

# Functional Requirements

## 1. Login

- a. Actors: All actors should be able to login.
- b. Pre-condition : to have an account
- c. Work flow : user enter username and password
- d. Exception: wrong username or password, the user should reenter then again.
- e. Post condition : the user success to login

## 2. Review Profile

- a. Actors: All user should be able to review their profile.
- b. Pre- condition: to login to the profile.
- c. Workflow: user click on “review profile” button.
- d. Exception: user did not login.
- e. Post condition: user enter to “Review profile” page.

## 3. Search :

- a. Actors: All actors should be able to search.
- b. Pre-condition: to be login.
- c. Workflow: user enter query to search and click search button.
- d. Exception: user did not logging the system, the user enter unrelated keys.
- e. Post condition: system show result of searching query.

## 4. Create profile

- a. Actors: Employer, Freelancer.
- b. Pre-condition.
- c. Workflow: user enter unique username , password , email, birth date, telephone.
- d. Exception: username used by another user, password is week, wrong date format, wrong email format.
- e. Post condition: the user create profile and the data saved in database.

## 5. List task

- a. Actors: Employer.
- b. Pre-condition: user is logging into the system.
- c. Workflow: user click “list task” button.
- d. Exception: there is no tasks to show.
- e. Post condition: System list the available tasks.

## 6. Reject task:

- a. Actors: Employer.
- b. Pre-condition: Freelancer apply the task.
- c. Workflow: Employer reject the freelancer request.
- d. Post condition: the task app is rejected.

## 7. Get request

- a. Actors: Employer
- b. Pre-condition: the Employer should be logging into system.
- c. Workflow: user click task requests button.
- d. Exception: there is no task requests.

- e. Post condition: user show task requests.

**8. Accept task**

- a. Actors: Employer
- b. Pre-condition:
- c. Workflow:
- d. Exception:
- e. Post condition: when the employer accept task request the task is

**9. Create Edit task**

- a. Actors: Employer
- b. Pre-condition:
- c. Workflow:
- d. Exception:
- e. Post condition

**10. Feedback:**

- a. Actors: Employer
- b. Pre-condition :
- c. Workflow:
- d. Exception:
- e. Post condition:

**11. Send Complaint**

- a. Actors: Employer, Freelancer
- b. Pre-condition:
- c. Workflow:
- d. Exception:
- e. Post condition :

**12. Apply task:**

- a. Actors: Freelancer
- b. Pre-condition:
- c. Workflow:
- d. Exception:
- e. **Post condition :**

**13. Cancel App**

- a. Actors: Freelancer.
- b. Pre-condition: freelancer should be applied this task.
- c. Workflow: freelancer clicks cancel task.
- d. Exception:
- e. Post condition: the task is removed from freelancer tasks.

**14. Cancel Submitting**

- a. Actors: Freelancer
- b. Pre-condition: the freelancer should be applying this task and the Employer accept this task.
- c. Workflow: the freelancer click cancel submit for the task.  
The system penalty the freelancer by consume his rate.
- d. Exception:
- e. Post condition: the task is removed from freelancer tasks.

**15. Review History**

- a. Actors: all users should be able to review their history.
- b. Pre-condition: have an account, login.
- c. Exception: no history .
- d. Workflow: user click view history button .
- e. Post condition : user show his history.

**16. Replay complaint**

- a. Actors: Admin.
- b. Pre-condition:.....
- c. Workflow:
- d. Exception:
- e. Post condition:

**17. View statistics**

- a. Actors: Admin.
- b. Pre-condition: admin is logging in.
- c. Workflow: admin click view statistics button.
- d. Post condition: statistics appear .

**18. Remove Account**

- a. Actors: Admin.
- b. Pre-condition: admin is logging in.
- c. Workflow: admin open user profile and click remove account.
- d. Post condition: the user is deleted by the admin.

**19. Receive complaint**

- a. Actors: Admin.
- b. Pre-condition:
- c. Workflow:
- d. Exception:
- e. Post condition:

**20. View report**

- a. Actors: Admin.
- b. Pre-condition: admin is logging in.
- c. Workflow: admin click view report button.
- d. Exception:
- e. Post condition: admin show report of the system

**21. Ban Account**

- a. Actors: Admin.
- b. Pre-condition: admin is logging in.
- c. Workflow: admin open user profile and click ban button.
- d. Exception:
- e. Post condition: the user is banned from system.

# Non Functional Account

## Usability:

Prioritize the important functions of the system based on usage patterns. **Frequently used functions should be tested for usability**, as should complex and critical functions. Be sure to create a requirement for this.

## Reliability:

**Users have to trust the system**, even after using it for a long time. Your goal should be a long MTBF (mean time between failures). Create a requirement that data created in the system will be retained for a number of years without the data being changed by the system. It's a good idea to also include requirements that make it easier to monitor system performance.

## Performance:

What should system response times be, as measured from any point, under what circumstances?  
Are there specific peak times when the load on the system will be unusually high? **Think of stress periods**, for example, at the end of the month or in conjunction with payroll disbursement.

## Supportability:

The system needs to be **cost-effective to maintain**. Maintainability requirements may cover diverse levels of documentation, such as system documentation, as well as test documentation, e.g. which test cases and test