Meeting Scheduler

Functionality:

The application is used for planning meetings of a group of people. One user joins the group in the role of organizer,

which can invite other members to the planned meeting or remove them. The organizer can also propose events,

which serve as a means of conducting the meeting. Each member can vote for exactly one event within one meeting.

Voting has a given deadline within the meeting, the organizer can manually select the event in which the meeting will take place

(so it doesn't necessarily have to be the event with the most votes).

If he does not select the event manually, after the deadline the event will be decided automatically, the event with the highest number of votes will be selected.

Structure:

Model:

Main entities:

User - a user who can create meetings (becomes an organizer within the given meeting) and join other users' meetings

Meeting - planned meeting, contains a group of users who communicate with each other. It also contains a list of events proposed by the organizer,

which users vote for.

Event - a potential way of conducting a meeting. Contains the type of event - personal or online,

in the case of a personal event, the meeting place, in the case of an online event, the meeting platform (e.g. Zoom), and the date/time of the event.

Location - physical meeting place, contains the address and description of the location.

Persistence layer:

JPA and Spring DAO are used

Service layer:

Entities are manipulated by associated services using DAOs connected using spring dependency injection.

Controller layer:

Requests are processed by spring controllers, which use services, reconnected using spring dependency injection

Installation:

The application can be run using the IDE on localhost, or by packaging it into a WAR. It uses the postgreSQL driver for the database

must have access at startup. The application can be tested using a set of HTTP queries generated in Postman, the queries are available in git

project repositories.