**PROJECT architecture document**  
EVENT Seater pro

# Overview

## Project Background and Description

This project was initiated to facilitate the bride and groom in preparing for their wedding by organizing and arranging seating. The target clients are wedding planners. The project provides an initial solution for arranging guests at tables based on categories such as bride's side, groom's side, family, friends, and professional contacts.

## Project Scope

The scope of this project includes the development of a web application that allows the event host to enter guest lists directly or by uploading Excel, define the number of tables and specify seats at each table. The app will use an algorithm to automatically assign guests to tables based on predefined categories such as bride's side, groom's side, work, family or friends. The output will be a visual map of the venue with interactive tables showing the list of guests at each table. Out of bounds are functionality such as consideration of guest attendance confirmations (assuming the list provided to the algorithm includes only those who are certain to attend), accuracy about table placement, and considerations for table allocation based on age categories or health conditions.

## High-Level Requirements (Design)

The new system must include the following:

* A user-friendly web interface that requires no software downloads for access.
* Compatibility with various devices and browsers to ensure accessibility for all users.
* An ability to upload guest lists through both direct input and Excel files.
* An algorithm that organizes guests into tables based on predefined categories and preferences.
* A graphical representation of the venue with interactive tables for easy visualization and adjustments.

## Who are the users?

Administrator/ event host: can define the details of the event, enters the guest's details and preferences, manage guest lists and supervise the entire process of arranging the seating.

## Project entities (Architecture)

Fronted – vite, vew.js

Backend, server – Node.js

Database – Mysql workbench, migration

## Typical user flows

1. User Registration: The event host registers on the platform by providing necessary personal information such as name, email, and password.
2. User Login: After registration, the event host can log in using their credentials to access their personalized dashboard and manage their events.
3. Event Creation: The host creates a new event, entering specific details like the date, venue, number of tables, and seating capacity per table.
4. Guest List Management and Excel Integration: The host can download a blank Excel template from the website to fill out with guest information. Once completed, they can upload this file back to the platform, where the data is updated in an interactive table. Hosts can also add or modify guest details directly in the web table.
5. Seating Arrangement and Adjustments: Using the uploaded or directly entered guest data, the system generates initial seating arrangements based on predefined criteria such as relationship to the bride or groom, work, family, or friends.

## Epics and user stories and test plans scenarios

**Epic 1: User Registration and Authentication**

User Story: Register

**Description**: As a user hosting an event, I want to register for the event seating management system to utilize the seating arrangement algorithm.

**Subtasks**:

* Display a registration form with fields for name, email address, and password, along with password confirmation.
* Validate that all fields are filled correctly: ensure that the email address is not already associated with another user, the password and password confirmation match, the password meets complexity requirements, and the email address is in valid format.
* Provide a link that redirects logged-in users to the login screen.
* Display a login button for users who are not yet registered.
* After successful registration, display a success message and redirect the user to the login screen.

**Test Plan**:

* Verify that the registration form displays all required fields.
* Check validation rules for each field (email uniqueness, password complexity and matching, email format).
* Confirm redirection functionality and success message upon successful registration.

User Story: Login

**Description**: As a user hosting an event, I want to log in to the event seating management system to utilize the seating arrangement algorithm.

**Subtasks**:

* Display a login page prompting the user to enter their email address and password.
* Validate the email address and password format.
* After successful authentication, redirect the user to the system dashboard.
* Display an error message for incorrect login credentials.

**Test Plan:**

* Ensure the login page displays the correct fields.
* Confirm email and password format validations are enforced.
* Check redirection after successful login and error messaging for incorrect credentials.

**Epic 2: Guest List Management**

User Story: Import Guest List in Excel Format

**Description**: As a user hosting an event, I want to export an Excel file from the website, fill it with my guest list, and import it back into the website.

**Subtasks**:

* Display an Excel file with a table containing column headers for the information the user needs to fill in.
* Provide the ability to download the Excel file.
* Allow the user to import the filled Excel file.
* Check that the uploaded file is in Excel format and matches the sample file's columns.
* Display an error message if the uploaded file does not adhere to the expected format.

**Test Plan**:

* Test the functionality to download and upload Excel files.
* Verify file format and column matching validations.
* Ensure appropriate error messaging for format discrepancies.

User Story: Capture Guest List from Website and Add Guests

**Description**: As a user hosting an event, I want to view and add guests directly from the application interface without using Excel.

**Subtasks**:

* Display a table on the website containing the invited guests with details like name, email, and category.
* Include input fields to add a new guest with details.
* Validate the input data.
* Update the guest list table dynamically to reflect the newly added guest.

**Test Plan**:

* Ensure the guest list table and input fields are correctly displayed and functional.
* Validate input data handling and dynamic updating of the table.

**Epic 3: Automated Seating Arrangement**

User Story: Automated Seating Algorithm

**Description**: As a user hosting an event, I want the system to automatically arrange guests at tables based on predefined categories like bride side, groom side, family, and friends.

**Test Plan**:

* Test the algorithm's ability to categorize and place guests based on specified criteria.
* Verify the correct display of seating arrangements on the user interface.
* This structure provides a clear roadmap for development and testing, ensuring that all features meet user needs and functional requirements effectively.