

CSCI3100 asg1

February 7, 2025

1. Choice: 2
2. Choice: 1
3. Choice: 1
4. Choice: 1,2,3
5. Choice: 1
6. Choice: 2,3,4
7. Choice: 2,3,4
8. Choice: 1,4
9. Choice: 1,2
10. Choice: 1
11. Choice: 4
12. Choice: 3
13. Choice: 1,2,3,4,5
14. Choice: 4
15. Choice: 3
- 16.1.

Requirements:

- (1) The system shall support to be used on a mobile terminal, such as a mobile phone, a tablet.
- (2) The system shall support to identify the bird through inputs with: a) Image only b) Sound only c) Image and sound
- (3) The system recognition accuracy shall be comparable to bird experts.
- (4) The system shall give results within 30 seconds.
- (5) The recognition of birds shall include bird information of a) Bird name b) lifespan c) Habitats

Specifications:

- (1) The system shall be accessible as An iOS and an Andriod application, compliant with the latest iOS development guidelines; also shall be accessible as a web application, accessible via a standard website URL.
- (2) The server shall process the image and sound inputs using recognition algorithms, perform a database lookup in the bird database, and return the recognition results to the user interface.
- (3) The system server shall support the deep learning framework ResNet to ensure the accuracy of the recognition results.
- (4) The system shall support to accept sound input of no less than 5 seconds.

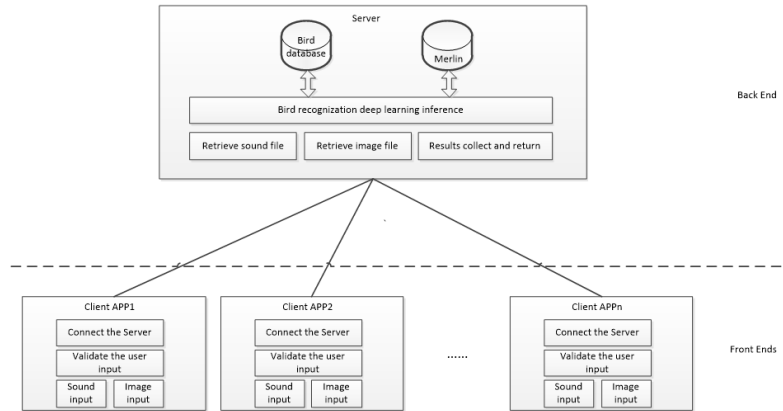
- Any sound records less than 5 seconds shall be rejected by the front end.
- (5) The system shall support the bird database of no less than 10,000 birds on the server.
- (6) The system shall support access to existing bird identification APIs (such as Merlin Bird ID) to enhance data sources.
- (7) The system shall use low-latency communication protocols (such as Web-Socket) to provide fast results delivery. (8) The system shall protect user privacy: (a) require user authentication to access search history (b) encrypt all user information during storage and transmission.

16.2.

The client server model is suggested. The reasons are:

- (1) The sound and image recognition processing capabilities of mobile clients are limited, so it is necessary to support the remote deployment of more powerful back-end servers to perform recognition processing and database lookup functions. In this case, the client server model provides the model for rough processing at front end and accurate but highly computation-required processing at back end.
- (2) The scenario of bird recognition application may be real-time identification outdoors. In this case, the input of bird images and sounds is likely to come from images collected on site, rather than just pre-stored images, so it is necessary to support image and sound input through mobile device clients APP. By using the client server model, the system can better receive the data sent from different front-ends (as illustrated in diagram) and do the prediction in the back end.
- (3) Since the product needs to maintain a database, it would be good to manage the database with server mechanisms and the users can access the database as clients.
- (4) Different users may access at the same time from different devices, with different user information (like password and account), and it is more convenient to manage using client server model.

A possible diagram is shown below:



17.

The log message is:

```
Reimplement printing the best solution to enhance code reuse
```

```
The function for printing the board can be used to print the
best solution. Doing this can prevent code duplication and
inconsistent implementations for the same purpose.
```

```
Format people often use when writin commit messages:
One-line summary of 50-100 characters,
followed by an empty line,
then some detailed explanations about the patch, if needed.
```

```
We do not refer to code, but focus on describing why the
changes have been made and how.
```

```
Present tense is used.
```

18.

Note: I think there are different understandings for the first 2 questions: a) compared to the original version of commit 25fcc83, what has changed in the version of commit a5832d. b) what are the differences of the 2 commit actions.

(1)

Answer in way a): *main.go* file was changed

Answer in way b): In commit 3a5832d, 1 file is changed: *main.go*. In commit 25fcc83, no files are changed (added 3 files *go.mod*, *main.go* and *solver_test.go*, and I think adding files should not be considered as changing files).

(2)

Answer in way a): No file added

Answer in way b): In commit 3a5832d, no file is added. In commit 25fcc83, 3 files are added: *go.mod*, *main.go* and *solver_test.go*.

(3)

With respect to commit 25fcc83, lines 95, 96, 97 which are:

```
    for i := range c.bestSol {
        fmt.Printf("% 3d\n", c.bestSol[i])
    }
```

are deleted and replaced with a new line:

```
    c.printBoard()
```

19. 3 branches

20.

(0) Work on doc/author_list:

```
git checkout doc/author_list
```

(1) command

```
echo -n "1155191596" | sha256sum
```

gets result "ee5fc3541b1264a24ef0ba31c4b6525fa88a693ddeed3e0c9646427d0e174f67".
So the hash value is "ee5fc3541b1264a24ef0ba31c4b6525fa88a693ddeed3e0c9646427d0e174f67"

(2) Create file with command

```
echo "ee5fc3541b1264a24ef0ba31c4b6525fa88a693ddeed3e0c9646427d0e174f67" > my_secret.txt
```

(3) Added with command

```
git add .
```

(4) Committed with command

```
git commit -m "A commit for adding my_secret.txt"
```

(5) Using

```
git log
```

to get the newest commit ID: **46aa1d8df7d0b6d1b9948907ee5a475ec4a994a521**.

(0) Work on main:

```
git switch main
```

(1) Merge using

```
git merge feature/add_secrets
```

(2) After trying to merge with

```
git merge doc/author_list
```

and getting the error

```
Auto-merging my_secret.txt
CONFLICT (add/add): Merge conflict in my_secret.txt
Automatic merge failed; fix conflicts and then commit the result.
```

The content of my_secret.txt becomes

```
<<<<<< HEAD
c775e7b757ede630cd0aa1113bd102661ab38829ca52a6422ab782862f268646
=====
ee5fc3541b1264a24ef0ba31c4b6525fa88a693ddeed3e0c9646427d0e174f67
>>>>>> doc/author_list
```

Following the policy, change it to

```
c775e7b757ede630cd0aa1113bd102661ab38829ca52a6422ab782862f268646
ee5fc3541b1264a24ef0ba31c4b6525fa88a693ddeed3e0c9646427d0e174f67
```

Then do

```
git add .
git commit
```

Using "git log" to get the newest commit ID: **e1dbf5b623b6a71733045366aaa07e34896dd82d**
22.

```
(base) leosunx@Leo:~/CUHK/csci3100/asg1/new_2_6/assignment_1_repo$ git log --graph --oneline --decorate --all
* e1dbf5b (HEAD -> main) Merge branch 'doc/author_list'
|
| * 46aa1d8 (doc/author_list) A commit for adding my_secret.txt
| * 2051bbb Merge branch 'feature/add_secrets'
|/
|/
| * a593383 (origin/feature/add_secrets, feature/add_secrets) add a file of secrets to be used by the application
| * f5efb0f (origin/main, origin/doc/author_list, origin/HEAD) add a doc storing the list of authors
|/
|/
* 3a5832d Reimplement printing the best solution to enhance code reuse
* 25fcc83 initial commit
```

23.

It could be deleted since it is no longer needed. For example "git branch -d branch_name" locally or "git push origin --delete branch_name" on remote.

24.

```
# make sure local commits are up to date
git add .
git commit -m "make sure up to date"
```

```
# push to remote repo
git push
```

```
# Then make pull request on web page
```