

Helper App

System Specification

Europe Center Helpers Center & course scheduler

Git

<https://gitlab.dwbn.org/dwbn-it/tools/ec-jobcenter-project>

Data set/get

Fields (for example):

:: status = is bound to datatable field "status" (gets data from there)

->status = writes to datatable field "status"

Discussion

Opinions and decision needed

A. Executive Summary

The App helps to organize and handle open jobs on (the EC)-courses, but can be migrated for other courses, too. Additionally it provides the latest course time schedule and informs about the changings. This is done via an up-to-date time schedule and help requests plan and an „alert“-functions for urgent needed jobs and schedule changings.

Friends are able to pick a help request directly over the App. They can login via their DW-Connect account or their course registration data.

The App can be downloaded from the SC website. The registration form would include it, too.

It is possible to have the App functioning locally without Internet (Fe. by setting up the Admin-software on a local Server which is reachable by local network only)

Version 1 will be "Intranet only" because Internet is sometimes very weak in EC and we need no realnames and e-Mail registration from friends for the app, too.

The Helpers Center functions like this: we are like a hub for all SC teams. Precise team representatives send us the info how many helpers they need, at what time till when and what exactly helpers need to do. This info may be updated couple of time during the day. So we are collecting the info, put on the wall that everyone can see and run around to ask people to help. There will be two admins to run the app from the Helpers Center. One will be changed every second/third day to let all my team members get used to run it. And two people from the Communication/Program team, I believe)

B. App Description

1. Admin Backend (BE):

1.1 Manage Jobs

The admin is able to post and to see different jobs and job categories for different course days and can fill them with the amount of people needed. (Fe. Category „funky dishes“, friends needed: 50, day: 15. 09.2019, time from-to)

After posting the jobs they are listed in the App. If jobs are not taken about 6 hours before they should happen, the app alerts according to the app settings (see below – 1.3. Settings).

Additionally the admin chooses by setting up a new job if the job is alerted in the app.(fe because the job is new and urgent -> check a checkbox)

Of course the Admin BE is allowed to update and delete categories and jobs, date and amount of friends needed. He has an overview of the jobs like in the app and can update if fe more friends are needed.

We split up jobcategories- and jobadding- functions into different admin views and not give this possibility in the main “job”-view for admins(see below).

So that job editors find only *select* fields for jobname and jobcategories for daily work and must not name or rename concously defined jobtitles by a main admin. This should be done only once by a qualified admin, then jobs and categories are selectable in the “daily work job view”.

Additionally we split the functions for admins by using different views for helper center admins and event admins. (-> see view “course schedule”)

[View “login” \(admin\)](#)

No DWBN SSO login necessary (as described above).

We also do not need any register form for admins because those few admins (about 6 to 10 people we’ll set directly in the database over internal standard views (phpadmin)

As Markus mentioned below it is not good to do changings directly in phpmyadmin because that user has full rights on everything so a small view for adding and changing admins and admin params would be nice.

So this is a simple login form with maybe password resent function. No personal data needed.

Username (nickname or name/surname fields available)

Password

Olena: Just a question: can we use NO personal data? If the only things we need are to inform people how many potatoes need to be peeled and that next session will start one hour later, maybe we don't need to go to a personal level at all... When we send to the app users alerts, do we have to collect their cell numbers etc?

Can be much easier to make it less personal.

As agreed we will do no personal data. The communication for the PUSH alerts will function over the App-ID which will be set by downloading the App (as proposed in Skype discussion)

Stefan: Ive left name and surname in the "users-table" maybe for later use and added field "pass" for admin pw.

Ive added "appid" (set by downloading app) to the users-table for Application ID and communication between Server and Client.

This login view leads to the job overview (View "job".)

Changes meant as "jobs updates" are done a few times a day.

View "jobs"

admin-view for deleting, updating and creating jobs

The view for the admin is desktop optimized and workable from smartphone

Note for designer

Adding, editing and deleting jobs **in one view** is possible by programmers.

Needed data/formfields

- jobname (*selectfield*) :: jobs.jobname (*table.field in mysql-database*)
- jobcategory (*select*) :: jobcategories.category
- friends needed (select) :: jobs.persons
- from (text) :: jobs.timefrom (datetime)
- to (text) :: jobs.timetill (datetime) Always precise time needed for helpers fe from 11:00 till 13:00
- alertbefore (**overwrites standard alert setting - see View "settings"**)
- maybe checkbox (marker for updating/deleting job and then confirm all tasks via button)
- completed (when we got enough helpers - Stefan this may be counted by the app automatically)
- two type of alerts: news (new jobs/jobs updates and new program/program updates) and reminders about taken jobs (Olena suggested star system)

- active/passive -> jobs.active

URGENT PUSH message

- urgent (independent from *alertbefore* because job is really urgent) -> jobs.urgentmessage

Note: We solve that via an App-Push function for urgent jobs or event changings.

- button "publish job" (admin) or "select job" for the user frontend

- yes urgent jobs to be done needed (or "priority job" category/marker in case of urgent or when few type of jobs asked for the same time period)

Discussion

Maybe there is a color system which highlights urgent jobs.

Jobs are sorted by date and persons needed. Other sorting possibilities needed?

Olena prefers a "stars" system.

Additional elements on "jobs" admin view (if splitting is done as proposed):

- link to view "add jobcategory"
- link to view "add job" function
- link settings (button) (links to **View** "settings")

Olena: priority job marker can be a star chosen from a checkbox fe...

Discussion

Designer approach where general app settings is reachable from, maybe on every admin view as a gear wheel or something

Different views:

Communicators (Admins)

Helpers

On start screen: latest reviews/inspiration for other helpers. Maybe in version 1.1.

View "jobcategory"

admin - add/delete/update

View for admins to edit jobcategories

data/formfields

- jobcategory (admin: text, listfield combination for updating) -> jobcategories.category
- comment (admin: text, explanation of category) -> jobcategories.comment
- active/passive (general setting) -> jobs.active

- checkbox (marked for updating/deleting jobcategories)

View “jobnames” (admin)

View for admins to add/delete/update jobnames

data/formfields

- jobcategory (admin: listfield) :: jobcategories.category
- job (admin: textfield) -> jobs.jobname
- comment (admin: text, explanation of category) -> jobs.comment
- active/passive (general setting) -> jobs.active
- checkbox (marked for updating/deleting jobnames)

COURSE SCHEDULE (admin)

1.2 Manage Agenda and Schedule (admin)

The user frontend includes the actual course agenda which is of course the main reason why people download the App (the carrot). If there are any changes the APP informs actively by alert. The schedule data can be inserted and updated via the Admin Backend.

View “course schedule”

Admin - add/update/delete a course schedule

Event (text) → course_schedule.event

Eventfrom (datepicker)→ course_schedule.eventfrom

Eventto (datepicker)→ course_schedule.eventeventto

Comment (info on event) → course_schedule.comment

Changed → course_schedule.changed; if changed PUSH message and changed in course schedule, too

Changed Message -> course_schedule.changedmessage (PUSH message)

active/passive → course_schedule.comment (prepared/set online)

Info

Choosable schedule days from/to is restricted by datefrom/to in course settings

Datatable: I left it simple (see datatables @ the end of this doc), maybe we need some grouping possibility. Right now the grouping is simply after date which can be generated automatically by programm language.

(mg) >> I think it should be possible to group by course name also and sort by date or urgency below this grouping. (sts) Ive added courseid and an “urgent”-field (star system as proposed from Olena) to the datatable “course-schedule” and a filed “changedmessage” for the PUSH message for changes. Ive left name and surname in “users” even it is not needed for now.

GENERAL APP SETTINGS (admin)

2. Settings

The settings are used to set different main app params which could be:

- Set a new course, fe. „Summer course EC 2019“
- Archiving/backup/Cleanup: Save whole course for later documentation and analysis

Discussion Backup/restore function

Not needed in App, is done by server environment.

- Job alert function: Set different levels: Alert time before jobs get really urgent (fe 24 hours). PUSH and seen on the overview

Discussion

Settings not set on homescreen. This is a designer decision where to place. I find job view most important for daily work so the **view “jobs”** on first place for me.

In Detail now:

View “settings”

Generate a course (fe. EC summercourse) where jobs & categories are attached to

coursename (text) -> settings.coursename

status (list; possible settings active, passive, *backup*, *archived*) -> settings.status

startdate (datepicker) -> settings.startdate

enddate (datepicker) -> settings.enddate

alert (standart alert when job gets urgend; 0 = no standart alert) -> settings.alert

Discussion

Maybe the status appears as an “icon” (green, red,..) on every admin (backend) view.

Olena proposed a star system

Link “View/unblock adresses”

Discussion

View blocked addresses

Possibility to reactivate disturbing mac_adresses (for some reason blocked devices)

Must not be a view can also be over database standard view only (phpmyadmin).

Discussion brought that it is maybe no bad idea to have some control over the system.

Markus: I think that changes done by phpmyadmin are no good idea, because phpmyadmin is able to change everything in the database. Also delete everything. A small Admin-View will make the job in a more secure way. Stefan ok I agree than we should think about an admin view for setting an admin and changing/adding pw, too, eventually (see above)

USER Frontend (app for friends)

Mobile first approach

2. User Frontend (FE):

View “home” (frontend)

User view, first frontend view in App.

This should be exiting so maybe this could be

- showing some **news**, showing latest changes, showing urgent jobs

Discussion

If more news then only course schedule changings are available this would be a nice feature. Maybe this app is used to announce next travelteaches roundtable in the EC tent or other exiting stuff around EC. If needed we have to create datatable for this.

Discussion brought that that the App should be kepted plain and simple.

The home view should be a brief and interesting summary of what is going on. (Important changings on schedule, really needed jobs, something else ..)

This views should also have a logon-button, where you can logon for registering to jobs via DWBN SSO or if not available or possible (Intranet solution only) by registering to the App.

Markus: Let us talk about a dashboard for the user as a first view in the app. Different Buttons, like 'login/logout' or so should appear in a top-navigation fixed at the top of every single view of the app. Or should we take these buttons in a side-menu ?

View “course schedule” (frontend)

View course schedule”

Event (text) :: course_schedule.event

Eventfrom (datepicker) :: course_schedule.eventfrom

Eventto (datepicker) :: course_schedule.eventeventto

Comment (info on event) :: course_schedule.comment

Changed :: course_schedule.changed; if changed highlighted in user frontend and also screened on mobile messenger alert, **star system?**

Changed Message :: course_schedule.changedmessage (Info about the topic in push and this view)

View “jobs” (frontend)

View “jobs”

mobile first approach because mostly used on cellphones

Simply registering by buttonclick for jobs and a nickname for inspiring others (Maybe on dashbord)

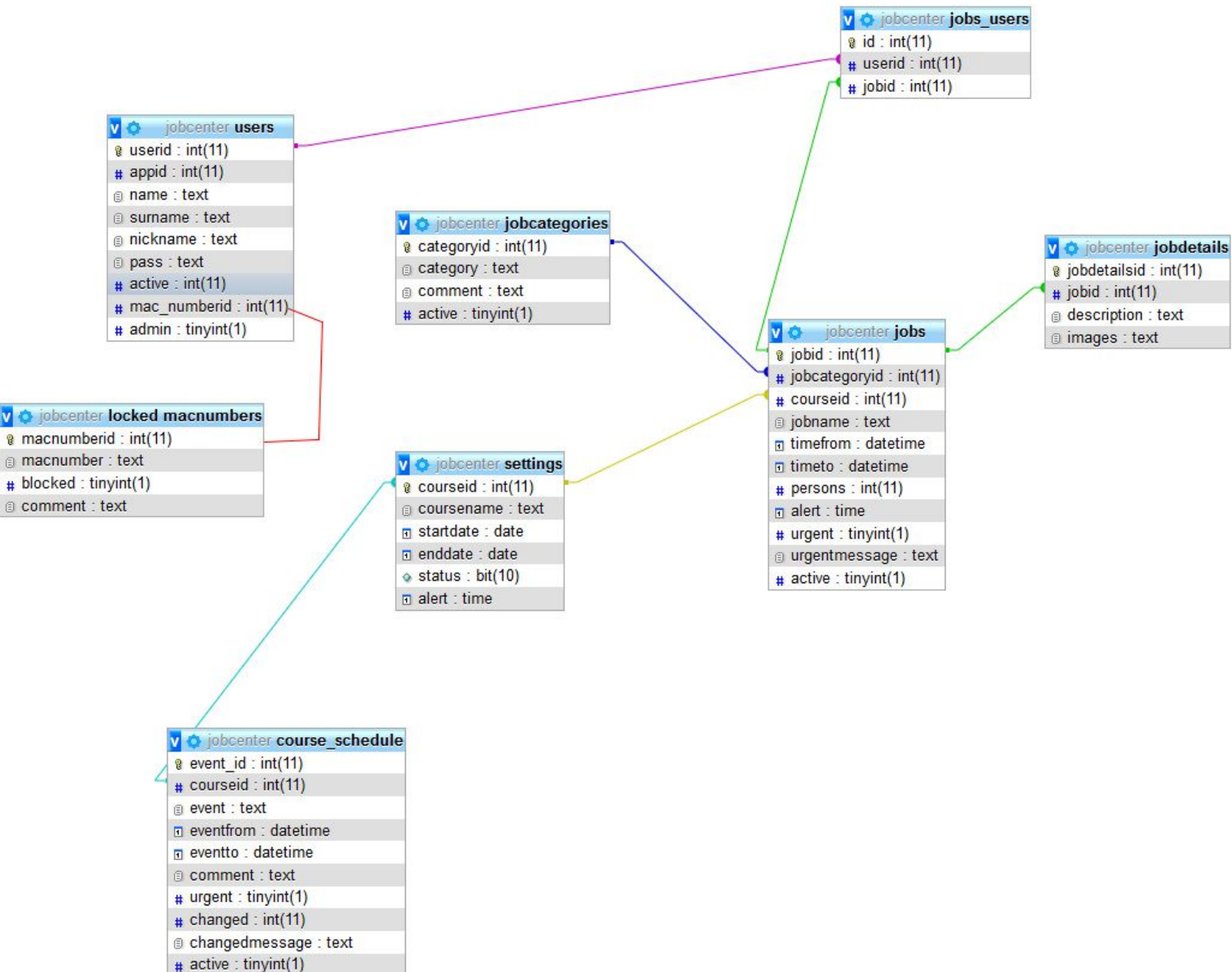
View “register for jobs” (frontend) data/formfields

- jobname (text) :: jobs.jobname
- jobcategory (text) :: jobcategories.category
- friends needed (text) :: jobs.persons
- from (text) :: jobs.timefrom (datetime)
- to (text) :: jobs.timefrom (dateto)
- how urgent (star system)
- comments, photo, some description on that job (on a detail view?) Stefan: Ive added a “jobdetail”-table to the database for descriptions, images ..
- nickname (text) -> users.nickname

Button :)

Actual "jobcenter" Database-structure, mysql, v.04.

Unbenannt *



C. Team

Olena, Management (kornyeyeva.olen@gmail.com)

Carlos; Interface Design (carlos.cubillos@gmail.com)

Stefan Spindler, Management, Programming (sales@turn360.com)

Markus Gasteiger; Programming (magast@web2you.at)

Jeremiah Hendren, Advisor (jeremiah.hendren@gmail.com)

Andreas (schoenefeldt.andreas@gmail.com, Advisory on DW Connect and on Programming)

D. Time Schedule

Concept -> End of Oktober 2018

System specification -> End of November 2018

Interface Design -> January 2019

Interface Programming, Backend Programming, API Programming

Testing

Roll Out, Marketing