Group reviewed: ql101-hz166
Reviewer: ry70

Code quality

What they did well:

1. Very clear model structures.
2. For each field which user may not enter values, there is a default value stated.
3. Variables are properly named with meaningful names.
4. Login page is clear: only register/login options showed
5. Exceptions are caught and taken cared of.
6. A driver cannot take his/her own ride.

What they could improve:

1. When a user tries to register as a driver, he/she must manually click “is_driver” field in the driver registration form.
2. A driver can be registered without entering plate number.
3. A driver cannot search for open rides if he/she does not register vehicle type: if a user states *null* in his/her vehicle request, which is default, the driver must enter *null* in his registration form to allow the driver to see this ride.
4. Number of passengers in a ride should not be negative anyways.
5. When a user tries to register as a driver, he/she must manually type in his vehicle type instead of a list of choices. This could be problematic as a user input is not bounded. For example, a lower-cased “suv” should be equivalent to “sUv”. As a result, if a open ride requires vehicle type of “suv”, the driver could not find this ride.
6. When there is no open rides, nothing is displayed. This could gives users an illusion of network failure.
7. The code needs more comments.

Vulnerability

What they did well:

1. Most database-related operations require login status.
2. Forms are validated before submission.
3. ForeignKey constraint is used properly.

What they could improve:

1. Some operations does not require login status, such as DriverInfo. If directly type the URL in the navigation bar, a 404 error is popped up instead of redirecting to driver registration page and at the same time exposes all possible URL patterns.
2. Username and password are submitted using plaintext as parameters in a POST request form, which can be inspected by a third party.

3. When a form is submitted, a “redirect” should be used instead of “render” as a user might accidentally hit a “back” button when he/she is waiting for a response. This might lead to a form being submitted twice.

Lessons Learned

Rui Yang - ry70

Hwk1

1. I did not make my code generic: many “if-else” statements were used to prepare for targeted situation. This is not professional.
2. I did not write a clear “readme” statement to illustrate how to run my code based on my testing procedure. This led trouble for my reporters but indeed expose more vulnerabilities of my code.
3. ForeignKey constraint was not used properly because at that time I was really confused about constraints. I would make up in the future.
4. Testing was not enough. I should have thought out of the box and test from different angles, such as from users, sharers, and drivers.

Hwk2

1. There were some mega-functions where one function lasted over one hundred lines. Altough I wrote many comments, this still made my code hard to understand.
2. Macros should have been named in capital letters.
3. Repetitive codes could be put in one function and being called multiple times to reduce code length and improve readability.
4. Exception handling was not enough. More “try-catch” statements should be used. Uncaught exceptions would result in termination of processes and this should not happen on server processes as servers should be run forever. Another example is when called *send*() and *recv*() as these two functions could return -1 and failed.
5. Concurrency was not implemented safely that not all critical sections were locked properly. For example, if cache reads were not locked, one thread may be writing into the cache while another thread was trying to read and yield the dirty data.

ECE568 HW3

NetID: tl259, ry70

Name: Tongbo Liu, Rui Yang

Review for HW2 of bh214 and js896

1. Code Quality

- Well designed infrastructure of the poject.
- Good abstraction used in their code, with each function being specific and simple. Though some of the abstraction is meaningless, it's generally good.
- Good use of C++ features.
- Well designed of OOD.
- The comments are concise and useful.
- Detailed requirements sheet and danger log, good resource for analyzing behavior of their code.
- Good naming for functions and variables, intuitive and easy to understand.

2. Security

- Cannot open online game pages like www.4399.com/flash/18012.html. And the program just terminated and throw an exception as below:

```
receiver_test_1  | terminate called after throwing an instance of 'connection_break_exception*'
ersshwk2bh214js896_receiver_test_1 exited with code 139
```

- When the proxy is run more than once, the log will continue writing into the log file starting from 0 without clearing the previous logs.
- When I tried several pages, it outputs "Cannot assign requested address" continously and the server will just terminate (also exited with code 139).

```
receiver_test_1  | Received string length = 0; Cannot assign requested address
receiver_test_1  | Received string length = 0; Cannot assign requested address
receiver_test_1  | Received string length = 0; Cannot assign requested address
receiver_test_1  | Received string length = 0; Cannot assign requested address
receiver_test_1  | Received string length = 0; Cannot assign requested address
receiver_test_1  | Received string length = 0; Cannot assign requested address
receiver_test_1  | Received string length = 0; Cannot assign requested address
receiver_test_1  | Received string length = 0; Cannot assign requested address
receiver_test_1  | Received string length = 0; Cannot assign requested address
receiver_test_1  | Received string length = 0; Cannot assign requested address
receiver_test_1  | Received string length = 0; Cannot assign requested address
receiver_test_1  | Received string length = 0; Cannot assign requested address
receiver_test_1  | Received string length = 0; Cannot assign requested address
receiver_test_1  | Received string length = 0; Cannot assign requested address
receiver_test_1  | Received string length = 0; Cannot assign requested address
receiver_test_1  | Received string length = 0; Cannot assign requested address
```

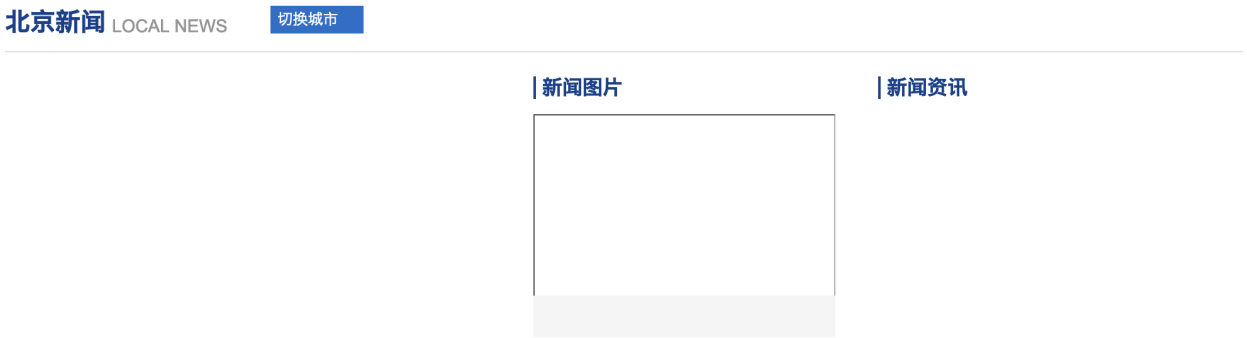
- Sometimes POST not working properly, e.g. when I register a new account the server just terminate and throw an exception as below:

```
receiver_test_1  | terminate called after throwing an instance of 'connection_break_exception*'
ersshwk2bh214js896_receiver_test_1 exited with code 139
```

- Cannot handle 404 and 502 error, e.g. when I input http://www.568hao.com/ the proxy server just terminated.

```
receiver_test_1  | terminate called after throwing an instance of 'get_addrinfo_exception*'
ersshwk2bh214js896_receiver_test_1 exited with code 139
```

- Sometimes cannot load the contents in some pages. And when such things happen, the server will just terminate (also exited with code 139).



- The server cannot work well when implementing multithread processing (e.g. open ten pages), and the server will terminate recursively (also exited with code 139).

```
receiver_test_1  | terminate called recursively
receiver_test_1  | terminate called recursively
receiver_test_1  | terminate called recursively
receiver_test_1  | terminate called recursively
receiver_test_1  | terminate called recursively
receiver_test_1  | terminate called recursively
ersshwk2bh214js896_receiver_test_1 exited with code 139
```

3. Conclusion

The server has good code quality, but it's not robust enough when hadling some pages and it cannot handle some errors as it will always exit when meet some errors and I didn't see them in danger logs.

4. tl259's Hw1 + Hw2 lessons

1. Hw1

1. The most insecure attribute is putting "pk" directly in the URLs. The URL might be changed and the results are non-deterministic.
2. With respect to code quality, it is hard to read without comment. Although the code is not repeated and also well-formed, adding some comment is till necessary for reviewers, or even the programmer himself in several weeks, to understand the meanings.
3. "View your non-complete orders!" and "View your finished orders" will take me to the request page without any explanation when I have no such order. I suppose that I should fill a form to check my orders, but it tells me that I create a new request. This also happens in other places. Maybe adding some pop-up notifications will be better.
4. When I want to join a ride, I try to check if there are available rides. However, there is no such option, and I have to create a share request. Then I can search for such rides in the "Request a share order" page. This seems to be quite confusing.
5. The "View your joined orders" shows I have no such order at first, but becomes blank after I finish a ride.
6. There are also good designs for security. For example, a user who is not logged in cannot access any page of this website. And when I try to edit the order owned by another user, it is forbidden. It also reports an error when accessing pages that does not exist.
7. The most common exploit would be changing the URLs, especially the "pk" values. And there are also weird things happening after editing the number of passengers.

2. Hw2

1. About the code equality, we should have good abstraction used in their code, with each function being specific and simple. Clean code makes it easy to read. We should make good use of C++ features, including classes and access control.Generally good OOD, but some code are not making the best use of single responsibility, like some time comparison in proxy.h. Most comments are short and helpful, but some helper functions are not well-named or commented, mainly in proxy.h, making their behavior harder to understand. There is little duplication, except for some if statements in cache.h, could probably be extracted into a new function. We have detailed requirements sheet and danger log, good resource for analyzing behavior of their code. Good naming for functions and variables, intuitive and easy to understand.
2. Cache not working properly in http://people.duke.edu/~bmr23/ece568/class.html, it is not cached but it should be.
3. There's no explanation of expire time if there's no expire time validation.

4. Cannot open online game pages like www.4399.com/flash/18012.htm, and 502 bad gateway is logged.
5. When the proxy is run more than once, the log will continue writing into the log file starting from 1 without clearing the previous logs.
6. There's parsing error caused following result.