

# Go Entry Task

## 1. Requirements

### Features

Implement a user manager system for user to login and edit their profiles. Users can login on web page.

After a user inputs his/her username and password, the backend system will authenticate his/her identity. If the login is successful, the relevant user information is displayed, otherwise an error message will be shown.

After a successful login, a user can edit the following information:

1. Upload a picture as his/her profile picture
2. Change his/her nickname (support unicode characters with utf-8 encoding)

User information includes: username (cannot be changed), nickname, profile picture. For test, the initial user data can be directly insert into database. Make sure there are at least 10 million user accounts in the test database.

### Time Limit

You need to finish this task in 5 working days.

### Performance

- Supports up to 1000 login requests per second (from at least 200 unique users)
- Supports up to 1000 concurrent http requests

### Environment

- Server: Virtual Machine on Working PC
- OS: CentOS 7 x64 or Ubuntu 14.04 above
- DB: MySQL 5.5 or above
- Client: Chrome and Firefox

### **Design Constraint**

- Separate HTTP server and TCP server and put the main logic on TCP server
- Backend authentication logic should be done in the TCP server
- User information must be stored in a MySQL database. Connect by MySQL Go client.
- Use standard library whenever possible.
- Web interface will not directly connect to MySQL. For each HTTP request, web interface will send a TCP request to the TCP server, which will handle business logic and query the database.

### **Design Considerations**

- Robustness
- Security
- Performance

## **2. Deliverables**

- Project source

- Design document
- Installation and maintenance documentation
- Performance tests report
- Concluding report

### 3. References

- Go: <http://golang.org>
- Coding style: <https://github.com/golang/go/wiki/CodeReviewComments>
- Testing: <https://golang.org/pkg/testing/>
- Profiling: <http://blog.golang.org/profiling-go-programs>
- Go Web application example: <https://golang.org/doc/articles/wiki/>
- Go editor/IDE
  - <https://github.com/fatih/vim-go>
  - <https://github.com/dominikh/go-mode.el>
  - <https://github.com/DisposaBoy/GoSublime>
  - <https://github.com/visualfc/liteide>
- MySQL client: <https://github.com/go-sql-driver/mysql>
- Redis: <http://redis.io>

- Redis Client: <https://github.com/go-redis/redis>