

Javadoc Interface Documentation

public class FriendList

A list of user profiles: these profiles were first added by request by an end user, which was then accepted by the owners of the user profiles. This list is viewed by the end user in the FriendsActivity.

Method Detail:

void addFriend(User user)
adds the current user to the friendlist
Parameter: user – the user profile to add to the friendlist

void removeFriend(User user)
If the specified user is in the friendlist, the user is then removed
Parameter: user – the user profile to remove from the friendlist

public class FriendListController

The FriendListController class takes in the user input and changes the data concerning the Friendlist within the model

Method Detail:

void addFriend(User user)
adds the current user to the friendlist
Parameter: user – the user profile to add to the friendlist

void removeFriend(User user)
If the specified user is in the friendlist, the user is then removed
Parameter: user – the user profile to remove from the friendlist

public class Inventory implements Search

The end user's inventory which contains a list of Items

Method Detail:

void addItem(Item item)
Adds the specified item to the end user's inventory

Parameter: item – the item to add to the friendlist

void removeItem(Item item)

If the specified item is in the inventory, the item is then removed from the inventory
--

Parameter: item – the item to remove from the friendlist
--

ArrayList<Item> searchByCategory(String category)

Searches the inventory for items within the specified category
--

Returns: an array list of all items whose category matches the string

Parameter: category – a string containing the category in which the search is based on
--

public class InventoryController

The InventoryController class takes in the user input and changes the data concerning the inventory within the model

Method Detail:

void addItem(Item item)

Adds the specified item to the end user's inventory

Parameter: item – the item to add to the friendlist

void removeItem(Item item)

If the specified item is in the inventory, the item is then removed from the inventory
--

Parameter: item – the item to remove from the friendlist
--

public class Item implements Serializable

The Item class is an object in which the user specifies a description of the item, the quality of the item, a photo of the item, whether the item can be viewed publicly or privately, the category of the item, and the quantity of the item which the user possesses.

Method Detail:

String toString()

Returns: The name of the item

boolean isPrivate()

Finds out whether the item can be viewed publicly or privately
--

Returns: a boolean variable, if it is true than the item can only be viewed privately

public class Pending

This model is for the pending friend requests, both the ones you sent (needs to be implemented) and the friend requests you have received. Here you can accept or reject a friend request. You can also cancel a previously sent request.

void acceptFriend(User user)

Accepts the friend request and adds them to the current user friend list. It also removes the pending friend request from the received pending list.
--

Parameter: user – the user profile to accept
--

void acceptAllFriends(ArrayList list)

This accepts all friend requests in Pending and adds them all to the current User friend list. It also removes all of the friend requests from the received pending list
--

Parameter: list – the list of user profiles to accept

void addPending(User user)

This adds a user to the sent pending list. This is invoked when the current User requests a friend.

Parameter: user – the user profile to send the request to

void cancelRequest(User user)

This removes a friend request from the sent requests pending list.
--

Parameter: user – the user profile which the request was made for

void cancelAllRequest(ArrayList list)

This removes a friend request from the sent requests pending list.
--

Parameter: list – the list of user profiles which the requests were made for
--

void removeRequest(User user)

This removes a friend request from the received friend requests pending list.

Parameter: user – the user profile which who made the request

void removeAllRequest(ArrayList list)

This removes all pending received friend requests.
--

Parameter: list – the list of user profiles who sent the pending requests

public class PendingController

The controller for the Pending class.

void acceptFriend(User user)

Accepts the friend request and adds them to the current user friend list. It also removes the pending friend request from the received pending list.
--

Parameter: user – the user profile to accept
--

void acceptAllFriends(ArrayList list)

This accepts all friend requests in Pending and adds them all to the current User friend list. It also removes all of the friend requests from the received pending list
--

Parameter: list – the list of user profiles to accept

void addPending(User user)

This adds a user to the sent pending list. This is invoked when the current User requests a friend.

Parameter: user – the user profile to send the request to

void cancelRequest(User user)

This removes a friend request from the sent requests pending list.
--

Parameter: user – the user profile which the request was made for

void cancelAllRequest(ArrayList list)

This removes a friend request from the sent requests pending list.
--

Parameter: list – the list of user profiles which the requests were made for
--

void removeRequest(User user)

This removes a friend request from the received friend requests pending list.

Parameter: user – the user profile which who made the request

void removeAllRequest(ArrayList list)

This removes all pending received friend requests.
--

Parameter: list – the list of user profiles who sent the pending requests

public class Profile

This is the profile of users within the app, whether it is the end user or the friend(s) of the end user. It contains the user's first name, last name, age, gender, email account, photo, phone number, and the name of the city in which the user resides.