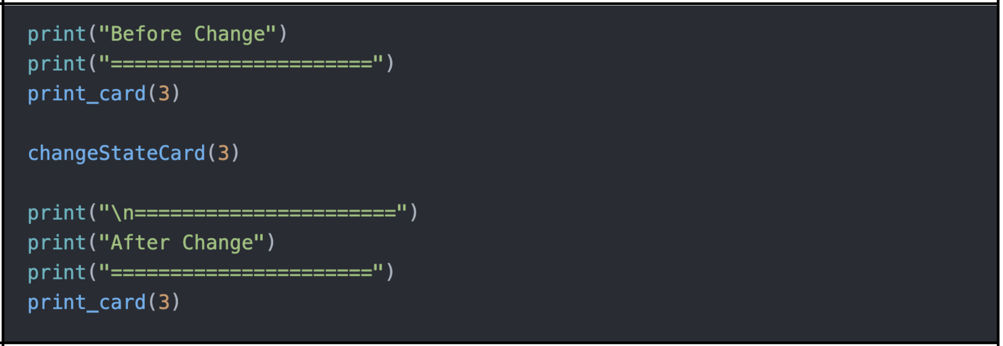
Example code	`print("=============================")`
	`print("Before Change")`
	`test_db(2, 'user')`
	
	`changeShelfTime(2, 12)`
	
	`print("\n=============================")`
	`print("After Change")`
	`test_db(2, 'user')`

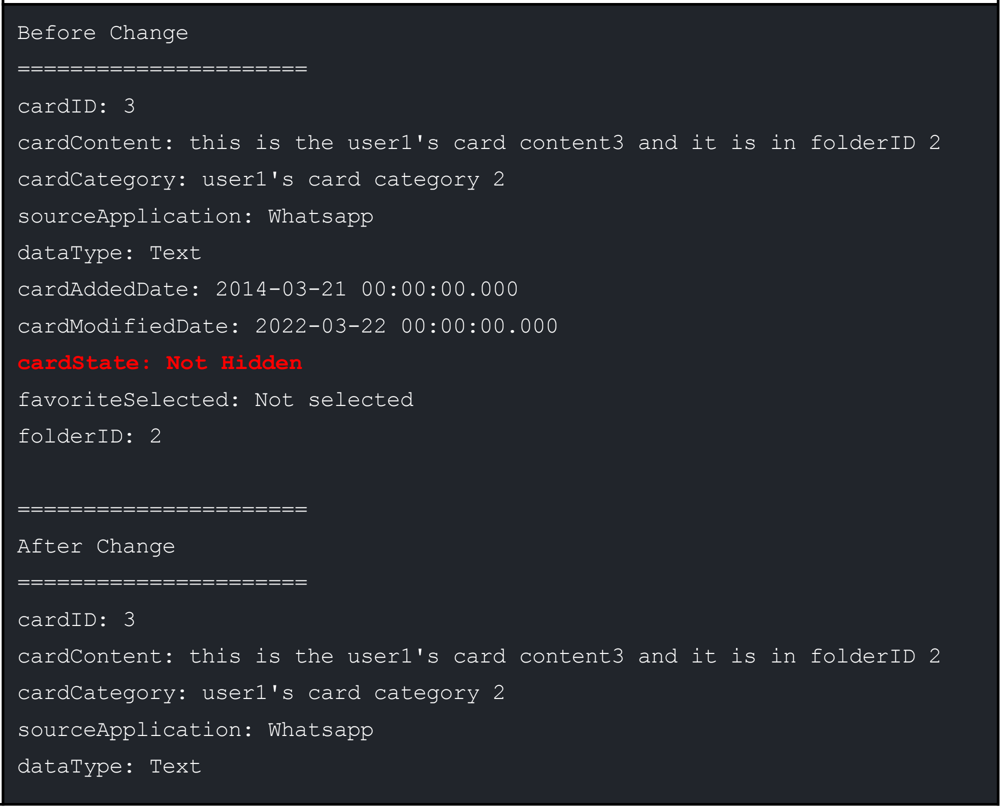
Output	N/A because there is no print() inside the changeShelfTime() function. However, you can use test_db() function to check the update. test_db() has been called twice to compare the before and after for you.

	`=============================`
	`Before Change`
	`=============================`
	`userID: 2`
	`password: tempPassword2`
	`saltText: tempSaltText2`
	`currentCardID: 8`
	`currentFolderID: 3`
	`shelfTime: 140`
	
	`=============================`
	`After Change`
	`=============================`
	`userID: 2`
	`password: tempPassword2`
	`saltText: tempSaltText2`
	`currentCardID: 8`
	`currentFolderID: 3`
	`shelfTime: 12`

16. changeStateCard(cardID)

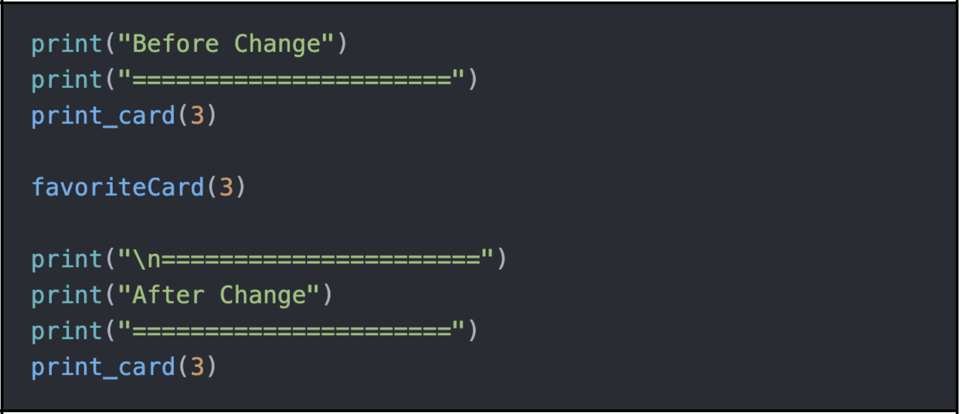
Parameters	- cardID
Return Value	N/A
Explanation	This function toggles the card state from hidden to not hidden and vice versa. This function can be called when the user presses the hide button or not hide button which can be selected from the menu.

Example code	
	print("Before Change")
	print("======================")
	print_card(3)
	
	changeStateCard(3)
	
	print("\n======================")
	print("After Change")
	print("======================")
	print_card(3)

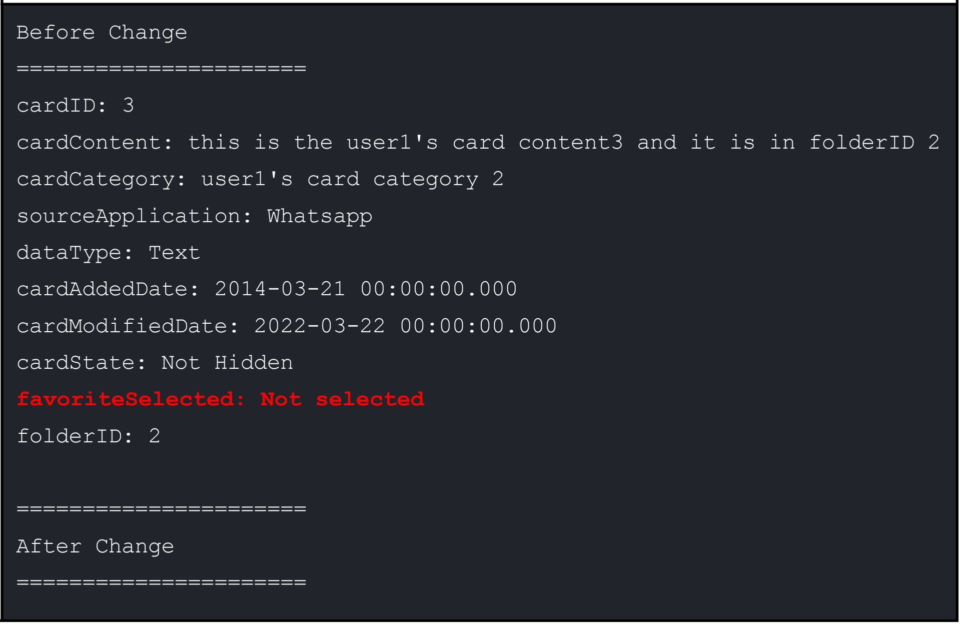
Output	N/A because there is no print() inside the changeStateCard() function. However, you can use print_card() function to check the update. print_card() has been called twice to compare the before and after for you. The changed part has been **marked red** for your convenience!
	 Before Change ====================== cardID: 3 cardContent: this is the user1's card content3 and it is in folderID 2 cardCategory: user1's card category 2 sourceApplication: Whatsapp dataType: Text cardAddedDate: 2014-03-21 00:00:00.000 cardModifiedDate: 2022-03-22 00:00:00.000 **cardState: Not Hidden** favoriteSelected: Not selected folderID: 2 ====================== After Change ====================== cardID: 3 cardContent: this is the user1's card content3 and it is in folderID 2 cardCategory: user1's card category 2 sourceApplication: Whatsapp dataType: Text

	cardAddedDate: 2014-03-21 00:00:00.000 cardModifiedDate: 2022-03-22 00:00:00.000 **cardState: Hidden** favoriteSelected: Not selected folderID: 2

17. favoriteCard(cardID)

<table>
<tr><td><b>Parameters</b></td><td>- cardID</td></tr>
<tr><td><b>Return Value</b></td><td>N/A</td></tr>
<tr><td><b>Explanation</b></td><td>This function toggles the card state from favorite selected to not selected and vice versa. This function can be called when the user presses the favorite button or not favorite button which can be selected from the menu.</td></tr>
<tr><td><b>Example code</b></td><td>
<pre>
print("Before Change")
print("======================")
print_card(3)

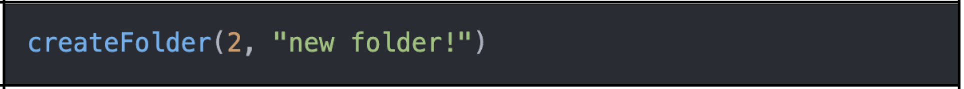
favoriteCard(3)

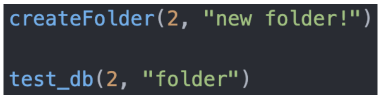
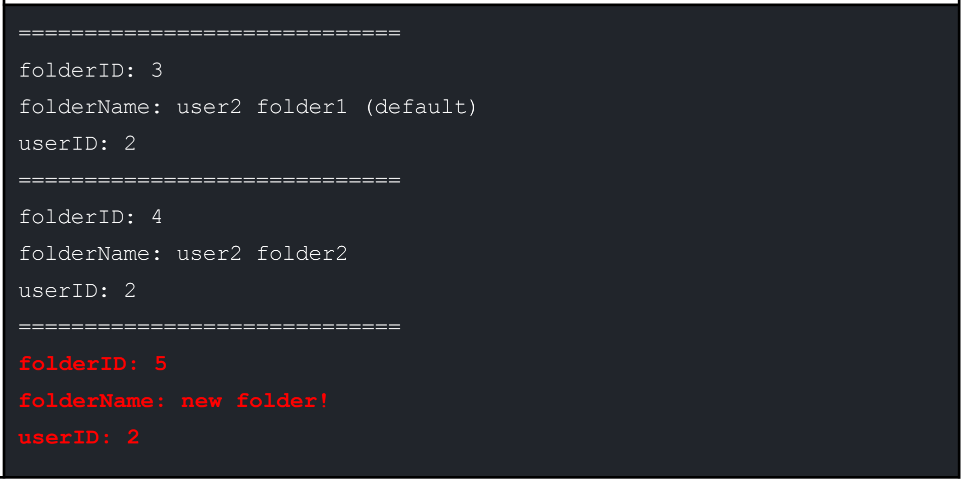
print("\n======================")
print("After Change")
print("======================")
print_card(3)
</pre></td></tr>
<tr><td rowspan="2"><b>Output</b></td><td>N/A because there is no print() inside the favoriteCard() function. However, you can use print_card() function to check the update. print_card() has been called twice to compare the before and after for you.
The changed part has been <b>marked red</b> for your convenience!</td></tr>
<tr><td>
<pre>
Before Change
======================
cardID: 3
cardContent: this is the user1's card content3 and it is in folderID 2
cardCategory: user1's card category 2
sourceApplication: Whatsapp
dataType: Text
cardAddedDate: 2014-03-21 00:00:00.000
cardModifiedDate: 2022-03-22 00:00:00.000
cardState: Not Hidden
favoriteSelected: Not selected
folderID: 2

======================
After Change
======================
</pre></td></tr>
</table>

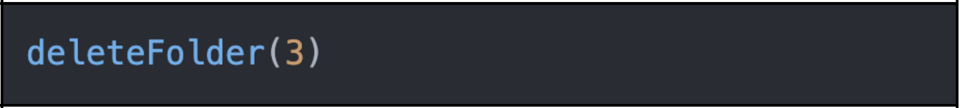
	cardID: 3 cardContent: this is the user1's card content3 and it is in folderID 2 cardCategory: user1's card category 2 sourceApplication: Whatsapp dataType: Text cardAddedDate: 2014-03-21 00:00:00.000 cardModifiedDate: 2022-03-22 00:00:00.000 cardState: Not Hidden **favoriteSelected: Selected** folderID: 2

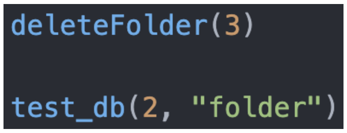
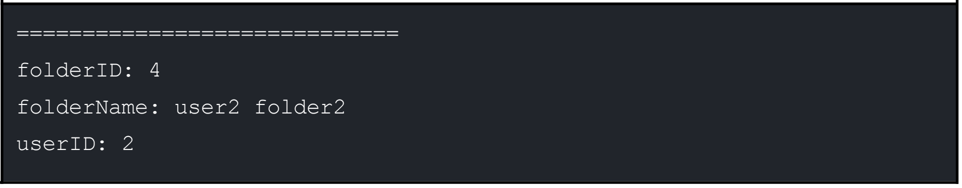
18. createFolder(userID, folderName)

Parameters	- userID - folderName
Return Value	N/A
Explanation	This function receives above information and adds new folder into the database
Example code	 `createFolder(2, "new folder!")`

Output	N/A because there is no print() inside the createFolder() function.
	However, you can use test_db() function like below to check the update.
	
	`createFolder(2, "new folder!")`
	
	`test_db(2, "folder")`
	The changed part has been **marked red** for your convenience!
	
	`==============================`
	`folderID: 3`
	`folderName: user2 folder1 (default)`
	`userID: 2`
	`==============================`
	`folderID: 4`
	`folderName: user2 folder2`
	`userID: 2`
	`==============================`
	`folderID: 5`
	`folderName: new folder!`
	`userID: 2`

19. deleteFolder(folderID)

Parameters	- folderID
Return Value	N/A
Explanation	This function deletes folder from the database.
Example code	 `deleteFolder(3)`

Output	because there is no print() inside the deleteFolder() function. However, you can use test_db() function like below to check the update.
	
	`deleteFolder(3)`
	
	`test_db(2, "folder")`
	If you see, the folder3 has been deleted from the database.
	
	`=============================`
	`folderID: 4`
	`folderName: user2 folder2`
	`userID: 2`