

Problem : Design a job searching and posting platform

Thing(Object)/Entity:

Information(Data)/(State):

Services(Behaviors)/Actions

Thing:

User:

Data:name, id, email, resume

Behaviors:

Job:

Data:title, requirement, category, postTime, hiring

Behaviors: isOpen(), apply()

SearchingService:

Data:

Behaviors: search{searchByTitle(), searchByCategory(),

searchByPostTime()}

PostingService:

Data:

Behaviors: createNewJobPost()

LoginToPlatformAuthService

Sequence of invoking behaviors on objects:

User user

Job job

SearchingService: searchService

PostingService: postService

user.LoginToPlatformAuthService -> platform: authorize

if (authorize is true):

Switch:

case Search:

SearchService.searchByTitle -> title:

collection of jobs ordered by title match score

//SearchService.searchByCategory ->

category: collection of jobs belongs to the input category

//SearchService.searchByPostTime ->

beginTime, endTime: collection of jobs between the time range

Loop: (returned job collection)

if job.isOpen():

job.apply() ->

user.name, user.resume, user.email

else:

continue;

case Post:

PostingService.createNewJobPost ->

user.name, title, requirement, category, postTime(currentTime)

else

user.canNotPostOrSearch()