

# Specification Document MaintControl

Students' names:

Nave Shimoni, Lior Fridman, Reuven Bar, Matan Avital

Moderators' names:

Tal Gumai, Mark Israel

Table of contents:

Project's Goal & system decription.....	page 4
Screens.....	page 5
Server side.....	page 6
Server architecture.....	page 7
Screen details.....	page 8-13
Methodologies.....	page 14-15
Platforms.....	page 16
Database structure & flowchart.....	page 17-19

## 1. Project's Goal:

The goal of the project is to create a generic WEB-based system, which provides a solution to the problem of supervision and control of the maintenance carried out by different companies in different locations in the country, thus optimizing the daily work of the maintenance workers.

## 2. System Description:

### Software application:

Will be developed as a responsive Web application based on REACT, both for computers and mobile phones.

### Data reception/Database:

The method of receiving the data will be carried out on a SQL relational database based on Postgres. These tables will store the parameters that will be reflected in the system, as well as the user details and other settings.

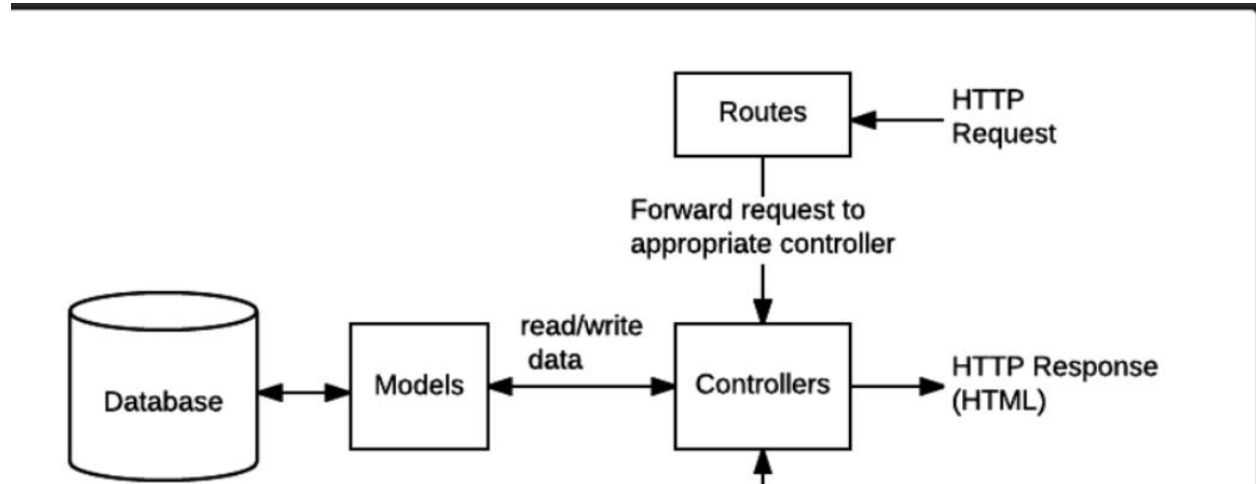
Screens:

- 1) Login screen
- 2) Reset password
- 3) Sign up
- 4) Guides
- 5) Daily missions
- 6) Task page
- 7) Mission's history
- 8) Admin users management
- 9) User's information(location, role, company, etc.)

Server side:

- 1) API:
  - a) Administrator – add/update/delete/get/search users
  - b) Manager – upload/ delete/download/Edit guides
  - c) Manager – add/update/delete/get/search missions
  - d) Manager – add/update/delete/get/search tasks
  - e) Maintenance – comment, add photo, get task, mark task status
  - f) User - login/get user settings/download guides
- 2) Using express middleware
- 3) Connection to PostgreSQL DB on GCP SQL instance
- 4) Using routes and controllers
- 5) Using buckets in GCP to upload and download files and images

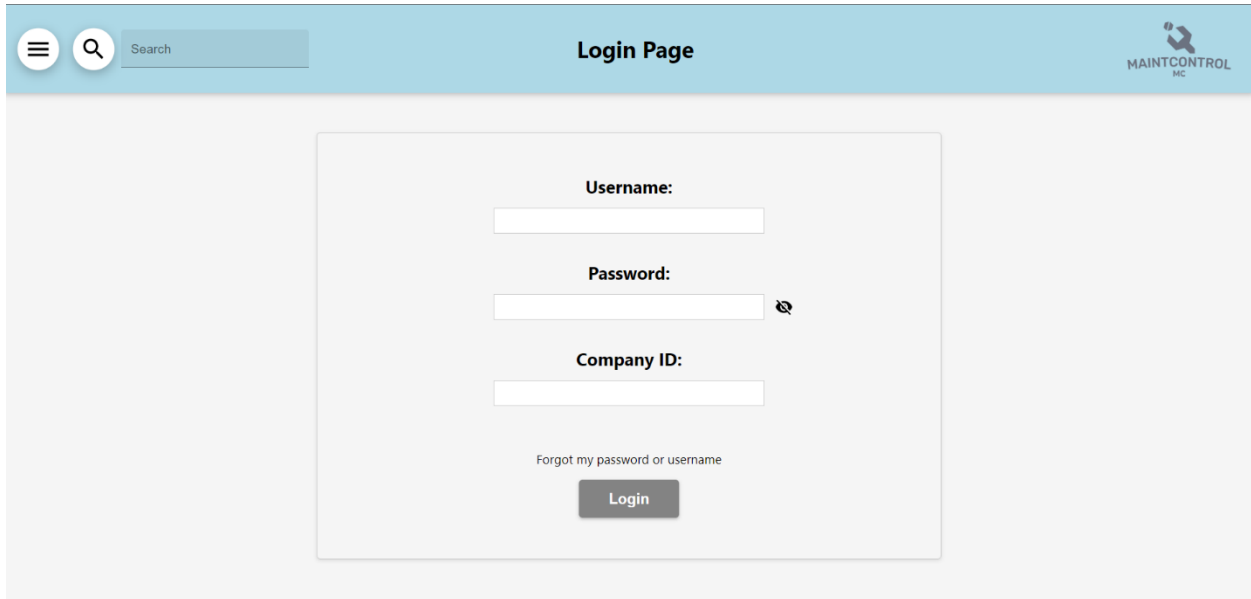
Server architecture:



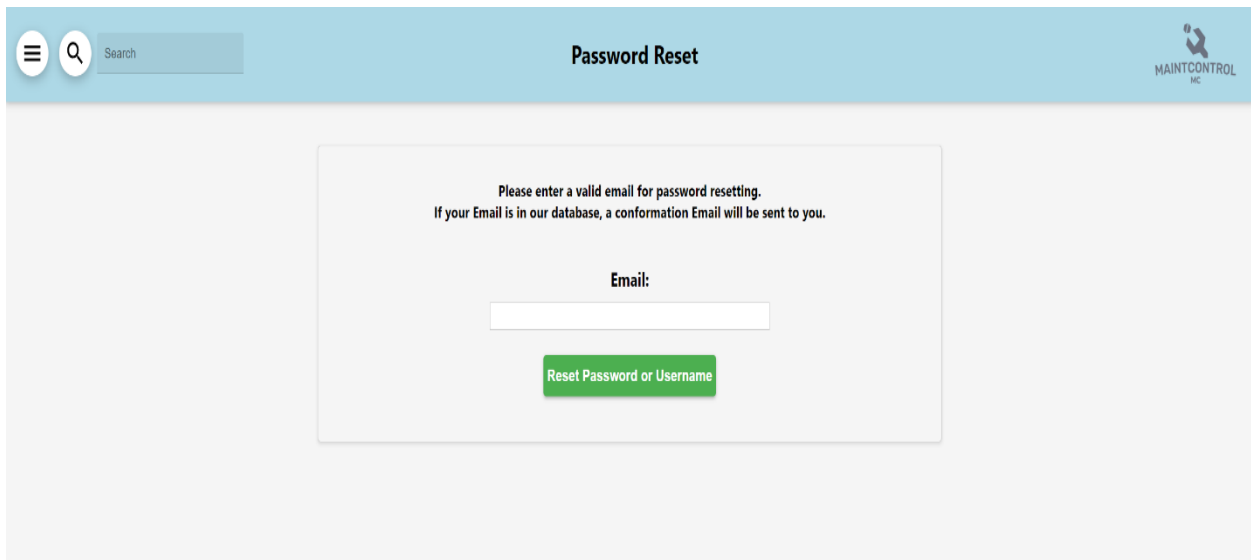
### 3. Screen's details:

#### a) Login screen:

Each user will enter the user information generated for him, a password and a company ID.



The screenshot shows the 'Login Page' interface. At the top, there is a navigation bar with a search icon and the text 'Search'. The page title is 'Login Page'. The main content area contains a form with three input fields: 'Username:', 'Password:', and 'Company ID:'. Below the 'Password:' field is a small eye icon for toggling visibility. Underneath the 'Company ID:' field is a link that says 'Forgot my password or username'. At the bottom of the form is a 'Login' button.



The screenshot shows the 'Password Reset' interface. At the top, there is a navigation bar with a search icon and the text 'Search'. The page title is 'Password Reset'. The main content area contains a form with a single input field labeled 'Email:'. Above the input field, there is a message: 'Please enter a valid email for password resetting. If your Email is in our database, a conformation Email will be sent to you.' Below the input field is a green button labeled 'Reset Password or Username'.



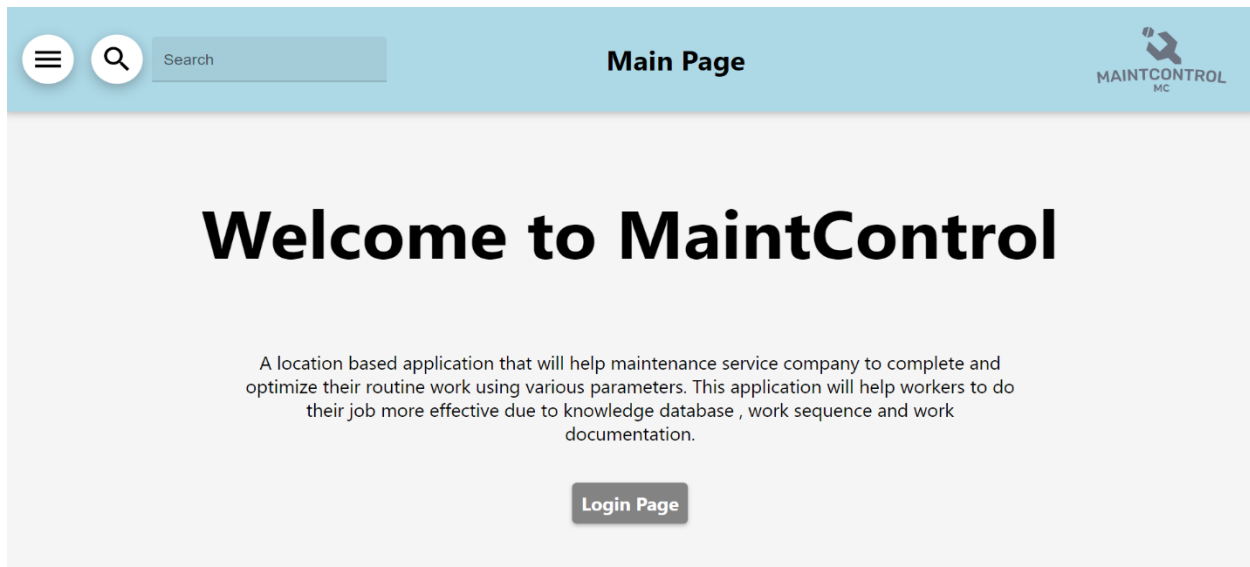
b) Home screen:

The homepage changes according to the type of user (there are 3 types of users):

Maintenance man: screens of Emergency & Daily missions, sub-missions, locations on the map, visit history and guides will appear.

Manager: In addition to the other screens mentioned above, manager will have screens that show the availability of various maintenance personal will appear, as well as the option to edit various guides.

Administrator: In addition to the other screens mentioned above, user management screens will appear (can add users/ update users and their permissions/ delete users).



c) Admin screen:

The admin page is used only by administrators, and it includes list of all users and their roles. In this page the administrator can add/update/delete any user.

Admin page

Search a user...

Administrator  Manager  Maintenance

Users	
lior fridman	administrator
nave shimoni	administrator
dan dany	administrator
mark marky	manager
tal taly	manager
alon alony	manager
matan matani	maintenance
tomer tomeri	maintenance
noa kirel	maintenance
neta barzilay	maintenance



Add a User

<b>Username:</b> Please enter a username	<b>Password:</b> Please enter an 8 characters password
<b>First name:</b> Please enter a first name	<b>Last name:</b> Please enter a last name
<b>Email address:</b> Please enter a valid email	<b>Phone number:</b> Please enter a valid phone number
<b>Living Address:</b> Please enter an address	<b>Geographic Area:</b> Choose Geographic Area
<b>Authorization:</b> Choose authorization	<b>Company ID:</b> Please enter a company id

Add a user

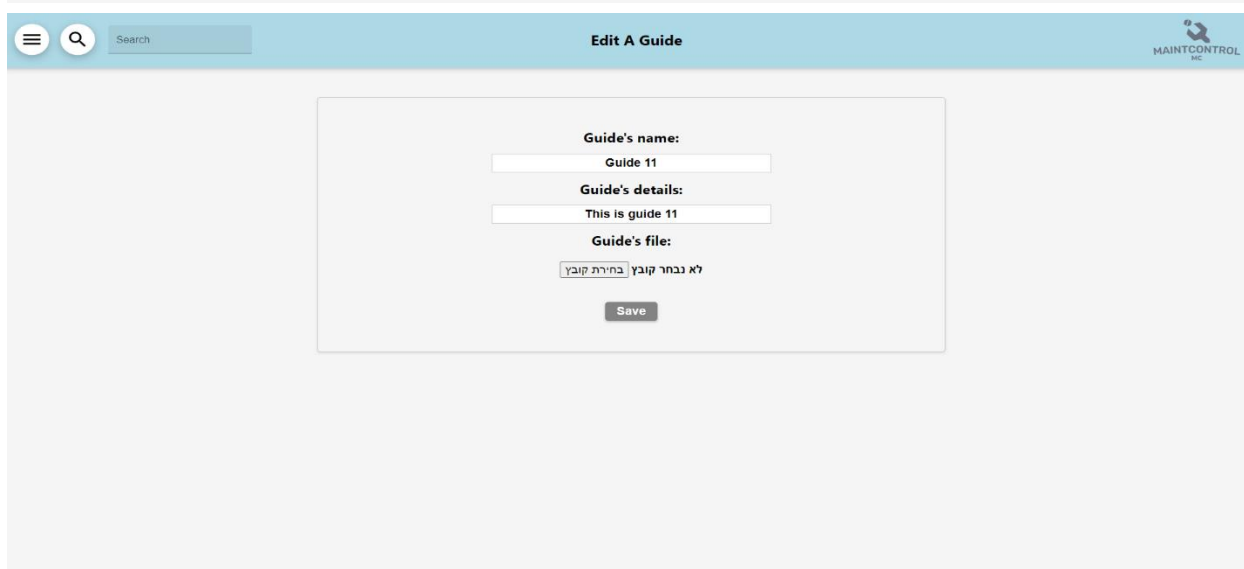
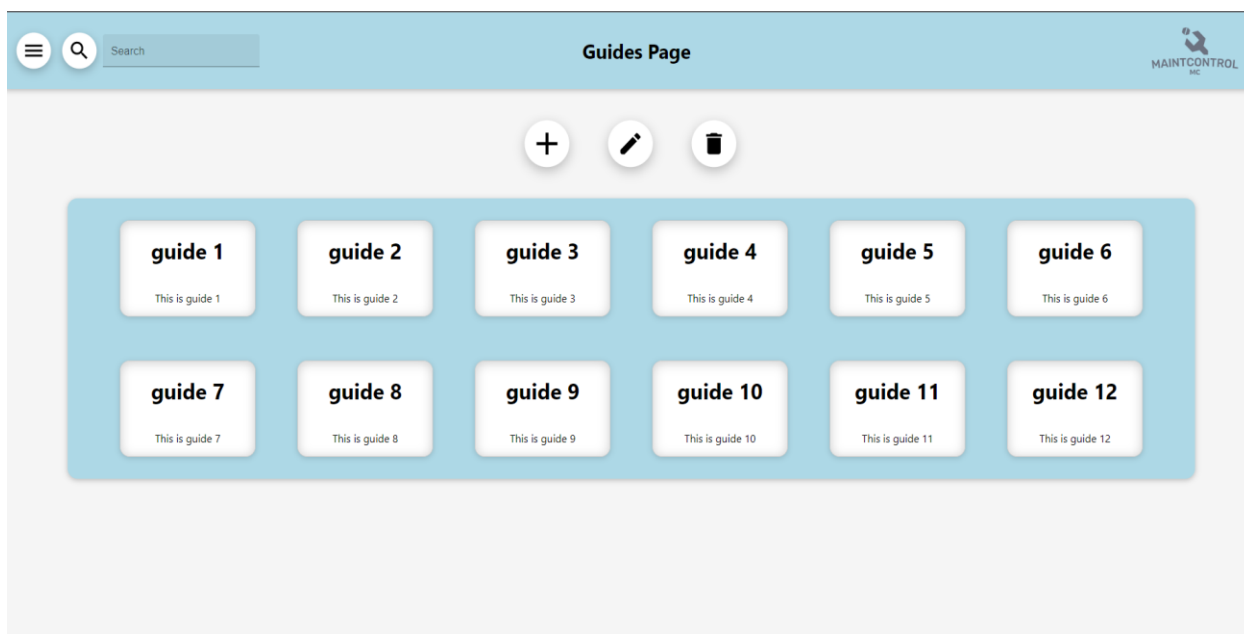
Edit A User

<b>Username:</b> dan	<b>Password:</b> *****
<b>First name:</b> dan	<b>Last name:</b> dani
<b>Email address:</b> dan@email.com	<b>Phone number:</b> 0501234568
<b>Living Address:</b> Please enter an address	<b>Geographic Area:</b> Choose Geographic Area
<b>Authorization:</b> Choose authorization	<b>Company ID:</b> 1

Save

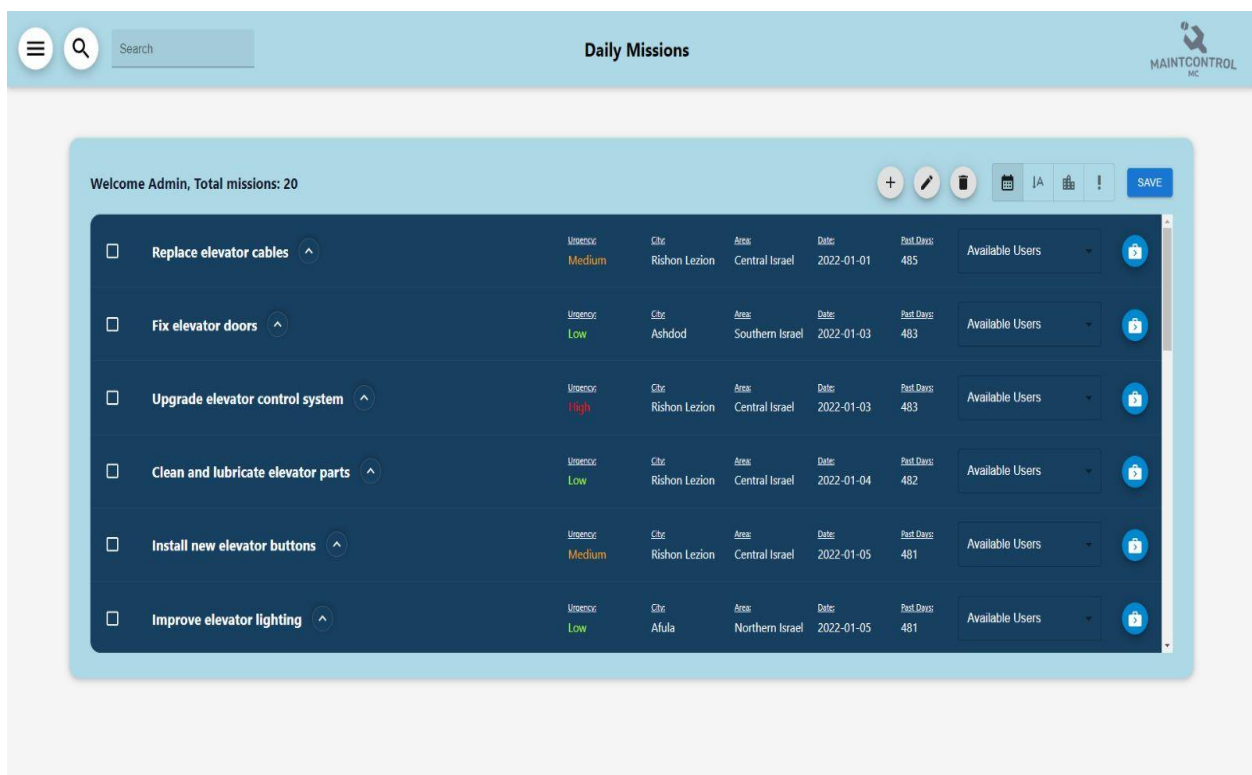
d) Guides screen:

The guides screen includes different guides detailing the procedures for different tasks. The maintenance person will be able to view the manuals without being able to edit or delete them. The administrators will be able to edit the guides as needed and add new ones to the bucket on the GCP. Also, clicking on the guide will take you to the guide page where the general details of the guide and a PDF or WORD file will usually be displayed.



e) Missions & tasks:

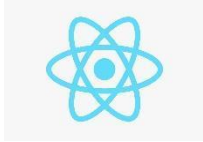
The missions page contains list of missions that needs to be completed and filled in by the maintenance man. Each mission includes all the necessary details, the sub-tasks, location on map and the history about the place.





#### 4. Methodologies:

##### a) Front



React is a JAVASCRIPT based library, used to build various WEB systems. Its flexibility and the convenience of working with it and building an internet system for mobile and different types of computers are the features that led us to choose this FRAMEWORK as opposed to others.

##### b) Backend



We chose Node.js technology following the decision to choose the React FRAMEWORK, and build a powerful and flexible server side quickly. The tool allows you to build different routers and controllers in a quick and convenient way.

We will use Express - backend framework for node.js. It is a middleware that is very convenient to work with in order to create various REST API requests with the server.

c) Development

language:

Backend: The system will be developed in the Node.js development language

Frontend: HTML, CSS, JS-React.

d) Version Control

Working with GitHub.

e) Data Base

Working with Postgres - an excellent tool for building SQL schemas that will serve us when using the coordinates and the API used to integrate the map. Also, SQL tables are a very convenient way to create permissions and user details, and also tables containing the tasks will make their management easier.

## 5. Platforms:

### Development environment:

The system will be developed in the Visual Studio Code development environment. The environment supports debugging, Git view control, syntactic illustration of code fragments, intelligent code completion, and lateral code changes.

We also use docker, docker-compose for uploading images in development and also use them on the GCP.

For working with PostgreSQL tables we use pgAdmin.

### Deployment environment:

The system will be deployed on GCP (Google Cloud Platform).

GCP is a powerful environment that includes the following things:

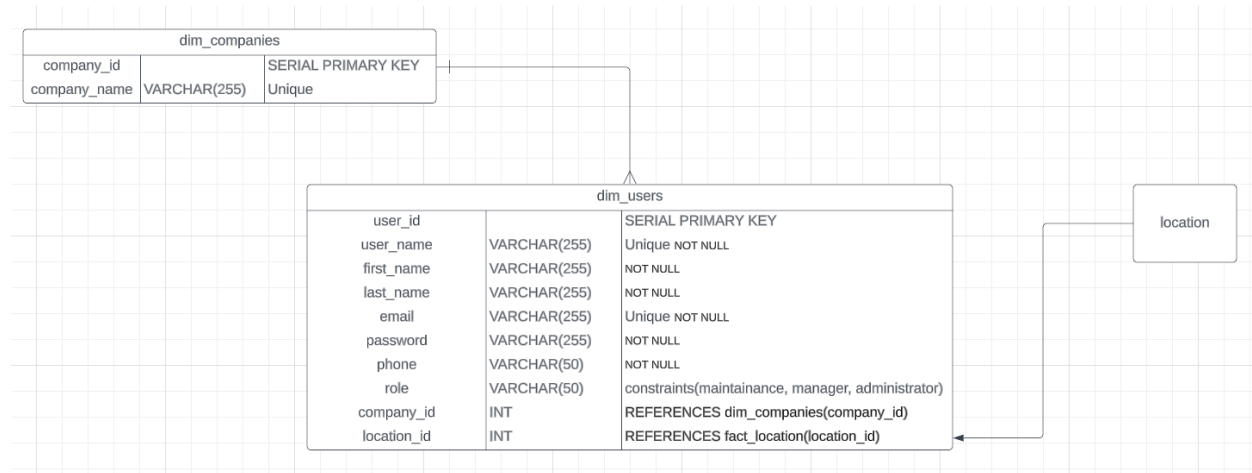
- 1) Container registry - upload docker images.
- 2) Buckets – cloud storage for uploading and downloading files
- 3) Secret manager – handling configuration (environment variables and secret files)
- 4) VPC – configure networks on cloud
- 5) SQL instances – provides environment to create databases in different types (we used PostgreSQL).

And many more features.

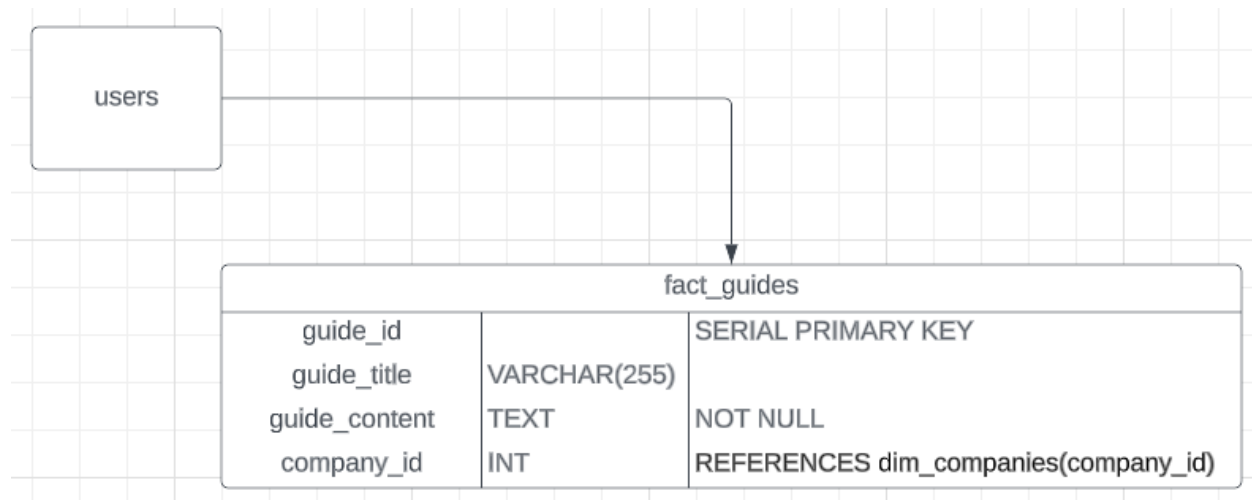


## 6. Database Structure:

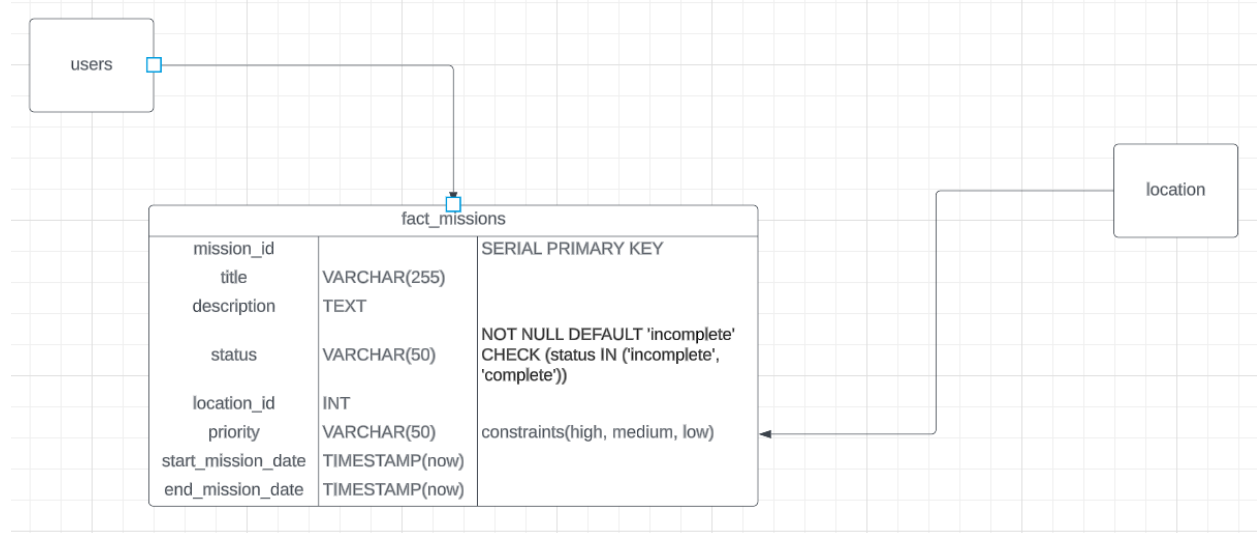
### Users:



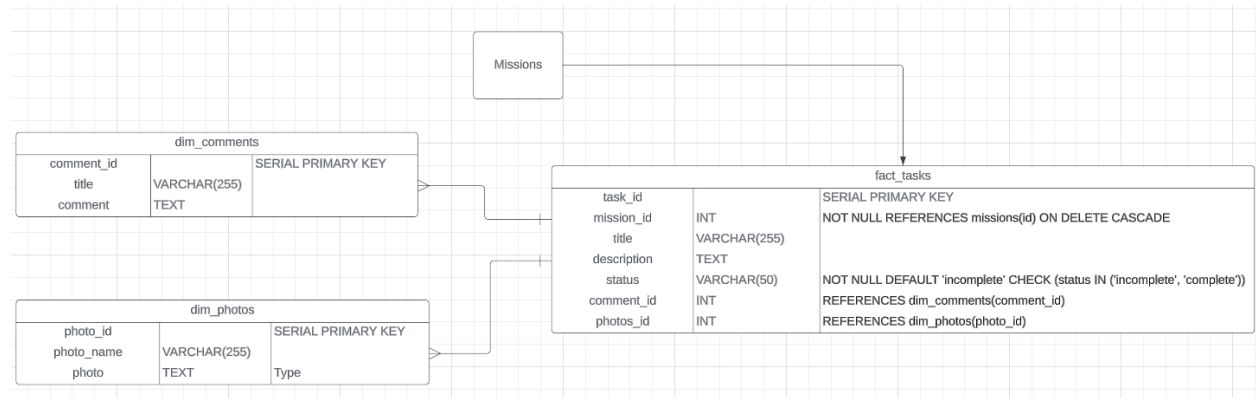
### Guides:



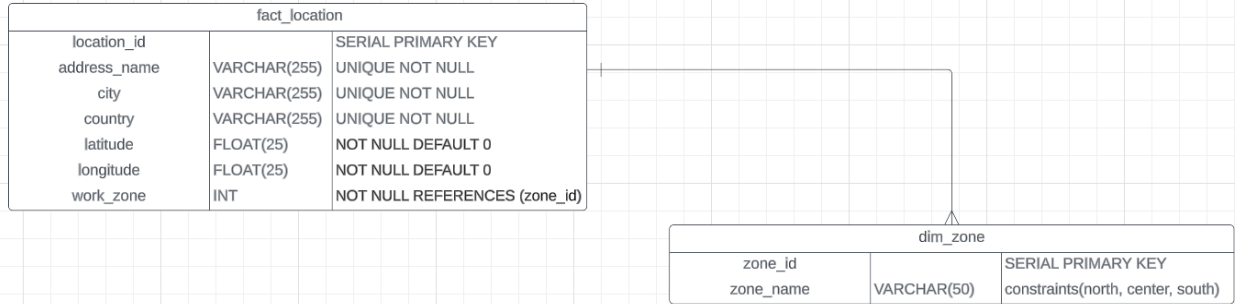
## Missions:



## Tasks:



Location:



7. Flowchart:

