

Offline assignment

Task

The task is to develop a solution for the below mentioned problem. You are expected to build a CLI (command line interface) application, which should take input on STDIN and post output on STDOUT (please note error messages are to be directed to STDOUT and not STDERR). There is no need for any other interface (Web, APIs etc). The key focus area will be the design of the solution and the structure of the code. Weightage will be given to clear design, extensibility of code to allow easy addition of features, modularity to ensure clear separation of concerns. Apart from the mentioned test cases, we may manually run a few additional test cases, which are closely related to the ones mentioned.

You are expected to provide a tar / zip file of the entire source tree, along with the git history. Please do not send binaries. Please include a standard Makefile (and/or a build.sh as required), to build the project and produce all the binaries needed. Also, please include / produce a "run.sh" to actually run the project with all the parameters as needed.

Environment

No libraries apart from the chosen language's standard library installed in the default installation should be used. The implementation should accompany a **build.sh**, **run.sh** and a **README** file which mentions details of the version of the language compiler/interpreter being used etc. In case of non-standard compiler/interpreter, please mention the same in the readme along with the URL to download and steps to execute. The following dependencies can be assumed to be present on the machine:

Additional notes

Please ensure concepts of extensibility, ease of testing, abstraction and appropriate application of type safety are demonstrated.

Actual Problem statement

“cloudShop” is a marketplace platform which allows users to ‘buy’ and ‘sell’ items. The task is to implement the core functionality of a marketplace with the following entities:

- User
 - Anyone who uses the marketplace
 - Identified by username, which should be unique throughout the platform, but case **insensitive**.
 - Each user can create any number of listings having any category.
- Listing
 - A listing of an item put up for sale on the marketplace. Only registered users should be allowed to buy or sell items.
 - Listings should have the following fields:
 - Title
 - Description
 - Price
 - Username
 - Creation time
 - Each listing can be associated with only 1 user and 1 category.
- Category
 - Groupings of listings of the same "category". E.g. Electronics, Fashion etc

Only Create, Read and Delete operations are required to be implemented. Update operation can be omitted.

Please ensure the design is clean enough to add related features without much changes to the existing code and its structure.

Following are the actions that a user can perform on the marketplace:

- REGISTER <username>
Responses:
 - "Success"
 - "Error - user already existing"
- CREATE_LISTING <username> <title> <description> <price> <category>
Responses:
 - "<listing id>"
 - "Error - unknown user"
- DELETE_LISTING <username> <listing_id>
Responses:
 - "Success"
 - "Error - listing does not exist"
 - "Error - listing owner mismatch"
- GET_LISTING <username> <listing_id>
 - Responses:
 - "<title>|<description>|<price>|<created_at>|<category>|<username>"
This command should return any listing queried according to listing_id, not limited to listings created by the user. Username is taken just for the purpose of authentication.
 - "Error - not found"
 - "Error - unknown user"
- GET_CATEGORY <username> <category>
Responses:
 - "Error - category not found"
 - "Error - unknown user"
 - "<title>|<description>|<price>|<created_at>
<title>|<description>|<price>|<created_at>
<title>|<description>|<price>|<created_at>
This command should return listings of the specified category in descending order by create time (ie, created_at).
- GET_TOP_CATEGORY <username>
 - Responses:
 - "Error - unknown user"
 - <category name> (Category having the highest total number of listings).
This command should consider all listings in the marketplace and not just the listings created by the user issuing the command. Username is taken just for the purpose of authentication.
 - This operation is expected to be a read heavy operation as it can be used on the home page etc. Please ensure suitable optimization for the same.

Example flow (with output)

The application is expected to show a "# " prompt to accept input.

```
# REGISTER user1
Success

# CREATE_LISTING user1 'Phone model 8' 'Black color, brand new' 1000 'Electronics'
100001

# GET_LISTING user1 100001
Phone model 8|Black color, brand new|1000|2019-02-22 12:34:56|Electronics|user1

# CREATE_LISTING user1 'Black shoes' 'Training shoes' 100 'Sports'
100002

# REGISTER user2
Success

# REGISTER user2
Error - user already existing

# CREATE_LISTING user2 'T-shirt' 'White color' 20 'Sports'
100003

# GET_LISTING user1 100003
T-shirt|White color|20|2019-02-22 12:34:58|Sports|user2

# GET_CATEGORY user1 'Fashion'
Error - category not found

# GET_CATEGORY user1 'Sports'
T-shirt|White color|20|2019-02-22 12:34:58|Sports|user2
Black shoes|Training shoes|100|2019-02-22 12:34:57|Sports|user1

# GET_TOP_CATEGORY user1
Sports

# DELETE_LISTING user1 100003
Error - listing owner mismatch

# DELETE_LISTING user2 100003
Success

# GET_TOP_CATEGORY user2
Sports

# DELETE_LISTING user1 100002
Success

# GET_TOP_CATEGORY user1
Electronics

# GET_TOP_CATEGORY user3
Error - unknown user
```

Consolidated STDIN:

```
REGISTER user1
CREATE_LISTING user1 'Phone model 8' 'Black color, brand new' 1000 'Electronics'
GET_LISTING user1 100001
CREATE_LISTING user1 'Black shoes' 'Training shoes' 100 'Sports'
REGISTER user2
REGISTER user2
CREATE_LISTING user2 'T-shirt' 'White color' 20 'Sports'
GET_LISTING user1 100003
GET_CATEGORY user1 'Fashion'
GET_CATEGORY user1 'Sports'
GET_TOP_CATEGORY user1
DELETE_LISTING user1 100003
DELETE_LISTING user2 100003
GET_TOP_CATEGORY user2
DELETE_LISTING user1 100002
GET_TOP_CATEGORY user1
GET_TOP_CATEGORY user3
```

Expected STDOUT:

```
Success
100001
Phone model 8|Black color, brand new|1000|2019-02-22 12:34:56|Electronics|user1
100002
Success
Error - user already existing
100003
T-shirt|White color|20|2019-02-22 12:34:58|Sports|user2
Error - category not found
T-shirt|White color|20|2019-02-22 12:34:58|Sports|user2
Black shoes|Training shoes|100|2019-02-22 12:34:57|Sports|user1
Sports
Error - listing owner mismatch
Success
Sports
Success
Electronics
Error - unknown user
```