CRC Card

BY: Group Alifafa

Entity

User

- Each user has an unique ID, a userName, an associated password, an avatar
- Create a user ID when Creating a new user account.
- It has a list of friends, each friend is a User object.
- It stores message history between the user and the user’s friends.
- It has a list of posts that was posted by the user.
- It has a list blocked user, each blocked user is an User object. The blocked user cannot add this user as friend.
- It has a request list attribute which stores a string of friend requests.
- It has a string attribute called loggedIn, which indicate whether the user is logged in or not.

- Post
- Sharingcentre
- PtoPMessage
- PtoPMessageManager

StoreUser

- Store the existing user accounts - In the storage, each user must have a unique ID	User

PtoPMessage

- Store the content and created time of a person -to -person message
- Store the sender and the receiver of a person -to -person message
- Get the content, created time, sender, receiver of a person -to -person message

- PtoPMessageManager
- User

SharingCentre

- Has a list of posts from the user itself and the user’s friends
- Has a list of notifications containing informations
    - Who liked my post (the User, when the post is liked, the specific post)
    - Who commented my post (the User, when the post is commented, the specific post)
- Contain a state whether or not there is new post or new notification coming to this user’s SharingCentre

Post

SharingCentreManager

Notifications

- Each Notification contains the User who liked or commented, the time, and the regarding post
- Has a status distinguishing whether this is a notification about like or comment

Post

SharingCentre

ParagraphPost

- Contain a string of the time the post was posted
- Contains the visibility of the post, whether it is private or public
- Contains a string of the location of the post that was added by the user
- Contains an int of how many likes the post has
- Contains a list of all the comments made to the post
- Contains a post ID

- Picture post(subclass of para. post
- Post manager
- Sharing centre

PicturePost

ParagraphPost

- Contain a string of the time the post was posted
- Contains the visibility of the post, whether it is private or public
- Contains a string of the location of the post that was added by the user
- Contains an int of how many likes the post has
- Contains a list of all the comments made to the post
- Contains a list of pictures uploaded for the post

- Paragraph post(super class)
- Post manager
- Sharing centre

Use Case

Abstract

UseCase

UserManager, PtoPMessageManager

- This is an abstract class.
- All the use cases should extend this class.

Controller (UseCase is an attribute of Controller)

UserManager

PtoPMessageManager

PostManager

UserManager

Responsibilities

- It can create a User
- Find friend for a given user with given friend's userName.
- Login
  - Return a password with given ID after searching in StoreUser. Return null if such user does not exist.
- Return user with given ID
- Change Username
- Change Password
- Add Friend, Remove Friend
- Return user’s request list
- Change LogIn status

User

Group

StoreUser

PtoPChatController

PtoPMessageManager

Responsibility	Collaborators
- Created a person -to -person message - Send a person -to -person message - Return person -to -person message history between two users - Delete person -to -person message	- User - PtoPMessage - PtoPChatController

PostsManager

- Delete a post
- Edit a post
- Favorite a post (save it to favorite)
- Comment a post
- Start a post
  - Post a post

- Paragraph post
- Picture post
- Sharing centre

Controller

Abstract

Controller

UserController, ChatController

- This is an abstract parent class for all controllers. - It contains an array of string that stores user prompts, and also contains a UseCase object.	UseCase

UserController

Controller

LogInController, AccountRegistrationController, MessageFriendController

- This is an abstract class that extends Controller Class - It is the parent class for all UserControllers. - User will be logged in after enter correct ID and password. UserController will check if their ID and password match. If the ID does not exist, return “ID does not exist” - When users are logged in, they can change their password - Get all user prompts	UserManager

LogInController

UserController

- This class is responsible for managing the user input and calling associated UserManager when the user wants to log in.	UserManager

AccountRegistrationController

UserController

- This class is responsible for managing the user input and calling associated UserManager when the user wants to create an account.	UserManager

CommandExecutor

- It will evaluate the input from the user and decides which Controller to call.	UserController ChatController

ChatController

Controller

PtoPChatController

- A parent class of PtoPChatControl	Controller PtoPChatController

PtoPChatController

Controller

PtoPChatController

- Send message from a user to another user by getting their user id

ChatController

UserManager

PtoPChatManager

Presenter

ChatPresenter

Use the Chat manager to get all messages of a chat, then return a string that shows the these messages.

Interface

CommandUI

- This is the interface of our program. - User can enter the name of the tasks they want to complete. (eg. Log in, Create an Account, Message a Friend). The interface will choose the correct controller based on the input. - The interface can prompt the user to enter their information related to the specific task, the input will be sent to the correct controller.	CommandsExecutor Controller