

Sequence Diagrams

```
1 @startuml
2 'https://plantuml.com/sequence-diagram
3
4 'Sequence Diagram to create jobs
5
6 autonumber
7
8 actor Employer
9 participant Client<<Web/Mobile>>
10 participant Backend<<Servers>>
11 participant Database
12
13 Employer -> Client: Request home page
14 Client -> Backend: Request home page
15 Backend -> Client: Home page
16 Client -> Employer: Show Home Page
17 Employer -> Client: Click "Create Job" button
18 Client -> Backend: Request job creation form
19 return Job creation form
20 Client -> Employer: Display job creation form
21
22 Employer -> Client: Fill up application details
23 Employer -> Client: Click "Submit" button
24 Client -> Backend: Send the form
25 Backend -> Backend: Perform validation
26 alt #eafaf1 Validation Success
27     Backend -> Database: Insert job in database
28     alt #ebf5fb Insertion Success
29         Database -> Backend: Insert job successfully
30         Backend -> Client: Send job creation success message
31         Client -> Employer: Display job creation success message
32     else Insertion Failure
33         Database -> Backend: Job insertion failure
34         Backend -> Client: Send job insertion failure message
35         Client -> Employer: Display job insertion failure message
36     end
37 else Validation Failure
38     Backend -> Client: Send validation error message
39     Client -> Employer: Display validation error message
40 end
41
42 @enduml
```

Sequence Diagrams

```
1 @startuml
2 'https://plantuml.com/sequence-diagram
3
4 'Sequence Diagram to browse and apply jobs
5 autonumber
6
7 actor JobSeeker
8 participant Client<<Web/Mobile>>
9 participant Backend<<Servers>>
10 participant Database
11 participant FileStorage
12
13 JobSeeker -> Client: Go to home page
14 Client -> Backend: Go to home page
15 return Home page
16 Client->JobSeeker: Home page
17 JobSeeker -> Client: Enter search keywords & filters
18 JobSeeker -> Client: Click "Search" button
19 Client -> Backend: Job search query
20 Backend->Database: Retrieve Jobs using search query
21 Database->Backend: Job list
22 Backend -> Client: Job list
23 Client->JobSeeker: Show job list at left panel
24 loop#85c1e9 #f8f9f9 Job list
25     JobSeeker -> Client: Check <b>i</b>th job summary
26     alt #eafaf1 if job seeker want to know more
27         JobSeeker -> Client: Click <b>i</b>th job's title
28         Client -> Backend: Request the <b>i</b>th job
29         Backend -> Database: Query the <b>i</b>th job
30         Database -> Backend: Return <b>i</b>th job's data
31         Backend -> Client: Return <b>i</b>th job's detail
32         Client -> JobSeeker: Show <b>i</b>th job's detail at right panel
33         alt #ebf5fb interested to apply
34             JobSeeker -> Client: Click "Apply" button
35             Client -> Backend: Request job application form
36             Backend -> Client: Job application form
37             Client -> JobSeeker: Display job application form
38             JobSeeker -> Client: Upload resume
39             Client -> Backend: Upload resume
40             Backend -> FileStorage: Store resume
41             return storage link/id
42             Backend -> Client: storage link/id
43             Client->JobSeeker: Display resume uploaded successfully
44
45             JobSeeker -> Client: Fill up application form
46             JobSeeker -> Client: Submit application
47             Client -> Backend: Send application
48             Backend->Backend: validate application
49             alt #eaecee Validation Success
50                 Backend -> Database: Insert application
51                 Database -> Backend: Send id and success message
52                 Backend -> Client: Send id and success message
53                 Client -> JobSeeker: Show success message
```

Sequence Diagrams

```
54         else Failure
55             Backend -> Client: Sent error message
56             Client -> JobSeeker: Show error message
57         end
58     end
59 end
60 end
61
62 @enduml
```