

Calendar Subscription Program

Overview:

The goal of the Project is, to create an program that offers a Calendar Subscription via ics/webdav format. Therefore the program offers the ability to create and administer multiple events in multiple calendars and administer the subscriber of these calendars. The program have to offer a subscription option for google calendar, android calendar, outlook, outlook 365 calendar, ios calendar and others. The whole system have to be designed for multilingual use and also be optimized for mobile phone use. Along with the Laravel/HTML/react application there have to be a mobile app, that can distributed via the known app-stores.

Technology that have to be used:

- Laravel (PHP)
- Laravel migrations for database
- react
- react nativ (for mobile app)
- database have to be encrypted
- HTML5 / CSS
- javascript / jquery / Bootstrap
- Rest API

Programs / Platforms that have to be used:

Process/Bugtracking : [freelancer.com](https://www.freelancer.com) (via docx files)

Version Tracking : gitlab.com

General:

Up to 20% changes to the functions or the design can occur during the implementation of the project.

Detailed Description:

Step 1: Basic functions

1.1 Company

1.1.1 individual login

1.1.2 address

1.1.3 settings

1.1.4 registration wizard

1.2 Calendar

1.2.1 Calendar view

1.2.2 Calendar listview

1.2.3 create calendar

1.2.4 create event

1.3 interfaces

1.3.1 create subscription link

1.3.2 calendar subscription

1.3.3 analyses

1.4 Public view

1.5 Design

1.6 super admin area

1.6.1 Overview of companys

1.6.2 Register new Company

1.6.3 ability to deactivate company login

1.6.4 eMail log

Step 2: advanced features

2.1 advanced analyses

2.1.1 geolocation / worldmap

2.1.2 mask subscribers per timeperiod/calendar

2.1.3 Dashboard

2.1.4 link shortener / link tracking

2.1.5 calendar cardview

2.1.6 event cardview

2.2 HTML subscription code generator

2.3 advanced super admin area

2.3.1 license models

2.4 Subscriber manager

2.4.1 change the subscribed calendar

2.4.2 add a description

2.4.3 see statistics per subscriber

Step 3: mobile app

3.1 login

- 3.2 main screen / dashboard
- 3.3 menu structure
- 3.4 calendar views
 - 3.4.1 calendar
 - 3.4.2 calendar cardview
 - 3.4.3 event cardview
- 3.5 analyses
- 3.6 calendar share functions

- 4. miscellaneous
 - 4.1 settings
 - 4.2 sending eMails
 - 4.3 pb-login
 - 4.4 Maintenance mode
 - 4.5 Database
 - 4.6 saving pictures
 - 4.7 technical documentation

- 5. Milestones

Step 1: Basic functions

1.1 Company

All options and information are related to an individual company with a unique company id.

1.1.1 individual login

Each Company has its own individual login. An eMail Address has to be used as user name. Therefore a login mask has to be implemented.

1.1.2 address

The Company has to have a Mask with company information to be edited. Which contains the following fields, the fields marked with "*" are mandatory fields:

- Name of the Company*
- Street name and number*
- Postalcode*
- City*
- Country*
- telephone number*
- email address (could be different to the login eMail address)
- website url
- Tax id (text field)
- Bank account information (IBAN, BIC)

- username (when changed with confirmation eMail)
- password (with confirmation)

1.1.3 settings

2nd Mask for the company is the settings mask, with the following fields:

- select the calendars to be private by default
- select the calendars to ask for the eMail as default
- default calendar in the view and for the events

1.1.4 registration wizard

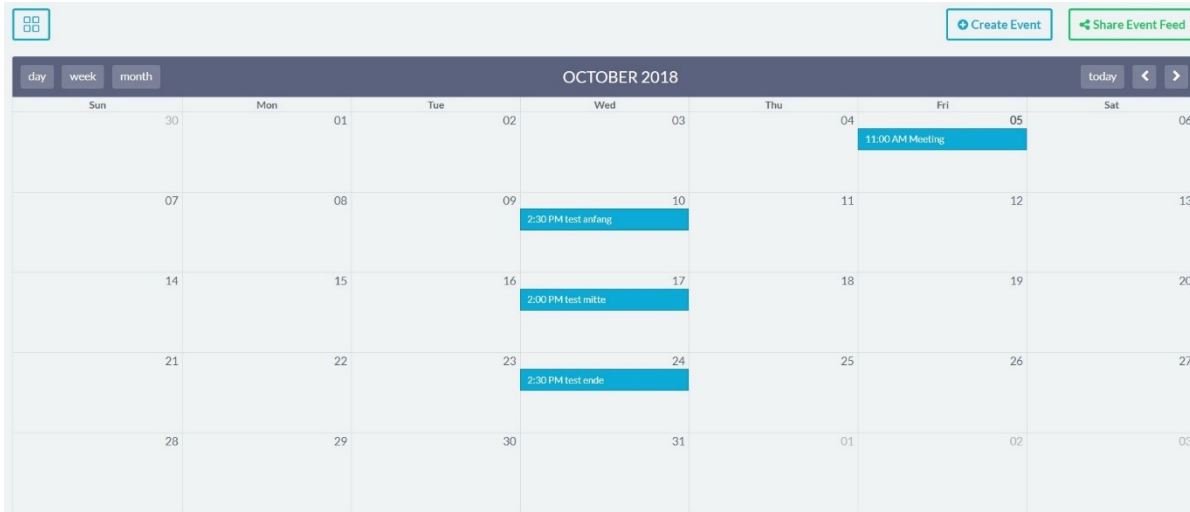
There has to be a wizard for the registration of a new company. With few steps, to get the needed information.

1. Login and Password (with password confirmation and check if eMail is not used before)
2. Address (see mandatory fields from "1.1.2")
3. choose price model and sepa (fields for IBAN and BIC and a checkbox for sepa, a text, all are mandatory)
4. Registration confirmation (Text: "thanks for registering a new account" and a note to inform the user to confirm his eMail address)
5. (no extra site) An eMail has to be sent with a confirmation link inside.
6. (if the link is clicked) Get to the login page with a hint that the eMail is confirmed.

1.2 Calendar

Creating an overview of the calendars.

1.2.1 Calendar view



Picture 1 – Calendar view

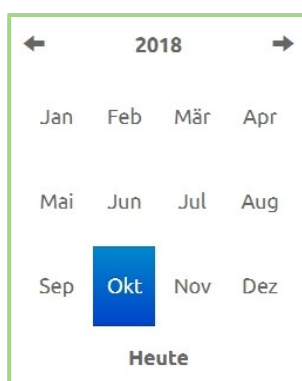
Users need a calendar to have an overview of upcoming events.

It's a simple calendar with all events linked to it as shown in *Picture 1*. The user can switch between different views (day, week or month).

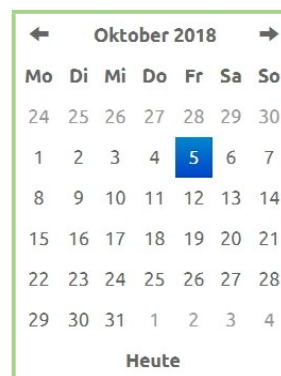
The calendar should contain a left and right button to quickly jump **one** day, week or month (depending on the current view) forward/backward. A "today"-button jumps back to the current day, week or month.

There should be an icon/a button behind to the shown date (*Picture 1* – "October 2018") to jump quickly to a specific date. When clicking on it, a small window with another calendar will be displayed to search for that specific date. It starts with the selection of a month of the current year (*Picture 2*); the year can be changed with the arrows to the left and right.

To finish the selection, a specific day has to be selected (*Picture 3*).



Picture 2 – Month selection



Picture 3 – Day selection

The Calendar view (*Picture 1*) should jump to that date and will show the day, week or month (depending on the selected view before).

A “Create event” and “Share calendar” button like in *Picture 1* is necessary to add more events (see create event) and to share the current calendar (see share feature).

Following points are not included or not visible in the pictures above:

- The user is able to create multiple calendars which include events for subscribers of this specific calendar. Therefore, it is necessary to display multiple calendars in one view. This should be realized by setting calendars active or inactive for the calendar view.
 - ⇒ The menu button top left (*Picture 1*) opens a dropdown menu which shows all available calendars with their specific calendar-colors which are set in the calendar options.
 - ⇒ The shown calendars can be activated by clicking the box next to it or the calendar name.

1.2.2 Calendar listview

A list with the calendar information, with ability to create a new or edit the calendar. See 1.2.3 for details.

1.2.3 create calendar

There have to be a mask to create calendars for the company. Each Calendar has it's own individual and unic id. The following Information have to be collected:

- name of the calendar
- switch/checkbox whether the calendar is private or not
- switch/checkbox whether the eMail address of the user is necessary, see 1.3.1 for more detail.
- switch/checkbox whether the calendar is offline or not
- picture of the calendar

Each Calendar has it's own unic url. See 1.3.2 for more detail.

1.2.4 create event

There have to be a mask to create events for the company. Each event is related to one specific calendar and has a unic id. The following information have to be collected:

- summary (name of the event)
- date and time pick for the start of the event
- some ability to choose how long the event is or to choose another date and time for the end of the event
- location of the event
- description
- url
- ability to choose an alarm for the event
- related calendar

The fields for the event are orientated on the ICS / VCARD standard and should be saved to be conform with that standard. See 1.3.2 for more information.

There have to be a button to copy the event to a other calendar.

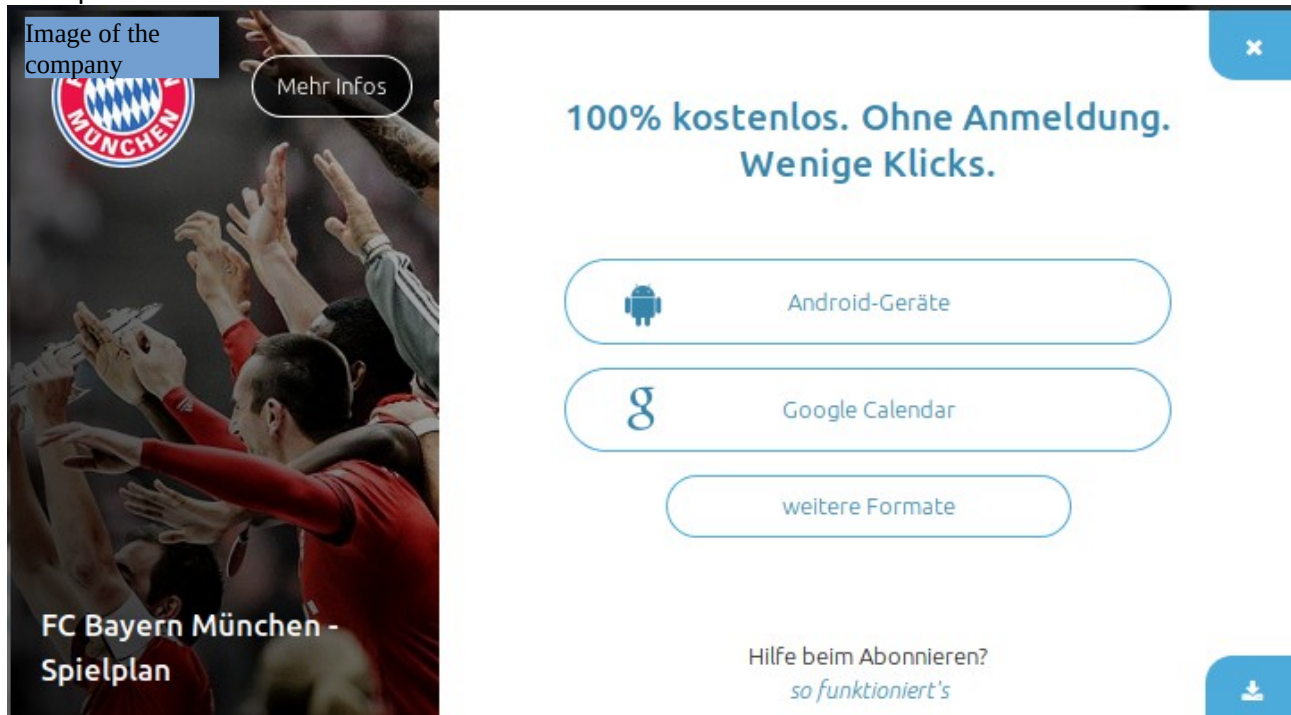
1.3 interfaces

1.3.1 create subscription link

Each Calendar has it's own link. If you hit the link, you get to a page in which you can subscribe to that calendar.

For the subscription a modal or site have to be shown, with the following content:

Example:



A button for each calendar type, for example google calendar, ios calendar, android calendar.

In the button, the protocol, that's needed is defined, for example ios calendar:

webdav://subscription.link

There have to be a individual link created per user, that are related to the anonymous user and the calendar and linked to the php that creates the ics file. Also there have to be an option to ask the user for his eMail-address, before subscribing, if the eMail field is necessary depends on the option in the calendar.

1.3.2 calendar subscription

If a user subscribe to a calendar, a entry in the subscriber table has to be created.

If the subscription link is requested. The PHP should choose the calendar to load via the subscriber table. Once the appointments are loaded, a ics file have to be generated. The ics file have to contain all appointments for the actual month and 2 month in the future.

1.3.3 analyses

Every call of a calendar via the individual link should be saved with the browsers "user agent" that made the call. Along with the subscriber that the link is related to and the ip of the request.

1.4 Public view

When the main domain is browsed, than a list of all not privet calendars would be shown.


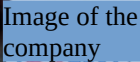
The view should be in Cards Design. With the calendar name and picture. If you click on an calendar a modal pops over, that have the ability to subscribe to a calendar.

For more details to the modal see 1.3.1.

In the top right have to be a button for "login", this refers to the login page.

In each calendar card have to be an option for “see information” if this is clicked details of the calendar have to be displayed:

Example:



COMPANY + LINK TO COMPANYSELECTED VIEW

KALENDER

EVENTS
32

subscription Modal


Title of Clandar and Link to subscribe modal ✓

Name of the Company
fcbayern.com

Beigetreten am 11.08.2015

Anbieter-Impressum

Abonniere den Kalender Spielplan kostenlos und verpasse nie wieder Dein Lieblings-Event!



Name of the Calendar

Description of the Calendar


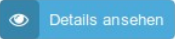





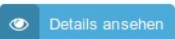
Das Abonnieren ist für Dich völlig kostenlos und funktioniert mit allen gängigen Kalendern. Klicke zum Abonnieren von "FC Bayern München" einfach auf den Button "Kalender abonnieren".

Sei Deinem FCB so nah wie nie zuvor!

Subscription Modal

Back to Company selected Calendar view

count of eventsin the calendar calendarname

	20	Okt Sa 2018 15:30	VfL Wolfsburg - FC Bayern München Beschreibung einblenden	
	23	Okt Di 2018 18:55	AEK Athen - FC Bayern München Beschreibung einblenden	
	27	Okt Sa 2018 15:30	1. FSV Mainz 05 - FC Bayern München Beschreibung einblenden	
	30	Okt Di 2018 20:45	SV Rödinghausen e.V. - FC Bayern München Beschreibung einblenden	

The top picture in the page should be the calendar image. And on the left there should be the logo of the company. The QR-Code also refers to the subscription link of the calendar. On the bottom right, the events should be displayed. The blue button for each event refers to a detail view of the event. Example:

VfB Stuttgart - Borussia Dortmund | 8. Spieltag

Mercedes-Benz Arena, Mercedesstraße, Stuttgart

Saturday, 20. Oct 2018

09:30 to 11:30
15:30 to 17:30 (CEST)

09:00

10:00

11:00

12:00

VfB Stuttgart - Borussia Dortmund | 8. Spieltag

15:30 (CEST)

Mercedes-Benz Arena, Mercedesstraße, Stuttgart

Calendar

VfB Spielplan

Alert

None >

URL

[Link](#)

Notes

Notiz

8. Spieltag. Fix terminiert.

Ausverkauft

Tickets gibt es hier: [http://ausverkauft](#)

1.5 Design

Overall Design Template will be provided when we award you to the project.

1.6 Super Admin area

There have to be a special login for super admin with the following functions.

1.6.1 Overview of companys

A site with a list of all registered companys. With an option to change the company parameters see 1.1.2 for the fields. In addition to these fields there have to be a field for "salesforce number".

1.6.2 Register new Company

A mask to create a new company, manually without the registration wizard or eMail sending.

1.6.3 ability to deaktivate company login

In the mask that are described in 3.1 and 3.2 there have to be a checkbox to deactivate the company login. Also we need an option to do this automatically within a date and time.

1.6.4 eMail log

Once the eMail is send successfully or not, the "mail-daemon" calls a callback url and sends back the status. We need a table to see this messages. The messages comes also via http-request in

application/json format, a interface/api have to be implemented for that. The traffic is encrypted via openssl. The detailed description will be provided when we award this project to you, the description comes with sample php.

Step 2

2.1.1 geolocation / worldmap

If the calendar is called by the user, the ip and the geolocation are to be determined.

- Function: Display a worldmap, showing the calendar subscriptions grouped by there location (get the geolocation of the ip from requesting devices), implemet the function to sort by time period, see Analysis
- Design: Display a Worldmap, make color gradations for different continents Continents to Display: - Europe - North America - South America - Asia - Afrika - Australia
- Example:



2.1.2 mask subscriber per timeperiod / calendar

Function: Display Subscriptions in selected period of time (one week, one month, one years)

Design: 3 Tabs(one for each period, see above), use different colors for the bars(blue bars for weekly, green for monthly, orange for yearly)

- | | | |
|--------|---|----------------------------|
| Week: | - | one bar per weekday, e.g 7 |
| Month: | - | one bar per Day, e.g. 30 |
| Year: | - | one bar per Month, e.g. 12 |

Example:



2.1.3 Dashboard

Here there should be short links to “create calendar” and “create event” with a button and symbol. In addition there should be several cards with statistic information. For example: average subscriber per calendar this week, a list of the upcoming events, subscriber this day

2.1.4 link shortener / link tracking

We need a system to generate short links. If one of the links is clicked, a log entry should be set. This function should be available in the creating process of the Event. So you can track how often the url is clicked per event.

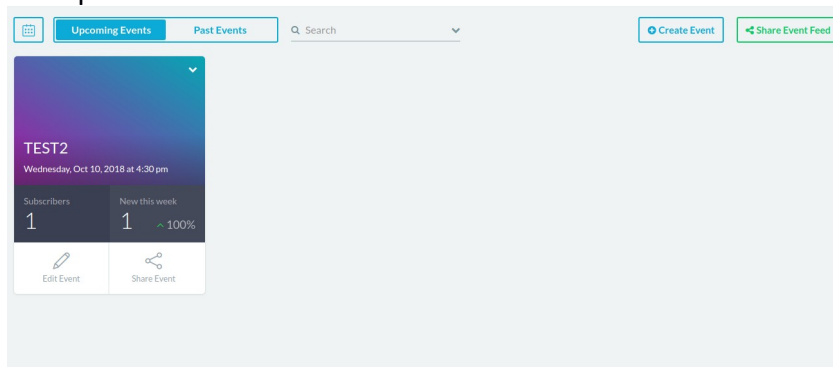
2.1.5 calendar cardview

There have to be an option to see all calendars in one list. This list have to have an cards design. The Calendars image has to be displayed at the top of each card, underneath the image have to be more information about the calendar, such as name, subscribers, new subscribers this week. And the ability to search through the calendars or sort them by name, creation date, subscribers. In the Calendar card should be a button to get into the event cardview for this calendar. And a button for sharing.

2.1.6 event cardview

A view in which all events are displayed in there chronological order. In the page the user can switch between upcoming and past events.

Example:



There should be a button to edit the event. And a button to share the calendar of the event. In the card of the event, the name and name of the calendar should be displayed. Also the subscriber of the calendar and the subscriber that are new in that week have to be displayed in the card.

In the Top of the view there should be a search field to filter the events.

2.2 HTML subscription code generator

Design: Modal-based

Function: Generate HTML Code for Buttons(to use in Emails or on Websites) to add the Calendar Subscription into the personal Calendar of a Customer.

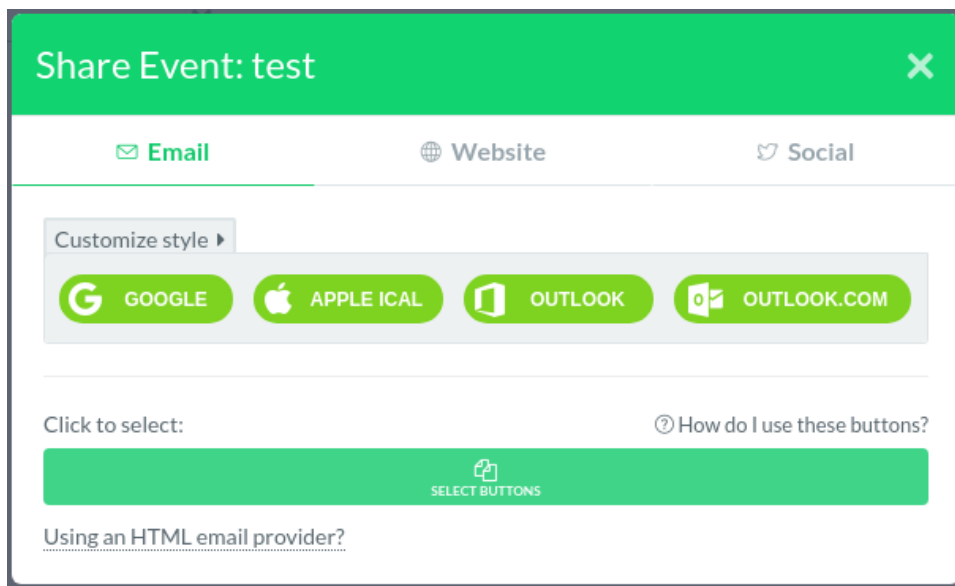
Multi-Tab-View:

- ⇒ Buttons for Email
- ⇒ Buttons for Website
- ⇒ Social Features

Buttons for Email

- multiple Buttons "Add to Calendar" for different Calendar-Provider(Apple, Google, Outlook, Caldav)
- Possibility to change style,color and icon of the buttons
- Generate HTML-Code + Button "Copy to clipboard"
- HTML Code have to be ready to insert it into several eMail providers HTML-eMails
for example: outlook, google, mailchimp, leadpages

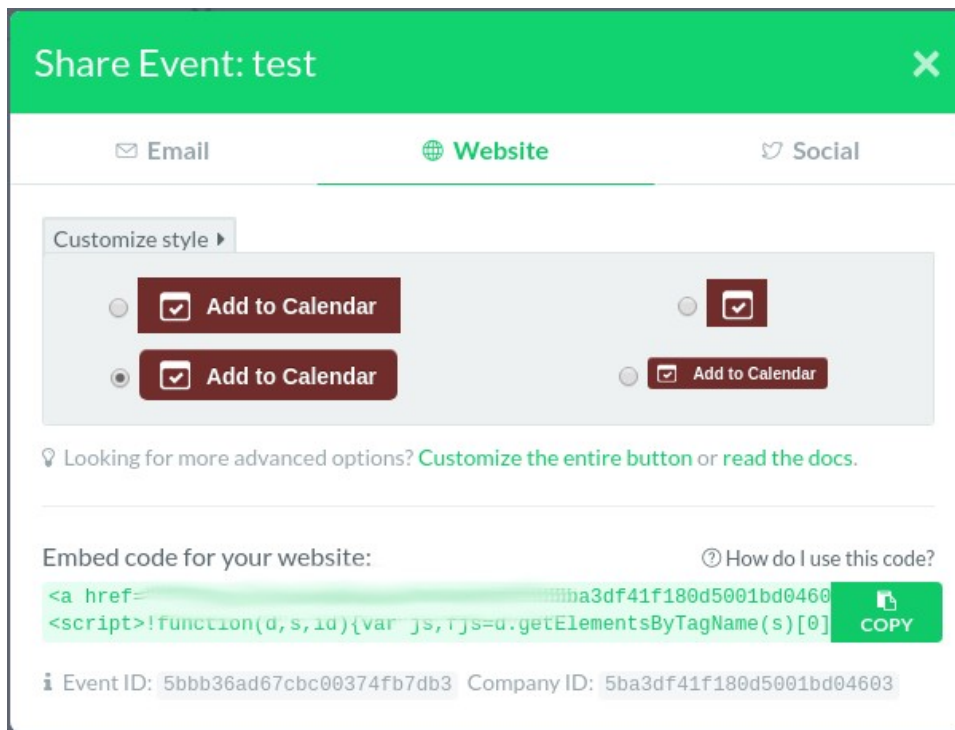
Example:



Buttons for Website

- Generate Buttons for HTML-Sites with Style-options: Size, Color, Icon
- Generate HTML-Code + Button "Copy to clipboard"

Example:



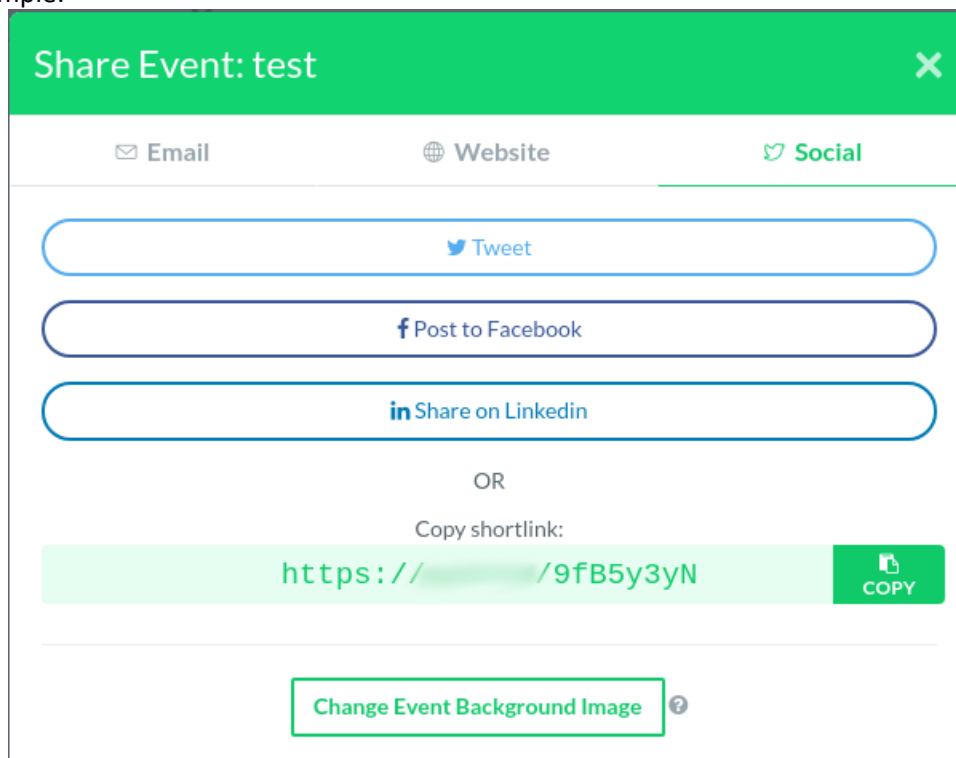
Social Features

Buttons to share Calendar on :

- Twitter
- Facebook
- LinkedIn

Short link to Calendar + Button "Copy to clipboard"

Example:



2.3 advanced super admin area

2.3.1 license models

Several license models can be defined and added to the users account. Each model has it's own restrictions. For example:

- number of calendars that the user can create
- number of events per calendar that the user can create
- number of subscribers per calendar
- function (for example if the HTML Editor is enabled)

2.4 Subscriber manager

The subscriber manager offers the ability to change the calendar for each subscriber or to unlock him. Basically it shows a list of all subscribers with few abilities and options for each subscriber. In the main page (subscriber list) the user can choose of several filters and orders, for example (last time update, subscription date, name, calendar)

2.4.1 change the subscribed calendar

If one of the subscribers is than clicked, in the corresponding page the user can change the calendar that the subscriber get, when he requested over the link.

2.4.2 change information

Also in this page, the user is able to change the eMail Address, the name and can add an description for the subscriber.

2.4.3 see statistics per subscriber

The user can see on which date/time the subscriber had requested the calendar updates.

Step 3: mobile app

The goal of this point is, to generate a react-native based app. That can be listed in the app stores.

3.1 login

When the app is started, first time, a login screen have to be displayed. With a link to register a new company, if the link is clicked you get to the registration wizard. The app have to have the ability to save that login information and login the user automatically on next startup of the app.

3.2 main screen / dashboard

If the user successfully logged in, the calendar cardview is displayed. In the top left corner is a hamburger button to display the menu. The menu swipes in from the left.

3.3 menu structure

- Home → Links to the Main Screen (same for the back button of the phone)
- Settings
- Statistics

3.4 calendar views

3.4.1 calendar

A calendar have to be displayed with the option to display only one of the calendars or more calendars at once. The view have to be similar to the calendar view in the web application.

3.4.2 calendar cardview

There have to be an option to see all calendars in one list. This list have to have an cards design. The Calendars image has to be displayed at the top of each card, underneath the image have to be more information about the calendar, such as name, subscribers, new subscribers this week. And the ability to search through the calendars or sort them by name, creation date, subscribers. In the Calendar card should be a button to get into the event cardview for this calendar. And a button for sharing.

3.4.3 event cardview

There have to be an option to see all events per calendar in a list. On the top you can swipe through the calendars. The events have to be displayed in cards design, too. With information about name, date and time.

3.5 Statistics

In this site statistics have to be shown. As described in the main application before (step 1 and 2).

3.6 calendar share functions

There have to be an option to share the subscription link of the calendar via the build in share functionality of the mobile phone. The link has to be at least shareable through twitter, instagram, whatsapp, email, sms and facebook.

3.7 Settings

Here the user can edit his settings, like described before in step 1 and 2.

4. miscellaneous

4.1 settings

eMail credentials have to be saved in a config file.

In superadmin login it has to be possible to change the superadmin password.

4.2 sending eMails

To send an eMail, we have an extra program called "mail-daemon" you can access it via Rest/API per http request in application/json format. You always get an true with an eMail id back, the traffic is encrypted via openssl.

The detailed description will be provided when we award this project to you, the description comes with sample php.

4.3 pb-login

We have to have an api to login, via an other login system via api. If we come over this api, the database name and server have to be overridden. And the user have to be logged in. With checking if the company_id fits. This api is also standard application/json http-request.

The traffic is encrypted via openssl. The detailed description will be provided when we award this project to you, the description comes with sample php.

4.4 Maintenance mode.

There have to be a ability to set the System in maintenance mode. Within this mode no login is possible, but super admin login.

4.5 Database

All user related data have to be encrypted, for example Addresses, eMail-address and names. The "Migration" feature of laravel have to be used. If there is something that have to be pre filled, when a new database is created, there have to be a defined seed for that.

All tables have to have the "datetime" fields for last update and creation timestamp. Along with the company id.

General connection:

Each element has only one company, each company has multiple calendars. Each calendar has multiple events. Each event has only one calendar. Each subscriber (id) has one calendar and one company_id. A calendar has multiple subscribers.

Structure Draft:

table : users (use the users table auto generated by laravel as a base)

- id
- company_id
- name
- email
- password
- password_confirm
- status
- timestamps

table : company

- id
- name
- street name and number
- Postalcode
- City
- Country
- telephone
- email
- website
- tax_id
- IBAN
- BIC
- license_model
- license_expire_date
- salesforce
- timestamps

table : company_configuration

- id
- company_id
- default_private
- default_email
- default_calendar_id
- timestamps

table : calendars

- id
- company_id
- name
- description
- tags
- public (t/f)
- status (t/f)
- picture
- not (t/f)
- timestamps

table : events

- id
- company_id
- calendar_id
- summary
- description
- destination
- date_from
- date_to
- not1 (datetime)
- not2 (datetime)
- not3 (datetime)
- timestamps

table : subscribers

- id
- company_id
- email
- name
- description
- calendar_id
- status
- link-hash
- timestamps

table : subscriber_logs

- id
- company_id
- subscriber_id
- user_agent
- location_la
- location_lo
- ip
- timestamps

table : license_models

- id
- name
- description
- price
- language
- timestamps

4.6 saving pictures

All pictures should be saved in an s3 bucket on amazon web service. The pictures have to be down sampled to the resolution that is max needed. Therefore we will give you a pre defined javascript/ php component that you have to use.

4.7 technical documentation

There have to be a technical documentation of the project. In this should be described how the program works, how the data are saved and which components are used.

5. Milestones

1.	Step 1	25 %
2.	Step 2	25 %
3.	Step 3	25 %
4.	bugfixing / testing	25 %

The 4th milestone contains an intensive test phase in which all bugs are repaired. The technical documentation is also included.

The Milestone 2 and 3 should be done in parallel.