⟨ Awesome 2018 Project ⟩

Multi Planner

6 people

Project Description

The software should allow to organize events between friends, with many features to help the organizers. It should help to find dates suitable for everybody, provide a way to keep tracks of the money spent for the events and the reimbursements, and give an easy way to set up the sitting arrangement.

Skills

Web

Databases

(*)[The skill scale is from 0 (Fundamental Awareness) to 6 (Expert).]

Level 1 Here are the tasks for the first development phase.

⋆⋆⋆ Basic features

Functions to create a new database; create users. All features (above and below) should be usable through a web interface. The web service should require to log in.

⋆ Events

A user can create an event, with a starting and an ending date, and may invite people to an event. Invited people may accept or decline the invitation.

⋆ Agenda

Each user can display its agenda, made of all its events and the ones he was invited to.

⋆ Notifications

The interface should feature notifications reminding users of nearing events. Reminders should be configurable: enabled/disable, delay.

⋆ ICS export

Calendars must be accessible as ICS, with authentication.

⋆ Accounting

The organizer can link an accounting to an event, which allows to register all the money transactions made for the event, i.e. who paid what for the event. The cost of the event is split evenly for all participants.

⋆ Balance

It should be possible to compute the current balances of an event accounting.

⋆⋆ Resolution

The tool should propose solutions to balance the account: optimizing either the total number of transactions, or the maximum number of transaction per user.

⋆ Relationships

A user can declare publicly that he is *friend with* somebody. More specifically, relationship can be secretly rated, from -10 to 10.

⋆⋆ Sitting arrangement

For an event, the organizer can provide a number of tables with their respective capacities. The application should propose a sitting arrangement for all the invitees, based on the secret relationship ratings, and a notion of optimality that should be specified clearly. The organizer may provide its own relationship rating between people if he wishes.

⋆⋆ Optimal Meetings

Based on a set of constraints and the agendas of the people invited, set up multiple meetings at once. Constraints include minimum/maximum delays betweens events, duration of events, and possible time ranges of an event. For instance, knowing that Paul and Anna wish to meet three times over the next four weeks, during afternoons, with four to ten days between meetings; that Paul and Marc wish to meet between 5 pm and 7pm on Tuesday or Friday; that Marc and Anna wish to meet in the morning any day but the week-end; given their respective agendas, provide suitable time slots that the suits all the constraints.

⟨ Awesome 2018 Project ⟩

Multi Planner

6 people

Project Description

The software should allow to organize events between friends, with many features to help the organizers. It should help to find dates suitable for everybody, provide a way to keep tracks of the money spent for the events and the reimbursements, and give an easy way to set up the sitting arrangement.

Skills

Web

Databases

(*)[The skill scale is from 0 (Fundamental Awareness) to 6 (Expert).]

Level 2

Level 1 must be unlocked to read this section

⋆ Developper Documentation — Required for lvl 2 validation

Document your project (not necessarily only in the source code) so that a newcoming developper could understand and contribute to the code.

⋆ Release — Required for lvl 2 validation

Produce a release as a source archive or git tag. The release files should have up-to-date README and INSTALL files and more generally allow anyone to deploy the application.

⋆⋆ Multiple currencies

Transactions, balance, etc. should be expressible with multiple currencies. Up-to-date exchanges rates should be obtained online.

⋆⋆ Flexible transaction split

When creating a transaction it should be possible to specify the amount that goes to each beneficiary. By default, the total amount is distributed evenly among beneficiaries, which should yield the same behaviour as in phase 1.

⋆⋆ Permissions

It should be possible to declare some users as application administrators. Additionally, each event and group must have a list of specific administrators. By default, the creator of a group or event is its only specific administrator. For a given group or event, only application admins or specific admins may change the list of specific admins. Similarly, only these users may add, invite or remove people from the event or group. For transparency, when viewing an event or group, the application admins and specific admins should be displayed.

⋆ Resolution validation

Resolution may be computed only by an admin of the event or group. It must be possible to apply it, adding transactions to the event or group that are marked as pending. These transactions must then be validated, upon receipt of the associated payment, by their recipient. Pending transactions are ignored for the computation of the balance and for new resolutions. The application of new resolutions should delete previously pending transactions.

⋆⋆ Integration Test

Continuous integration should feature at least three different tests that access pages from an application running on the CI server. One of the tests should make sure that it is possible to create an event, then edit it, and make sure that this operation does not duplicate the event.

⋆ Performances

The testing suite ran on the CI server should feature at least one performance test. This test should create a large number of transactions in some group, and then trigger the computation of the balance of some user, and check that it is performed quickly (typically less than a second). This computation time could be used as a metric to justify algorithmic improvements.