CT10A7011, Running a Software Project

AssocEase

User Manual

by
Janne Pennanen, Jesse Pasanen, Eduardo Lopez, Miro Hakuli, Nestori Kangashaka,
Rémi Laramas

Table of Contents

Introduction

AssocEase is a website that allows Finnish association's members to save time on 3 different aspects of an association life: registering members, scheduling events, and contacting people. The goal is to produce an open-source application that can be used as a starting point to develop a bigger one.

Team

This project was developed by the following team over a period of 4 month :

- Remi Laramas (remi.laramas@gmail.com)
- Janne Pennanen [janne.pennanen@student.lut.fi]
- Jesse Pasanen (jesse.pasanen@student.lut.fi)
- Eduardo Lopez Palomo(Eduardo.Lopez.Palomo@student.lut.fi)
- Nestori Kangashaka (nestori.kangashaka@student.lut.fi)
- Miro Hakuli (miro.hakuli@student.lut.fi)
- Shah Waqer (shah.waqer.kabir@student.lut.fi)

For local user

At the home page users can see the ongoing events. From the menu bar users can navigate to the desired options.

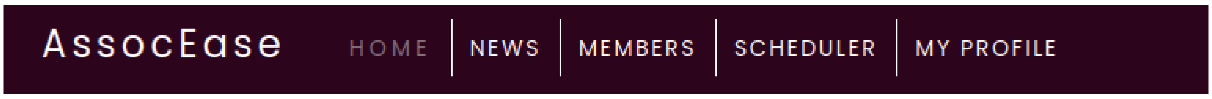
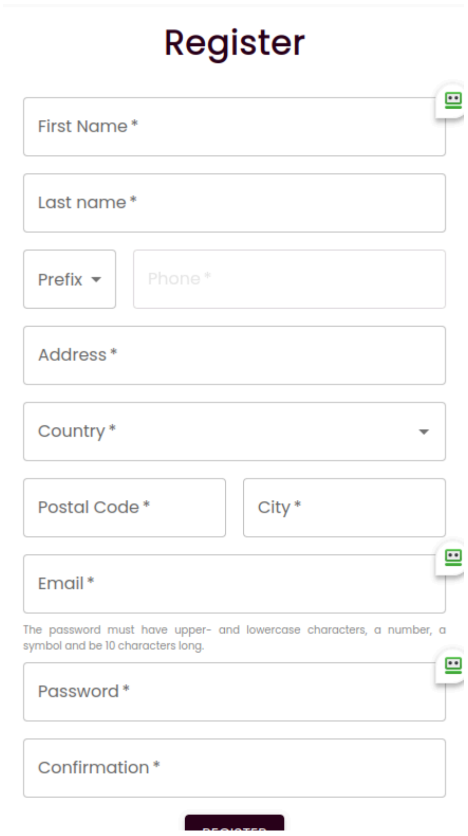


Figure 1 : Top bar

Normal Users

Registering

Before being able to do anything in the app, the user must either login or register a new account and login. Once the button register has been triggered, user needs to enter the * required fields with valid information. As the titles are shown in the input fields as placeholders, users should enter the information as per the form.



Register

First Name *

Last name *

Prefix Phone *

Address *

Country *

Postal Code * City *

Email *

The password must have upper- and lowercase characters, a number, a symbol and be 10 characters long.

Password *

Confirmation *

Figure 2 : Register Form

Login

It is a classic login screen with a regular login page behavior.

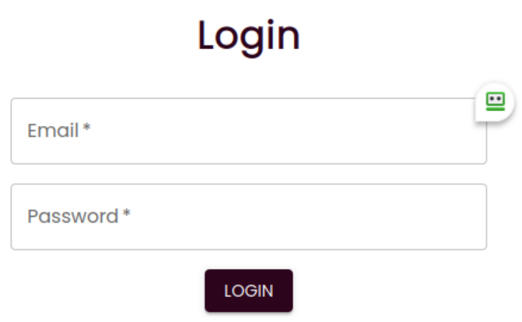
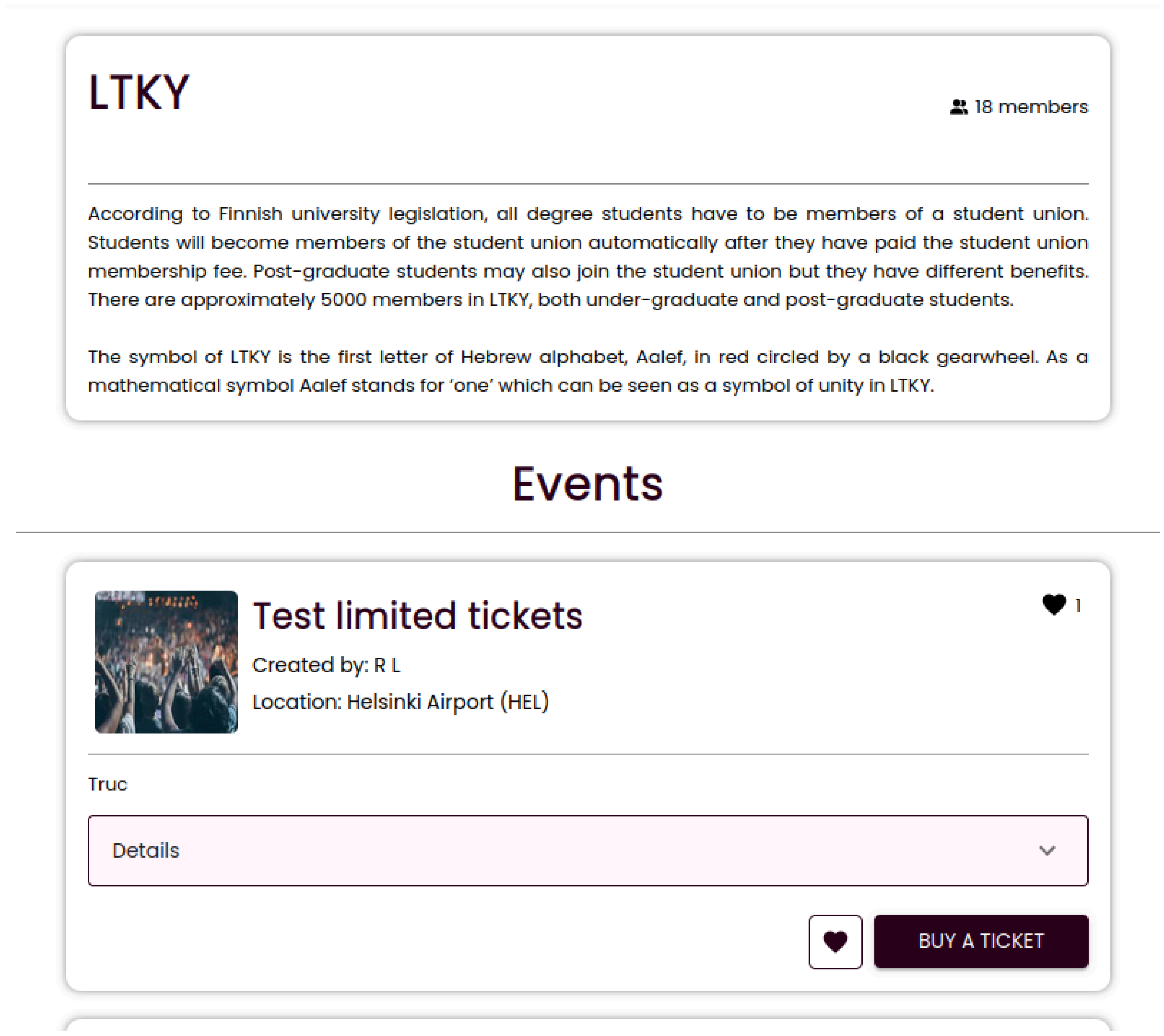


Figure 3 : Login Form

Home

From the home page, a normal user can see the different events separated in two categories, the past and future ones. By expanding the details section, the user should be able to see all event's details, including a map showing the address where the event will be held.
Liking an event or buying a ticket for it would make it appear in the "My events" section of the My profile page.


LTKY

18 members

According to Finnish university legislation, all degree students have to be members of a student union. Students will become members of the student union automatically after they have paid the student union membership fee. Post-graduate students may also join the student union but they have different benefits. There are approximately 5000 members in LTKY, both under-graduate and post-graduate students.

The symbol of LTKY is the first letter of Hebrew alphabet, Aalef, in red circled by a black gearwheel. As a mathematical symbol Aalef stands for 'one' which can be seen as a symbol of unity in LTKY.

Events

Test limited tickets

1

Created by: R L

Location: Helsinki Airport (HEL)

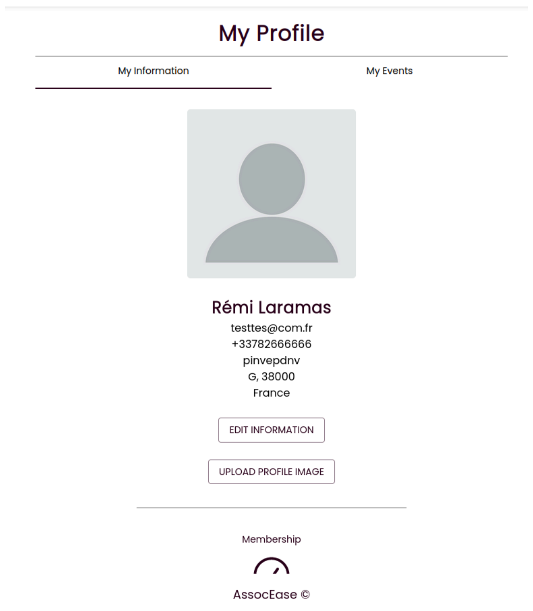
Truc

Figure 4 : Home Page

My Profile

From this page, a user can perform multiple actions :

- Edit their credentials
- Edit their billing information
- Pay their membership for being a part of the association
- Upload their profile image (max size is 2Mb)
- See and interact with the events they liked, bought a ticket to, or created themselves


My Profile

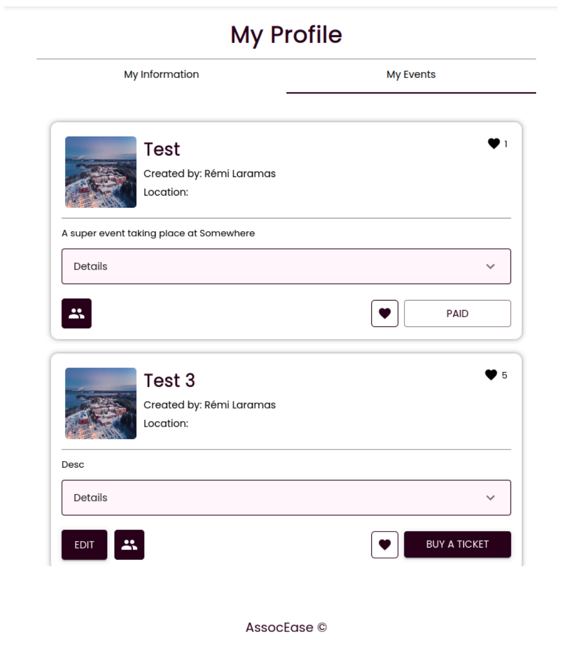
My Information | My Events

Rémi Laramas
testtes@com.fr
+33782666666
pinvepdnv
G, 38000
France

EDIT INFORMATION
UPLOAD PROFILE IMAGE

Membership

AssocEase ©


My Profile

My Information | My Events

Test
Created by: Rémi Laramas
Location:
A super event taking place at Somewhere
Details
PAID

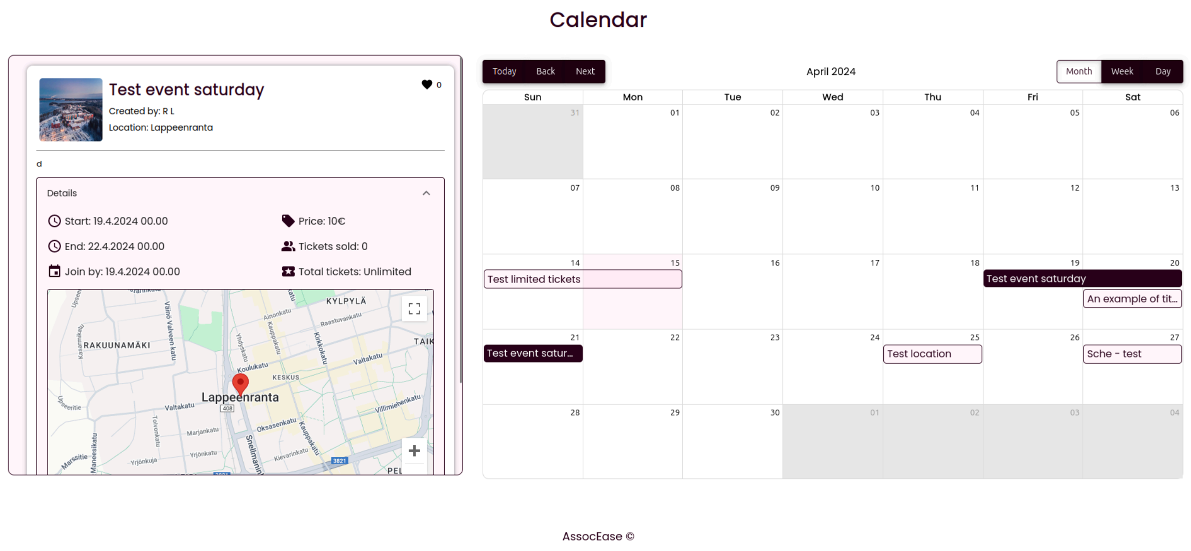
Test 3
Created by: Rémi Laramas
Location:
Desc
Details
EDIT
BUY A TICKET

AssocEase ©

Figure 5 : My Profile Information Tab *Figure 6 : My Profile events tab*

Scheduler

This page is another view for the user to see the global planning differently. In this view, events are spreaded across a calendar, according to their information (starting time, ending time, etc). As on the home page, the user can interact with event cards. Three possible displays are available : monthly, weekly and by day.


Calendar

Test event saturday
Created by: R L
Location: Lappeenranta
d
Details
Start: 19.4.2024 00.00
End: 22.4.2024 00.00
Join by: 19.4.2024 00.00
Price: 10€
Tickets sold: 0
Total tickets: Unlimited

Today Back Next | April 2024 | Month Week Day

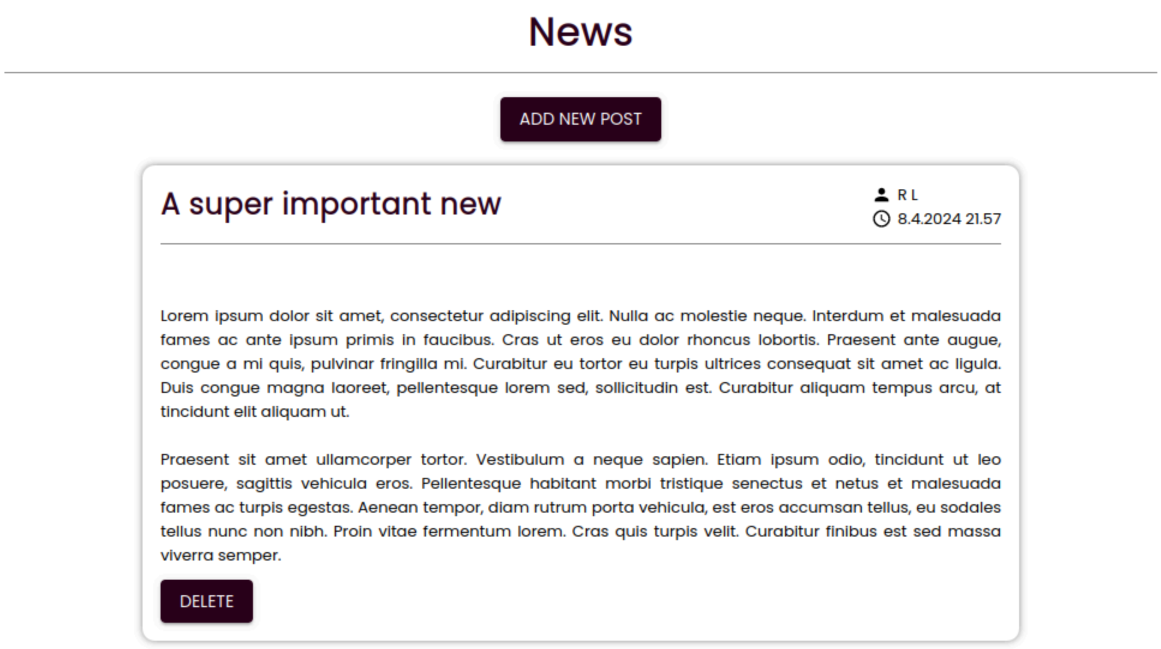
Sun	Mon	Tue	Wed	Thu	Fri	Sat
31	01	02	03	04	05	06
07	08	09	10	11	12	13
14 Test limited tickets	15	16	17	18	19 Test event saturday	20 An example of tit...
21 Test event satur...	22	23	24	25 Test location	26	27 Sche - test
28	29	30	01	02	03	04

AssocEase ©

Figure 7 : Scheduler view

News

A user can have their finger on the pulse about the association by going to the news page. It is a list of all the posts that have been written, its objective being to share quick information.


News

A super important new

R L
8.4.2024 21.57

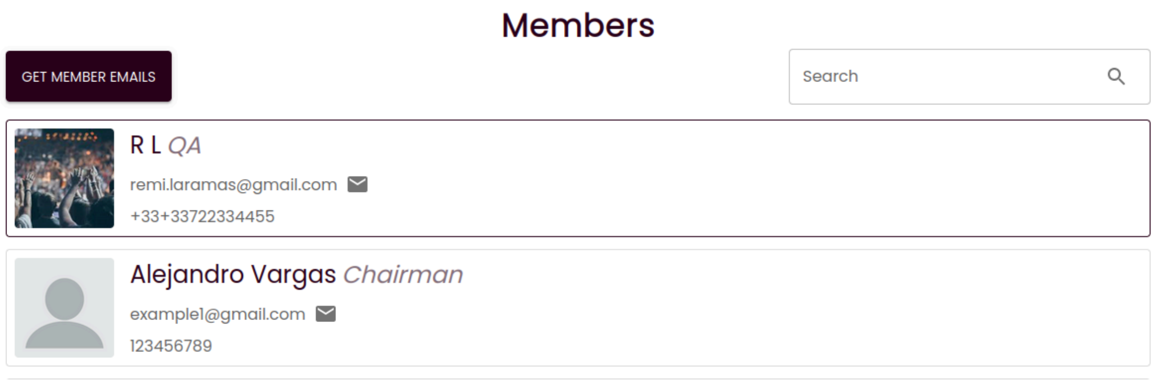
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla ac molestie neque. Interdum et malesuada fames ac ante ipsum primis in faucibus. Cras ut eros eu dolor rhoncus lobortis. Praesent ante augue, congue a mi quis, pulvinar fringilla mi. Curabitur eu tortor eu turpis ultrices consequat sit amet ac ligula. Duis congue magna laoreet, pellentesque lorem sed, sollicitudin est. Curabitur aliquam tempus arcu, at tincidunt elit aliquam ut.

Praesent sit amet ullamcorper tortor. Vestibulum a neque sapien. Etiam ipsum odio, tincidunt ut leo posuere, sagittis vehicula eros. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Aenean tempor, diam rutrum porta vehicula, est eros accumsan tellus, eu sodales tellus nunc non nibh. Proin vitae fermentum lorem. Cras quis turpis velit. Curabitur finibus est sed massa viverra semper.

Figure 8 : News page

Members

This page represents the list of actual registered members of the association. A normal user can copy email addresses and see user details.


Members

R L *QA*
remi.laramas@gmail.com
+33+33722334455

Alejandro Vargas *Chairman*
example1@gmail.com
123456789

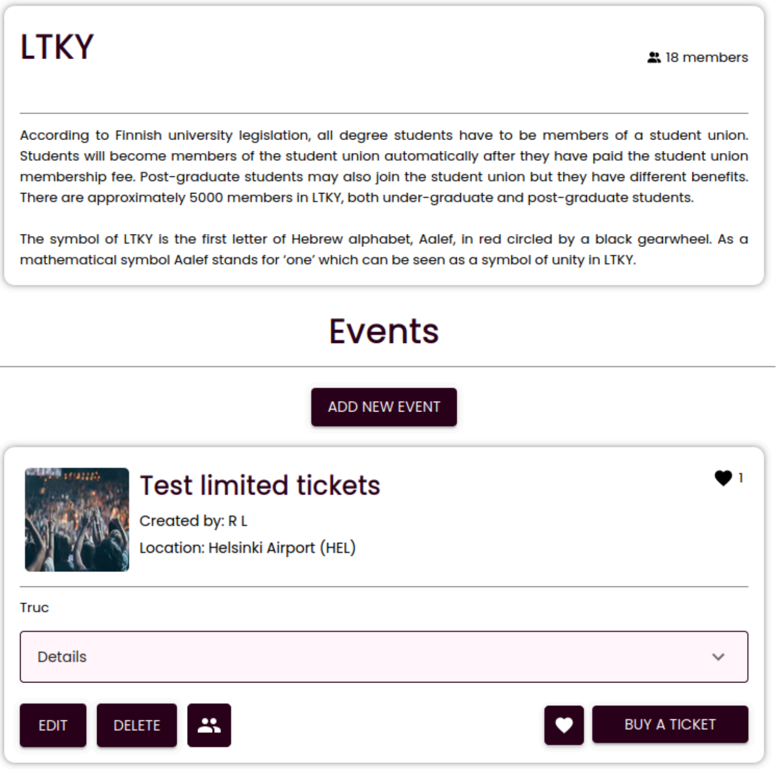
Figure 9 : Members Page

Administrators

An administrator is considered as someone that can manage an association. To do so, defining an account as administrator gives it the following rights :

- Create, edit and delete events
- See Membership status for every member
- Create posts
- Edit certain members account information

Events


LTKY

18 members

According to Finnish university legislation, all degree students have to be members of a student union. Students will become members of the student union automatically after they have paid the student union membership fee. Post-graduate students may also join the student union but they have different benefits. There are approximately 5000 members in LTKY, both under-graduate and post-graduate students.

The symbol of LTKY is the first letter of Hebrew alphabet, Aalef, in red circled by a black gearwheel. As a mathematical symbol Aalef stands for 'one' which can be seen as a symbol of unity in LTKY.

Events

ADD NEW EVENT

Test limited tickets

Created by: R L

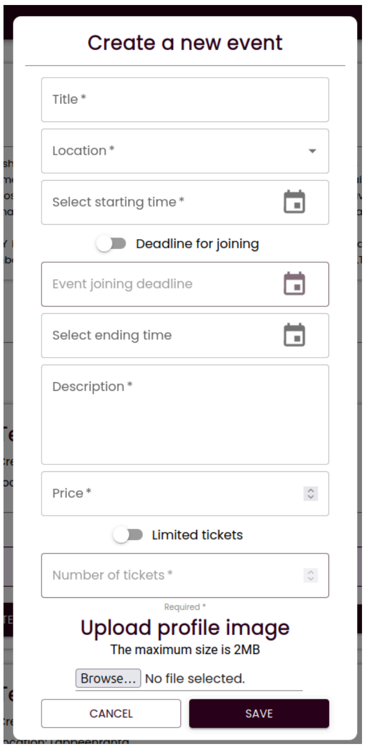
Location: Helsinki Airport (HEL)

1

Truc

Details

EDIT DELETE BUY A TICKET


Create a new event

Title *

Location *

Select starting time *

Deadline for joining

Event joining deadline

Select ending time

Description *

Price *

Limited tickets

Number of tickets *

Required *

Upload profile image

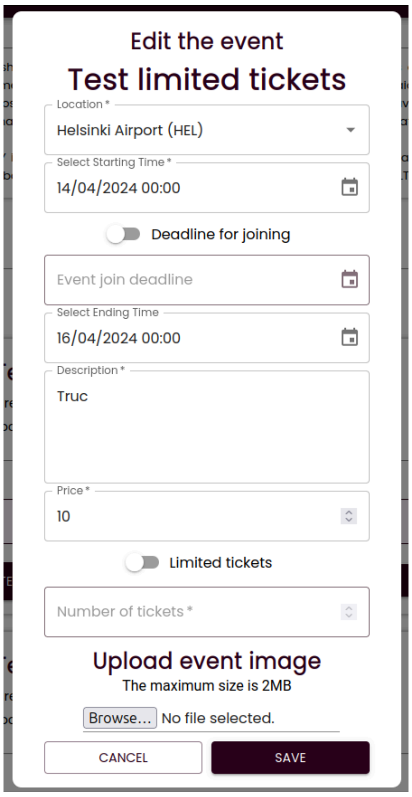
The maximum size is 2MB

Browse… No file selected.

CANCEL SAVE

Figure 10 : Home Page as admin

Figure 11 : Creating a new event


Edit the event

Test limited tickets

Location *: Helsinki Airport (HEL)

Select Starting Time *: 14/04/2024 00:00

Deadline for joining

Event join deadline

Select Ending Time: 16/04/2024 00:00

Description *: Truc

Price *: 10

Limited tickets

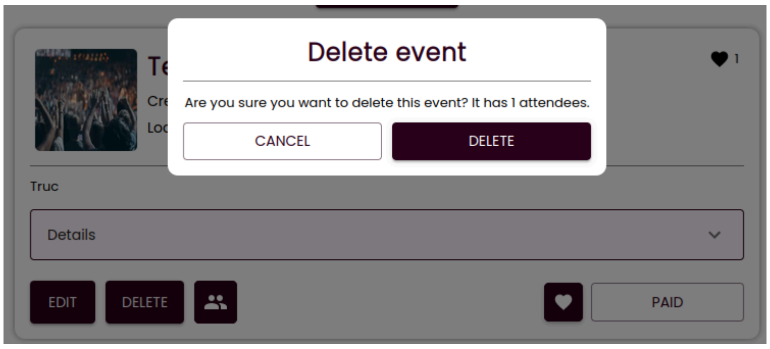
Number of tickets *

Upload event image

The maximum size is 2MB

Browse… No file selected.

CANCEL SAVE

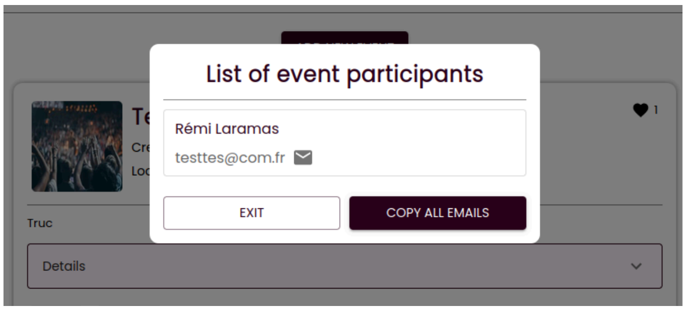

Delete event

Are you sure you want to delete this event? It has 1 attendees.

CANCEL DELETE

Figure 12 : Editing an event

Figure 13 : Deleting an event

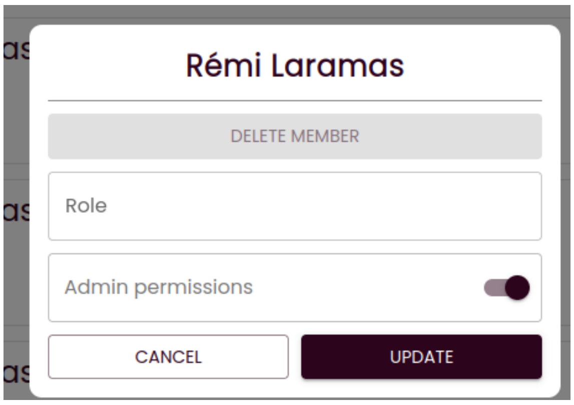

List of event participants

Rémi Laramas
testtes@com.fr

Figure 14 : Get the list of participants from an event

As an admin, events can be created, edited and deleted. Everywhere the event card appears, an admin will be able to perform these 3 actions. However, editions can't be performed on passed events. Also, on each event, an admin has the possibility to get the address of every person that has paid a ticket for that event.

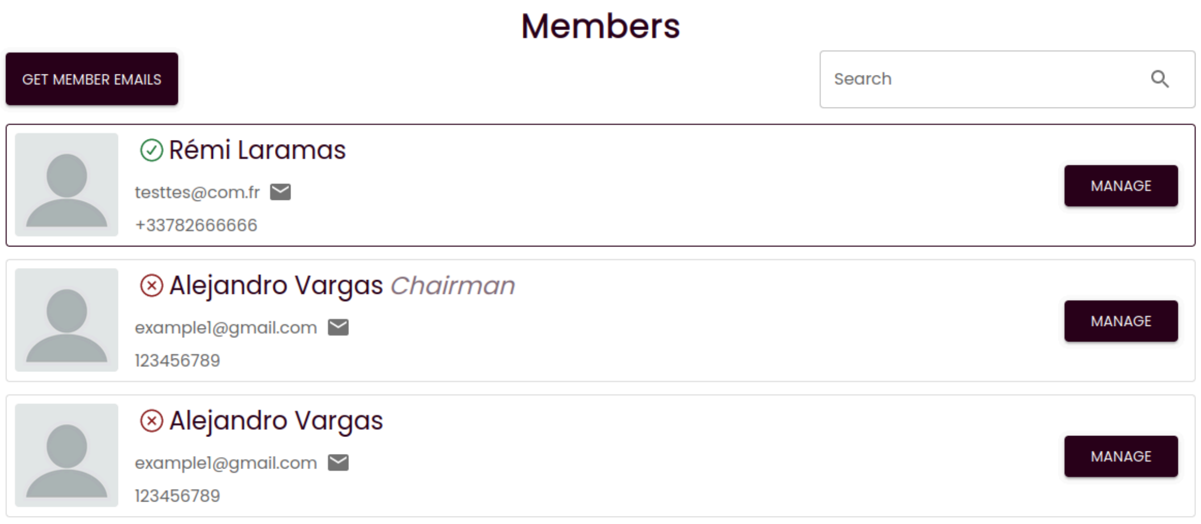
Members / Membership management


Rémi Laramas

Role

Admin permissions: on

Figure 15 : Editing a member's profile


Members

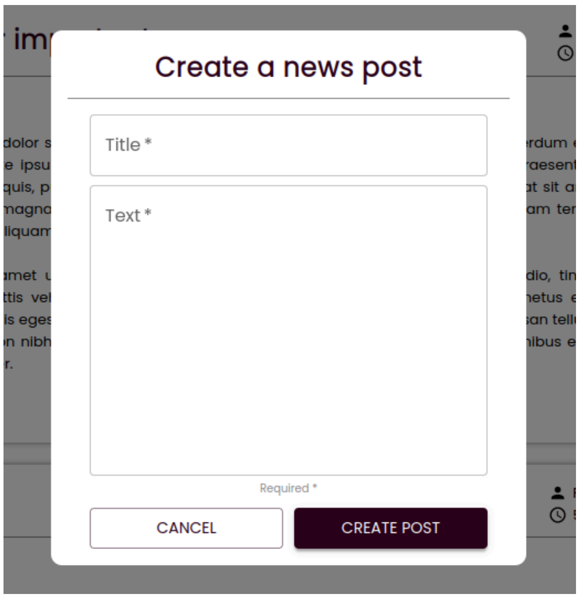
Rémi Laramas	testtes@com.fr, +33782666666
Alejandro Vargas *Chairman*	example1@gmail.com, 123456789
Alejandro Vargas	example1@gmail.com, 123456789

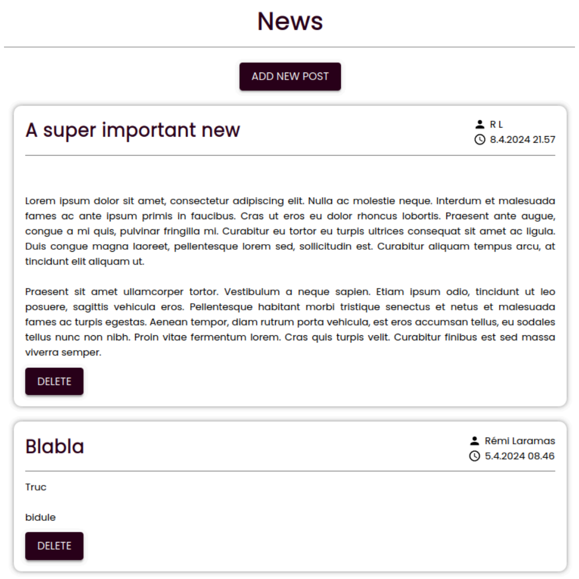
Figure 16 : Member's page as an admin

As an admin, a user can modify the role and the admin privileges of a member account. He can also delete these accounts. The role has no effect other than giving a title to the user, more details about this feature in the later parts of this document.

The green and red marks shown aside the names in figure 16 illustrate whether a user has a membership or not.

Posts management





A super important new

R L
8.4.2024 21.57

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla ac molestie neque. Interdum et malesuada fames ac ante ipsum primis in faucibus. Cras ut eros eu dolor rhoncus lobortis. Praesent ante augue, congue a mi quis, pulvinar fringilla mi. Curabitur eu tortor eu turpis ultrices consequat sit amet ac ligula. Duis congue magna laoreet, pellentesque lorem sed, sollicitudin est. Curabitur aliquam tempus arcu, at tincidunt elit aliquam ut.

Praesent sit amet ullamcorper tortor. Vestibulum a neque sapien. Etiam ipsum odio, tincidunt ut leo posuere, sagittis vehicula eros. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Aenean tempor, diam rutrum porta vehicula, est eros accumsan tellus, eu sodales tellus nunc non nibh. Proin vitae fermentum lorem. Cras quis turpis velit. Curabitur finibus est sed massa viverra semper.

Blabla

Rémi Laramas
5.4.2024 08.46

Truc

bidule

Figure 17 : Creating a new post

Figure 17 : News view as an Admin

Being an administrator also allows the user to create and delete posts in the news page.

For developers

App design

Software architecture

Assoc Ease is a classic website app that is built upon **REACT** coupled with the **MUI** css framework, **Express** js for the backend and **Mongodb** for the database. The following sections aim to explain the structure and how to improve the actual implementation of the system.

Functional view

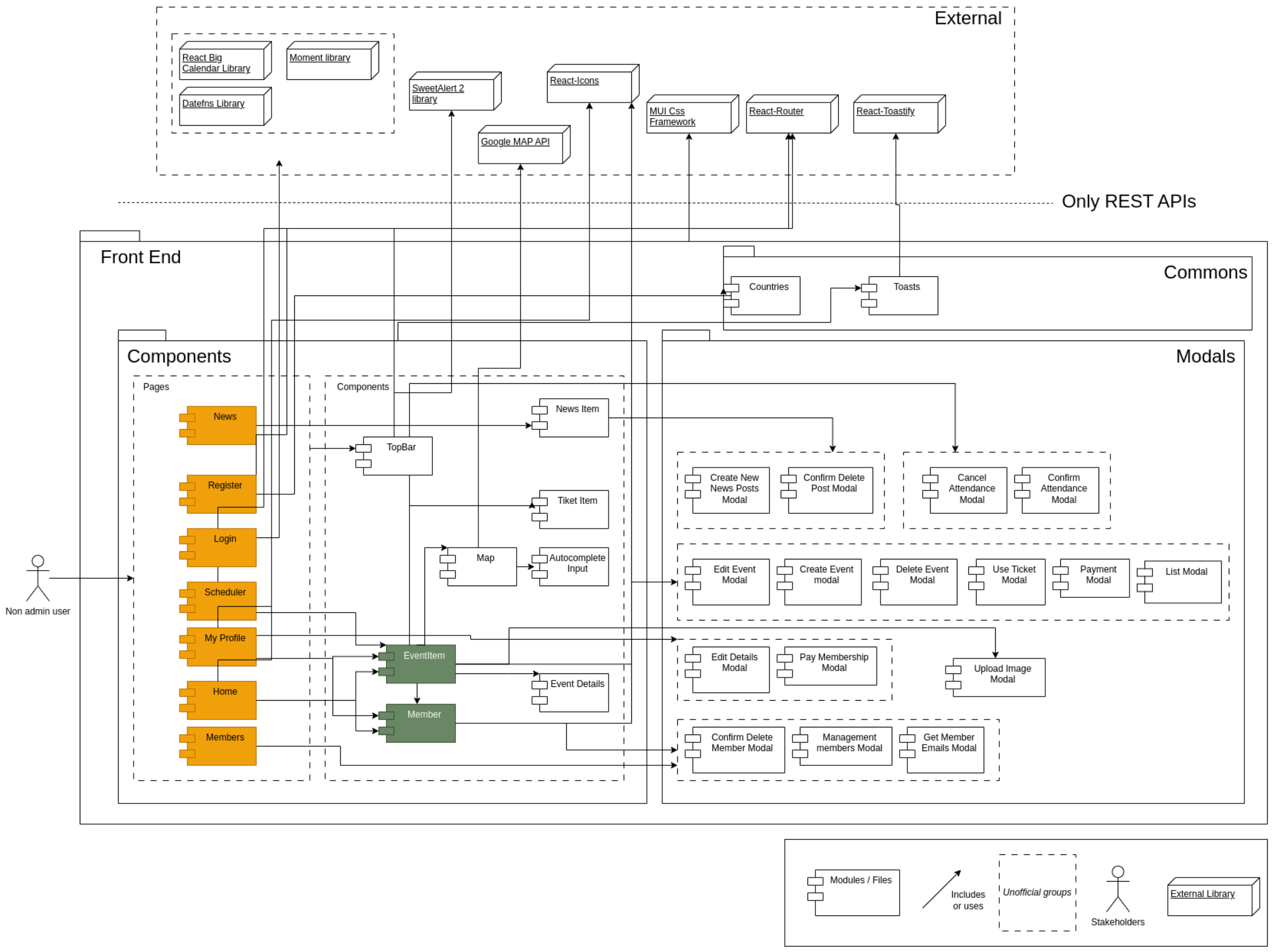


Figure 18 : Functional View

The figure above represents the actual state of the Front-End application part. Using this diagram and with the aid of the code, you should be able to understand

dependencies between components, and be able to target the components using the different libraries more easily.
You should take a look at it if you want to modify the dependencies (For example switching to another map SDK) or refactor the code in general.

Database diagram

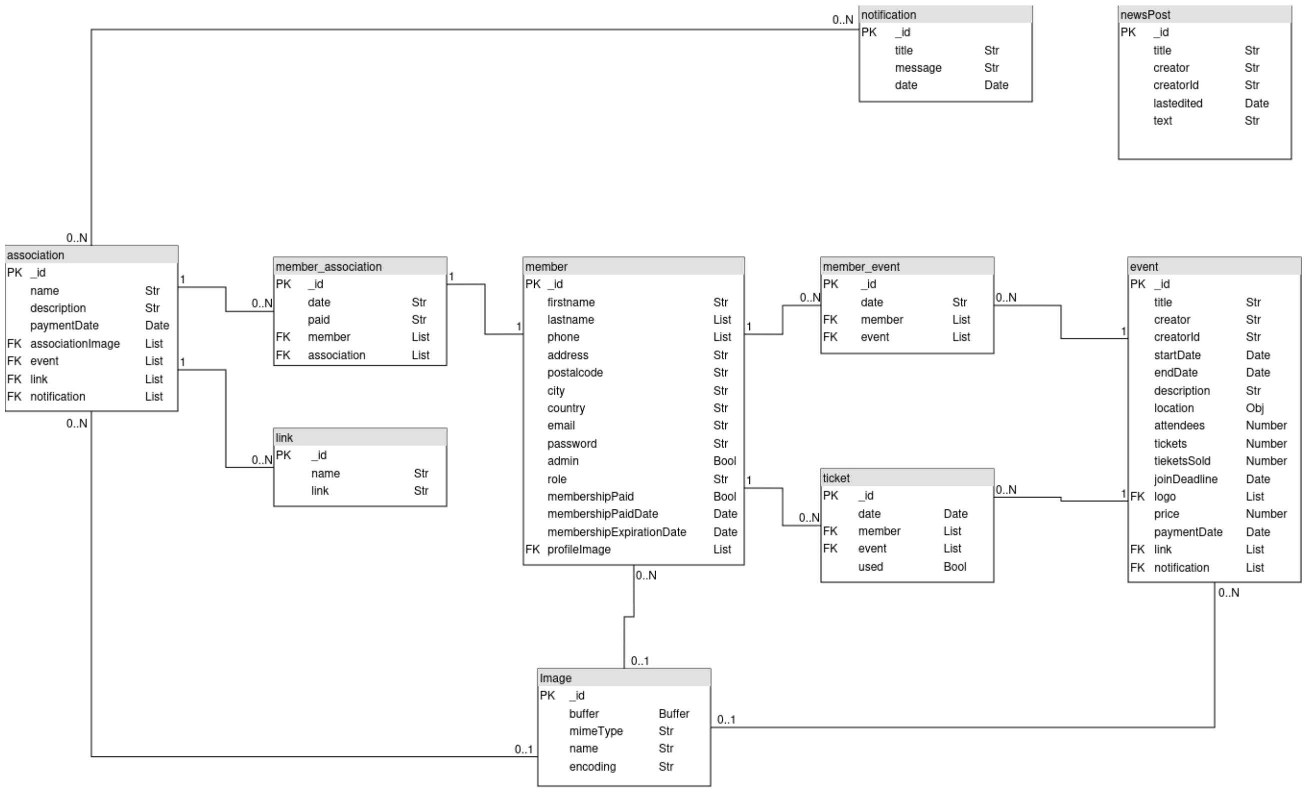


Figure 19 : Database diagram

The above diagram is a representation of the actual implementation of the databas's system. It has been developed using MongoDb for accessibility reasons, but it was thought of as a relational database upon some aspects (member_association, member_event, etc).
The meaning of each field is described in the corresponding source files.

Installation

Dependencies

To be able to develop new features on top of our project, you will need the following softwares and packages installed on your machine beforehand :

- NPM and NodeJS - Official installation documentation
- Mongodb, both ways have been tested and work
  - Atlas (online) - Mongodb Atlas
  - Community edition (local) - Mongodb Community Edition
- Git - Official git documentation

- Google MAP API - [Official Places API Documentation]

Once everything is installed, you can head on to the next section.

The project itself

To download and install our project, open a terminal and go to the directory where you want the project to be. Type the following commands :

```
# HTTPS
git clone https://github.com/jannejjj/RASP24.git
# or with SSH
git clone git@github.com:jannejjj/RASP24.git
```

The project should now be in the ***./RASP24*** subfolder.

To install the dependencies linked to the project, you can simply do the following :

```
cd RASP24
npm install
```

You should now have a project ready to be built. Before moving to the running part, it is necessary that we have configuration files to both backend and frontend sub-projects for the system to run.

You can do so by creating files named ***.env*** in both of ***./client*** and ***./server*** subfolders. You can then fill the files with the following contents :

---

server/.env

```
SECRET="something_here"
MONGODB_CONN_STRING="mongodb+srv://your_mongo_connecting_string"
```

- **SECRET :** is the value used to encrypt the data sent between the Front-End and the Back-End. It can be anything.
- **MONGO_CONN_STRING :** is the connection string to your mongo database, either the online or the local version of it :
    - local should look like this : *“mongodb://localhost:27017/assoceasedb”*
    - online should look like this : “mongodb+srv://your_mongo_connecting_string”

---

client/.env

```
REACT_APP_PROXY_TARGET="http://server:4000"
REACT_APP_API_KEY="your_key"
```

- **REACT_APP_PROXY_TARGET :** is used to route requests correctly to the Docker backend. If you are not using docker you should replace "server" by "localhost".
- **REACT_APP_API_KEY :** is used by the Google Map API to identify yourself to their services. It is needed if you want Event location maps to work.

Running the app

To run the app you can simply run the following commands from the top folder :

```
cd RASP24/
npm run dev:server
(in another terminal window) npm run dev:client
```

Or, if you have Docker installed:

```
cd RASP24/
docker-compose up --build
```

If you haven't received any error messages on the consoles, everything is working correctly. You should be able to look at AssocEase on **localhost:3000**.

What's next

From the beginning, the app has been thought of as a skeleton, a basis from which associations and student unions can build something greater and more suited to their needs.
The following paragraphs will give you an overview of what the original team of the project thought about and didn't complete so you have a starting point.

Notifications and Reminders

File	Function	Line	Utility
server/api.js	SendEventEditionNotification	828	Call CreateAndSendNotifications
	CreateAndSendNotifications	833	Working demonstration of an email parameterization and send.
	router.post("/editEvent" ...)	519	Call

			sendEventEditionNotification
	cron.schedule	738	Same concept as CreateAndSendNotifications but at a regular schedule and for deadline reminders.

Table 1 : Files related to Notifications and Reminders

One of the main branches that hasn't been completed is the notification one. As we built the app, we thought that to have a complete and reliable system, it should be necessary to have a system to remind people about the events and payment deadlines. Only a sketch was built until now.
The above list of functions is a good starting point if you want to get into the details of email sending. If you want to dig more, we suggest you to look at the following points :

- **Security improvements** : Password and email address are written in clear in the code, a first code refactor should be done to securise this more (.env variables, etc)
- **Email templates :** It is totally possible to make custom good looking emails. If you want to do so, you should take a look at MJML which has a great framework and templates you could elaborate on.
- **More reminders :** You have the code for basic reminders and notification, but there are a lot of application events for which the users and/or participants should be informed. When a news article is posted for example, or when an event is deleted : everything is possible !

Security Improvements

File	Function	Line	Utility
server/api.js	router.post('/register', ...)	161	Register a new user, with a blank role. Does this by decrypting what the front encrypted.
	router.post('/login', ...)	108	Comparing encrypted data to log in.
client/Register.js	submitForm	53	Encrypting and sending a registration request.

Table 2 : Files related to encryption

One of the main last steps before the project can be considered for production use is the security of it. Until now, only account's passwords are stored encrypted, and it is the first level of security. To improve the security of the software we suggest you to dive into the next few points :

- **SSL Certificates** : One easy way to upgrade the security of a website for SEO and make it trustworthy, is the use of SSL certificates. We suggest you look at Let's Encrypt as a starting point.
- **CSP (Content Services Policy) :** The MDN web docs upon this subject is really complete : it helps understand the concept and gives you common examples. This technology is really useful for preventing data injection attacks, and a lot of other ones.

Roles

File	Function	Line	Utility
client/Member.js	updateMembers	72	Update member's information and its role
server/api.js	router.post('/register', ...)	161	Register a new user, with a blank role.

Table 3 : Files related to roles

In the original implementation of the project, only a small role feature was implemented to let associations choose how to handle them, and what to do with them. That's why it is only represented as a String in the database and that nothing is done with it.
To get more in the details of implementing a complex role gesture, we suggest you look at what Discord made : it has a complete set of features related to this subject.

Conclusion

This project has been built by a team of 7 people during a course made for it. Our objective behind it was to challenge ourselves into doing something that works with the minimum amount of bugs possible, the maximum of features possible and that could be used for real.
In that sense we think that it is now possible to take the project and try to implement new features from it : different dashboards, treasurer responsibilities, reactiveness, etc.
We hope that you will enjoy playing around with the system, and we say thanks to you !

AssocEase Team