Work separation:

1. Hailei Yu

Task	Hours
Enable running user code on server	6
Detect compiler errors and runtime errors in user code	6
Detect timeout detection in user code	2
Separate oj_worker from main logic (for security reasons)	3
Dockerize oj_worker	16
Deploy oj_worker as a RESTful service	4
Use nginx as the load balancer	3
Enable static user code check for security	4
Enable the syntax highlight in the coding area	6
Clean and separate views	6
On-call duty (Site reliability)	varies

2. Jiaxi Xiong

Task	Hours
Problem model, adding fields with increasing needs	4
Problem management dashboard: enable add/edit problem	8
Problem management dashboard: internal test of problem	3
Problem management dashboard: visibility of problem	3
Access Control: two level user module - Admin and Normal user	6
Highlight optimization: integrate syntax highlight into django form	5
User problem list: pagination, live filter problem	2
Problem submmision: ajax submission, restore code, change coding language, parse submission result and render it on page.	5
Make UI uniform	10
Front-end clean	10
Global support :)	varies

3. Jing Li

Tasks	Hours
User registration, login/logout and UserProfile creation/edit	6
Configure the code from Grumblr to Bearcode	varies
User password reset/change	3
Submission history & Create front-end and forms for the history	7
Discussion board for each problem & Create a grumblr-style small forum for each problem as related discussion	4
Enable users to view all the statistics, acceptance rate, etc.	6
Enable users to randomly pick a problem to start	2
Show a ranking board of users ranked by acceptance rate	3
Problem history for each user per each problem	4
Search for topics in discussion (in both title and text)	4
Views separation & add comment & clean up the backend	7
Create problem data and corresponding test cases	6