

## Class Diagram

```
1 @startuml
2 'https://plantuml.com/class-diagram
3
4 class User {
5     -String firstName
6     -String lastName
7     -String email
8     -String username
9     -String encryptedPassword
10    -Address address
11
12    + User(String firstName, String lastName, String email, String username, String
encryptedPassword, Address address)
13
14    + String getFirstName()
15    + String getLastName()
16    + String getEmail()
17    + String getUsername()
18    + String getEncryptedPassword()
19    + Address getAddress()
20 }
21
22 class Employer {
23     -String license
24     -String registration
25
26     + Employer(String firstName, String lastName, String email, String username,
String encryptedPassword, Address address, String license, String registration)
27
28     + String getLicense()
29     + String getRegistration()
30 }
31
32 class AuthHandler {
33     -Database database
34     +User handle(username, password)
35 }
36
37 AuthHandler *-- Database
38
39 class JobSeeker {
40     -WorkExperience[] experiences
41     -String[] skills
42     -String[] educations
43
44     + JobSeeker(String firstName, String lastName, String email, String username, String
encryptedPassword,
45     Address address, WorkExperience[] experiences, String[] skills, String[] educations
)
46
47     + WorkExperience[] getExperiences()
48     + String[] getSkills()
49     + String[] getEducations()
50 }
```

## Class Diagram

```
51
52 User <|-- Employer
53 User <|-- JobSeeker
54
55 class WorkExperience {
56     -String jobTitle
57     -Employer employer
58     -Date startDate
59     -Date endDate
60     -String responsibility
61
62     + WorkExperience(String jobTitle, Employer employer, Date startDate, Date endDate,
String responsibility)
63     + String getJobTitle()
64     + Employer getEmployer()
65     + Date getStartDate()
66     + Date getEndDate()
67     + String getResponsibility()
68 }
69
70 JobSeeker "1" *-- "0..*" WorkExperience
71 WorkExperience o-- Employer
72
73 enum EmpType {
74     FULL-TIME
75     PART-TIME
76     INTERNSHIP
77 }
78
79 enum LocType {
80     REMOTE
81     HYBRID
82     ONSITE
83 }
84
85 class JobType {
86     -EmpType empType
87     -LocType locType
88
89     + JobType(EmpType empType, LocType locType)
90     + EmpType getEmpType()
91     + LocType getLocType()
92 }
93
94 JobType -- EmpType
95 JobType -- LocType
96
97 class Job {
98     -Long id
99     -String title
100     -String desc
101     -String salary
102     -String[] skills
```

## Class Diagram

```
103     -String[] requirements
104     -String[] responsibilities
105     -JobType jobType
106     -Employer employer
107     -Date expiresAt
108
109     +Job(String title, String desc, String salary, String[] skills, String[] requirements,
110         String[] responsibilities, JobType jobType, Employer employer, Date expiresAt)
111     + Long getId()
112     + String getTitle()
113     + String getDesc()
114     + String getSalary()
115     + String[] getSkills()
116     + String[] getRequirements()
117     + String[] getResponsibilities()
118     + JobType getJobType()
119     + Employer getEmployer()
120 }
121
122 Job *-- JobType
123
124 Employer "1" *-- "0..*" Job
125
126 class JobFactory <<FlyWeight>> {
127     -static Map<String, JobType>
128
129     +Job getJob(String title, String desc, String salary, Employer employer,
130         String[] requirements, EmpType empType, LocType locType, Date expiresAt)
131 }
132
133 JobFactory "1" *-- "0..*" JobType
134
135 class JobBuilder <<Builder>> {
136     -String title
137     -String desc
138     -String salary
139     -String[] skills
140     -String[] requirements
141     -String[] responsibilities
142     -EmpType empType
143     -LocType locType
144     -Date expiresAt
145     -JobFactory jobFactory
146
147     +JobBuilder setTitle(String title)
148     +JobBuilder setDesc(String desc)
149     +JobBuilder setDesc(String salary)
150     +JobBuilder addSkill(String skill)
151     +JobBuilder addRequirement(String req)
152     +JobBuilder addResponsibility(String res)
153     +JobBuilder setEmpType(EmpType empType)
154     +JobBuilder setLocType(LocType locType)
155     +JobBuilder setExpiresAt(Date expiresAt)
```

## Class Diagram

```
156     +Job build()
157 }
158
159 JobBuilder *-- JobFactory
160
161 class JobApplication {
162     -Long id
163     -Job job
164     -JobSeeker applicant
165     -File resume
166     -AplState state
167
168     +JobApplicationState getState()
169     +void setState(JobApplicationState state)
170     +void moveToNextStep()
171 }
172
173 Job "1" *-- "0..*" JobApplication
174 JobSeeker "1" *-- "0..*" JobApplication
175
176 interface JobApplicationState <<State>> {
177     +void moveToNextStep()
178 }
179
180 class NewState {
181     -JobApplication application
182     +void moveToNextStep()
183 }
184
185 JobApplicationState <|-- NewState
186 NewState *-- JobApplication
187
188 class AcceptState {
189     -JobApplication application
190     +void moveToNextStep()
191 }
192
193 JobApplicationState <|-- AcceptState
194 AcceptState *-- JobApplication
195
196 class RejectState {
197     -JobApplication application
198     +void moveToNextStep()
199 }
200
201 JobApplicationState <|-- RejectState
202 RejectState *-- JobApplication
203
204 class ClosedState {
205     -JobApplication application
206     +void moveToNextStep()
207 }
208
```

## Class Diagram

```
209 JobApplicationState <|-- ClosedState
210 ClosedState *-- JobApplication
211
212
213 class Database <<Singleton>> {
214     -User[] users
215     -Job[] jobs
216     -JobApplication[] applications
217
218     +static Database getInstance()
219 }
220
221 class CalPolyJobServer {
222     -Database database
223     -FileStorage fileStore
224     -RecommendationEngine recommender
225
226     +JobIterator getJobs(Users user, SearchFilter[] filters)
227     +Job getJob(Long jobId)
228     +Job updateJob(Job job)
229     +Job storeJob(Job job)
230     +Bool deleteJob(Long jobId)
231
232     +Users getUsers()
233     +User getUser(Long userId)
234     +User updateUser(User user)
235     +User storeUser(User user)
236     +Bool deleteUser(Long UserId)
237
238     +Applications getApplications()
239     +Application getApplication(Long aplId)
240     +Application updateApplication(Application apl)
241     +Application storeApplication(Application apl)
242     +Bool deleteApplication(Long aplId)
243
244     +String uploadFile(File file)
245     +Bool deleteFile(String id)
246 }
247
248 class RecommendationEngine {
249     +Jobs recommendJobs(User user)
250 }
251
252 class FileStorage {
253     +String storeFile(File file)
254 }
255
256 CalPolyJobServer *-- Database
257 CalPolyJobServer *-- FileStorage
258 CalPolyJobServer *-- RecommendationEngine
259
260 Database *-- User
261 Database *-- Job
```

## Class Diagram

```
262 Database *-- JobApplication
263
264
265 interface Iterator<T> <<Iterator>>{
266     +bool hasNext()
267     +T getNext()
268 }
269
270 interface JobIteratorByUser <<Iterator>>{
271     +JobIterator getJobs(User user, Filter[] filters)
272 }
273
274 JobIteratorByUser <|-- CalPolyJobServer
275
276 class JobIterator<Job> {
277     -int size
278     -int currentIdx
279     -Jobs[] jobs
280
281     +JobIterator(Jobs[] jobs)
282
283     +bool hasNext()
284     +Job getNext()
285 }
286 Iterator <|-- JobIterator
287 JobIteratorByUser -- JobIterator
288
289 @enduml
```