

# PURSUITER

## System Design Document

Katherine Pravdin  
Elena Manneh  
Davit Iskandaryan  
Mohammad Al-Qasem

# Table of Contents

<b>Dependencies</b>	<b>3</b>
<b>High-Level Description of Classes</b>	<b>4</b>
<b>System Architecture</b>	<b>9</b>
<b>State Diagram</b>	<b>10</b>
<b>CI/CD Pipeline</b>	<b>11</b>
Continuous Integration	11
Continuous Deployment	11

# Dependencies

## Front End:

- React
- Axios
- JS-cookie

## Back End:

- Mongoose
- Express
- Node
- Cors
- Bcrypt

## Database:

- MongoDB Atlas

## CI/CD

- GitLab CI
- Docker
- Docker Hub

The assumption is that the user will be on a web browser across any device.

# High-Level Description of Classes

<b>Class Name:</b> LandingPage	
<b>Parent Class (if any):</b> N/A	<b>Subclasses (if any):</b> N/A
<b>Responsibilities:</b> <ul style="list-style-type: none"><li>• Display the landing page for the applications.</li><li>• Provide navigation to signup or login</li></ul>	
<b>Collaborators:</b> <ul style="list-style-type: none"><li>• <code>useNavigate</code> from <code>react-router-dom</code> for navigation.</li></ul>	

<b>Class Name:</b> SignupPage	
<b>Parent Class (if any):</b> N/A	<b>Subclasses (if any):</b> ApplicantSignupPage, RecruiterSignupPage
<b>Responsibilities:</b> <ul style="list-style-type: none"><li>• Render signup page.</li><li>• Handles signup form submission.</li></ul>	
<b>Collaborators:</b> <ul style="list-style-type: none"><li>• <code>UserContext</code> for user data and authentication.</li><li>• <code>UserController</code> for backend API interactions.</li></ul>	

<b>Class Name:</b> ApplicantSignupPage	
<b>Parent Class (if any):</b> SignupPage	<b>Subclasses (if any):</b> N/A
<b>Responsibilities:</b> <ul style="list-style-type: none"><li>• Render signup page specifically for job-seekers.</li><li>• Handles signup form submission for job-seekers.</li></ul>	
<b>Collaborators:</b> <ul style="list-style-type: none"><li>• <code>SignupPage</code> for common signup functionality.</li><li>• <code>UserContext</code> for user data and authentication.</li><li>• <code>useNavigate</code> from <code>react-router-dom</code> for navigation.</li></ul>	

<b>Class Name:</b> RecruiterSignupPage	
<b>Parent Class (if any):</b> SignupPage	<b>Subclasses (if any):</b> N/A
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>• Store recruiter information in the database</li> </ul>	
<b>Collaborators:</b> <ul style="list-style-type: none"> <li>• <code>SignupPage</code> for common signup functionality.</li> </ul>	

<b>Class Name:</b> LoginPage	
<b>Parent Class (if any):</b> N/A	<b>Subclasses (if any):</b> N/A
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>• Render the login page for the application.</li> <li>• Navigate to the appropriate dashboard based on the user role.</li> </ul>	
<b>Collaborators:</b> <ul style="list-style-type: none"> <li>• <code>useNavigate</code> from <code>react-router-dom</code> for navigation.</li> <li>• <code>UserContext</code> for user data and authentication.</li> <li>• <code>UserController</code> for backend API interactions to authenticate the user.</li> <li>• <code>Link</code> from <code>react-router-dom</code> for navigation to signup pages.</li> </ul>	

<b>Class Name:</b> Dashboard	
<b>Parent Class (if any):</b> N/A	<b>Subclasses (if any):</b> ApplicantDashboard, RecruiterDashboard
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>• Display the user's dashboard with relevant information and options</li> <li>• Allow the user to update their profile information</li> <li>• Provide access to job search or posting functionalities</li> </ul>	
<b>Collaborators:</b> <ul style="list-style-type: none"> <li>• <code>UserContext</code> for user data and authentication.</li> <li>• <code>useNavigate</code> from <code>react-router-dom</code> for navigation.</li> <li>• <code>useState</code>, <code>UseContext</code>, and <code>UseEffect</code> from "<code>react</code>"</li> <li>• <code>Modal</code> from "<code>./Modal</code>"</li> <li>• <code>FaStar</code> from "<code>./react-icons/fa</code>"</li> </ul>	

<b>Class Name:</b> ApplicantDashboard	
<b>Parent Class (if any):</b> Dashboard	<b>Subclasses (if any):</b> N/A
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>• Display applicant-specific information (e.g. saved job listings, application status)</li> <li>• Allow the applicant to update their resume</li> </ul>	
<b>Collaborators:</b> <ul style="list-style-type: none"> <li>• <code>React</code> from <code>"react"</code></li> <li>• <code>fetchFavoritedJobs</code>, <code>fetchJobsForRecruiter</code> from <code>"../Dashboard"</code></li> </ul>	

<b>Class Name:</b> RecruiterDashboard	
<b>Parent Class (if any):</b> Dashboard	<b>Subclasses (if any):</b> N/A
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>• Display recruiter-specific information (e.g. posted job listings, applicant profiles)</li> <li>• Allow the recruiter to update their or job postings</li> <li>• Provide access to applicant search functionalities</li> </ul>	
<b>Collaborators:</b> <ul style="list-style-type: none"> <li>• <code>React</code> from <code>"react"</code></li> <li>• <code>fetchFavoritedJobs</code>, <code>fetchJobsForRecruiter</code> from <code>"../Dashboard"</code></li> </ul>	

<b>Class Name:</b> UserInformation	
<b>Parent Class (if any):</b> N/A	<b>Subclasses (if any):</b> N/A
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>• Store and manage user information (e.g. username, email)</li> <li>• Provide methods for updating user information</li> </ul>	
<b>Collaborators:</b> <ul style="list-style-type: none"> <li>• <code>UserContext</code> for user data and authentication.</li> <li>• <code>useNavigate</code> from <code>react-router-dom</code> for navigation.</li> <li>• <code>useState</code>, <code>UseContext</code>, and <code>UseEffect</code> from <code>"react"</code></li> </ul>	

<b>Class Name:</b> ApplicantList	
<b>Parent Class (if any):</b> N/A	<b>Subclasses (if any):</b> N/A
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>• Fetch and display a list of applicants for a specific job.</li> <li>• Display job details for a specific job.</li> <li>• Filter applicants based on a search term.</li> <li>• Handle applicant favoriting.</li> <li>• Display details and resume of the selected applicant.</li> </ul>	
<b>Collaborators:</b> <ul style="list-style-type: none"> <li>• <code>useParams</code> and <code>useNavigate</code> from <code>react-router-dom</code> for navigation.</li> <li>• <code>UserContext</code> for user data and authentication.</li> <li>• <code>DashboardController</code> for backend API interactions to fetch applicants and job details.</li> <li>• <code>FaStar</code> from <code>react-icons</code> for the favorite icon.</li> </ul>	

<b>Class Name:</b> Modal	
<b>Parent Class (if any):</b> N/A	<b>Subclasses (if any):</b> N/A
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>• Display a modal window with the provided content.</li> <li>• Handle the visibility of the modal based on the <code>show</code> prop.</li> </ul>	
<b>Collaborators:</b> N/A	

<b>Class Name:</b> NavBar	
<b>Parent Class (if any):</b> N/A	<b>Subclasses (if any):</b> N/A
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>• Render the navigation bar for the application.</li> <li>• Display different navigation links based on user authentication status and role.</li> <li>• Handle user logout and navigate to the home page.</li> </ul>	
<b>Collaborators:</b> <ul style="list-style-type: none"> <li>• <code>Link</code> from <code>react-router-dom</code> for navigation links.</li> <li>• <code>useNavigate</code> from <code>react-router-dom</code> for programmatic navigation.</li> <li>• <code>useLocation</code> from <code>react-router-dom</code> for determining the current route.</li> <li>• <code>UserContext</code> for accessing user data and authentication functions.</li> </ul>	

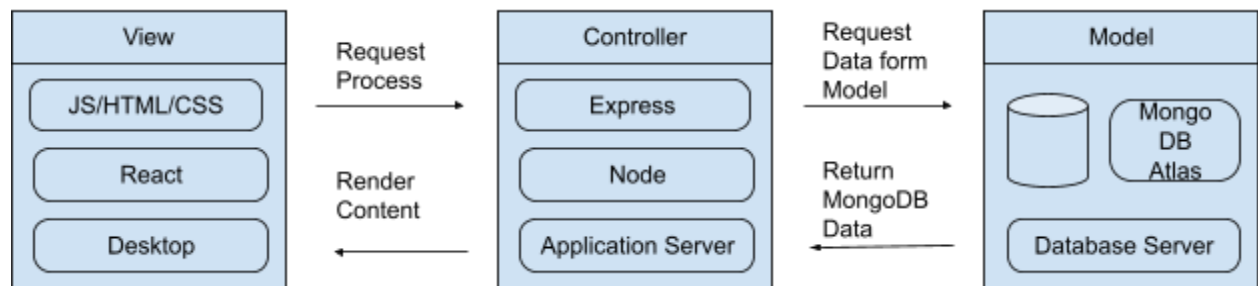
<b>Class Name:</b> UserContext	
<b>Parent Class (if any):</b> N/A	<b>Subclasses (if any):</b> N/A
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>• Manage user state</li> <li>• Provide functions to login, signup, update, and logout user</li> <li>• Persist user state in cookies</li> </ul>	
<b>Collaborators:</b> <ul style="list-style-type: none"> <li>• <code>React</code> from <code>"react"</code></li> <li>• <code>UserContext</code> for user data and authentication.</li> <li>• <code>UserController</code> for backend API interactions.</li> <li>• <code>Cookies</code> from <code>"js-cookie"</code>.</li> </ul>	

<b>Class Name:</b> DashboardController	
<b>Parent Class (if any):</b> N/A	<b>Subclasses (if any):</b> N/A
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>• Fetch all jobs</li> <li>• Handle job posting, editing, deleting, and applying</li> <li>• Handle favouriting of jobs</li> </ul>	
<b>Collaborators:</b> <ul style="list-style-type: none"> <li>• <code>axios</code> from <code>"axios"</code>.</li> </ul>	

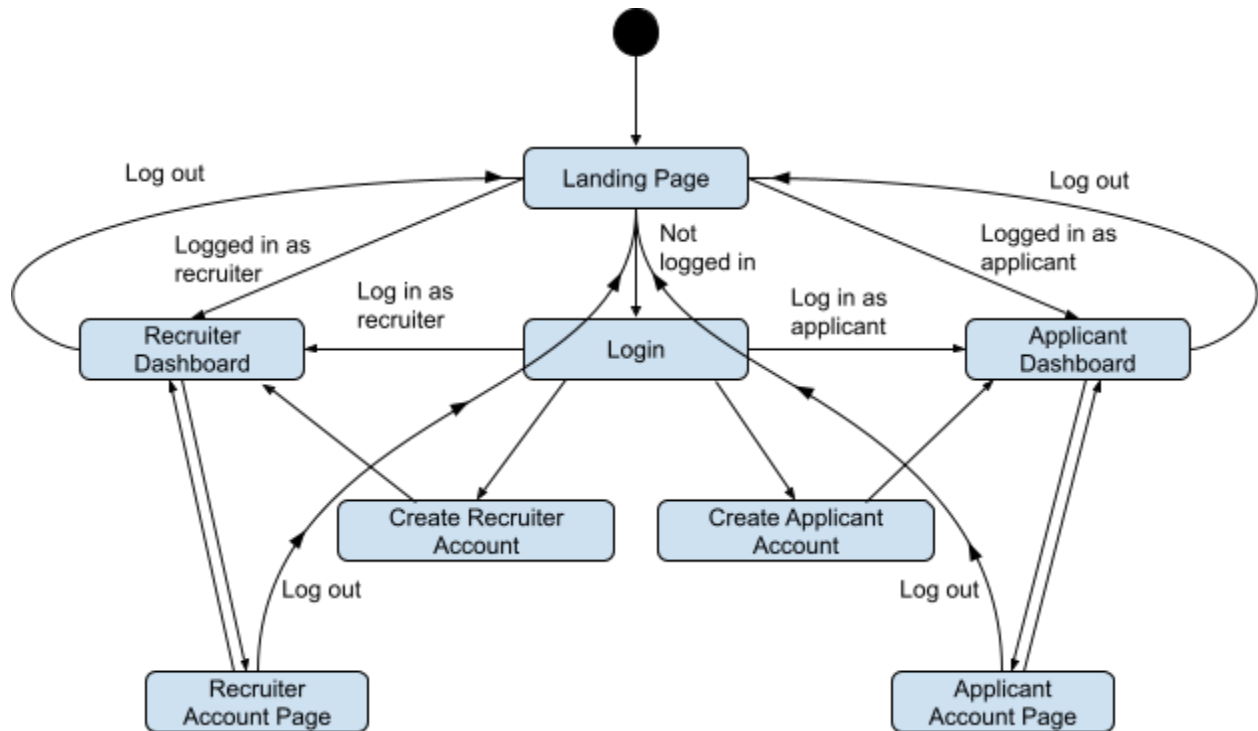
<b>Class Name:</b> UserController	
<b>Parent Class (if any):</b> N/A	<b>Subclasses (if any):</b> N/A
<b>Responsibilities:</b> <ul style="list-style-type: none"> <li>• Login user</li> <li>• Signup User</li> <li>• Update User information</li> </ul>	
<b>Collaborators:</b> <ul style="list-style-type: none"> <li>• N/A</li> </ul>	



# System Architecture



# State Diagram



# CI/CD Pipeline

## Continuous Integration

Pursuiter utilizes GitLab CI for continuous integration. The CI pipeline is triggered when code is pushed into the repository with the following steps:

- 1) Build
  - Build the docker image using the Dockerfiles.
- 2) Test
  - Runs automated tests using the built Docker image.
- 3) Push
  - Push the image to Docker Hub(mohammadalqasem/pursuiter-frontend & mohammadalqasem/pursuiter-backend)

## Continuous Deployment

The process pulls the latest image from Docker Hub and deploys it into the production environment with the following steps:

- 1) Pull Image
  - Pull the Docker Image from Docker Hub
- 2) Test
  - Automated tests are set up for the deployed container to ensure its functionality and performance