Cloud Native Project Documentation

Group: **CloudJB** (Jaromir Charles, Benjamin Herrmann)

Table of Contents

Description

The main functionality of our multi-tenancy cloud native application is to allow

- **tenants** (mainly the personnel managers) of said firm to organize their jobs and workers. They have the ability to invite employees to their company's PMS. The ability to create, delete and edit jobs as well as assign workers to a specific job is also permitted.
- **employees** to register to the PMS of their employer. With granted access, they have the ability apply for jobs of interest. The system also gives them a complete overview of the upcoming, applied, upcoming and completed jobs.

A better understanding of the functionality of the application can be seen within the following user stories.

Tenant: Register, unregister to PMS

As a tenant I would like to register/unregister my company to PMS.

Acceptance Criteria:

1. Possibility to register my company to use PMS's services.
2. Possibility to unregister my company to no longer use PMS's services.

Tenant: Invite workers

As a Tenant I would like to invite workers to join the company's "worker group" and also have an overview of the invited workers status.

Acceptance Criteria:

1. Text field to enter email addresses
2. Invite button to send invitation to the entered email addresses
3. A view of all workers as well as their invitation status, accepted or request pending.

Tenant: Create job

As a Tenant I would like to be able to create a new job listing., in order to add the job to the system as well as for the employees to be able to apply for the new job.
Acceptance criteria:

1. Creating a job is done with the click of a button `Create Job`.
    1. The ability to add the jobs title, location, description, start & end time, number of workers required.
2. The ability to either `Save` or `Cancel` creating the job.

3. The newly created job will appear in the list of jobs.

Tenant: View and edit jobs

As a Tenant I would like to see a list of all jobs, in order to have an overview of my company's jobs.

Acceptance criteria:

1. A list view of all jobs.
2. The ability to edit a job's information by clicking on the respective edit icon

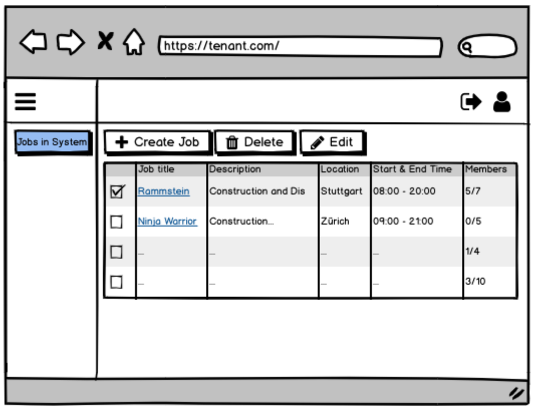
---

Tenant: Delete Job Listing

As a Tenant I would like to delete a job listing, so that employees can no longer apply for the job.

Acceptance criteria:

1. The ability to check mark a job and delete it.
2. The deleted job is no longer in the list of jobs.



	Job title	Description	Location	Start & End Time	Members
☑	Rammstein	Construction and Dis	Stuttgart	08:00 - 20:00	5/7
☐	Ninja Warrior	Construction...	Zürich	09:00 - 21:00	0/5
☐	–	–	–	–	1/4
☐	–	–	–	–	3/10

Tenant: View and accept applicants

As a Tenant I would like to see which employees applied for which job, so that i can create a team to partake in the job.

Acceptance criteria:

1. Each row in the job listing gets a new field named `Members`
    1. Example the field will show `5/7` meaning 5 people applied from needed 7 workers.
2. When job is clicked, redirected to another page with the job's information and two lists with `applied` and `selected` workers.
3. The ability to move the workers between both `applied` and `selected` workers list.

Employee: Join Tenant's PMS service, edit profile

As an Employee I would like to accept my companies invite to use PMS's services, as well as to edit my profile information.

Acceptance criteria:

1. Fill out profile information with name, address, phone number, insurance number
2. View and edit profile information.

Employee: Available Job list

As an Employee I would like to see a list of all available jobs my employing company currently has and the ability to click on them to see all the information, in order to apply for jobs.

<u>Acceptance criteria:</u>

1. See a list of all available jobs under `Available jobs`.
2. The ability to click on the job's title and see the full job's information.
3. The ability to apply for a job with an `Apply` button.
    1. This job will no longer be shown under the `Available jobs` list.

Employee: Applied Job list

As an Employee I would like to see a list of all the jobs I applied to, in order to have a separate overview between all jobs and my applied jobs.

<u>Acceptance criteria:</u>

1. A view `Applied Jobs`
    1. This view contains all the jobs that were applied to in the `Available jobs` list section.
2. The ability to cancel a job application with a `cancel` button.

Employee: Accepted Job list

As an Employee I would like to see a list of all the jobs i got accepted to, in order to know my upcoming jobs.

<u>Acceptance criteria:</u>

1. A view `Upcoming Jobs`.
    1. View shows the upcoming jobs along with the jobs information.
2. The ability to `cancel` a job application with a `cancel` button.

Application Architecture & Design

Describe the components and interfaces of the application

Describe how the most important use cases are implemented (dynamic view)

Describe which resources of the cloud provider you are using, and justify the usage and discuss alternatives

- Google Kubernetes Engine
    - Google's Kubernetes Engine has been used to run the application in a containerized environment. Kubernetes automates deployment, scaling and management of the containerized application.
    - Alternatives
- Firebase Firestore
    - The application uses a NoSQL Document store database because of its flexibility. It is more easy to add new "fields" to these documents as to their SQL counterparts. This makes implementation a lot easier and also makes it more flexible to configure different fields for different tenants.

    - Alternatives

Describe why your application has the five essential characteristics of a cloud service see

1. `On-demand self-service`
    1. A consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service provider.
2. `Broad network access`
    1. Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g., **mobile** phones, tablets, laptops and workstations).
3. `Resource pooling`
    1. The provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand. There is a sense of location independence in that the customer generally has no control or knowledge over the exact location of the provided resources but may be able to specify location at a higher level of abstraction (e.g., country, state or datacenter). Examples of resources include storage, processing, memory and network bandwidth.
4. `Rapid elasticity`
    1. Capabilities can be elastically provisioned and released, in some cases automatically, to scale rapidly outward and inward commensurate with demand. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be appropriated in any quantity at any time.
5. `Measured Service`
    1. Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth and active user accounts). Resource usage can be monitored, controlled and reported, providing transparency for the provider and consumer.

Describe how multi-tenancy is implemented

Multi-tenancy is achieved with our database model. The database separates the data on a tenant level as well as employee level. To ensure that no wrong information is leaked to the wrong tenant or worker, each request made contains the respective company's name when making requests on company level as well as the workers name on employee requests.

Describe why your application is cloud native, does it implement the 12F?

The application implements the twelve cloud native factors. **(1)** The application's `Codebase` is hosted on GitHub

Operations

Adding a new tenant

Installing application on the cloud provider

DevOps approach

As a team of 2 developers, we utilized `GitHub` to manage the changes made to the source code as well as for collaboration. To keep track of what needs to be done within the project, we used GitHubs project boards so that the team has an overview of what needs to be done, what is currently in progress, what needs to be reviewed and what has already been done. **INSERT PICTURE??**.