Vrije Universiteit Brussel

Web Technologies

2016-2017

Project Verslag

Naam:	*Rolnummers:*
Jasper Busschers	513534
Jens Cardon	0515320
Carlos Montero	0500677

December 19, 2016

Contents

1 Introduction

Zomomo is a website based on the frustrations we had with platforms like facebook. Where facebook doesn't provide you with the option to search for events by anything else then it's name, we decided to make it our project to create an easy to use website to browse through events. The framework we decided to use is dot net framework (asp.net mvc). None of us had any experience with this platform nor C-Sharp but that is excactly why we wanted to use it, and also because this would be easy to convert to a mobile app using xamarin. To supplement this we decided to use tools from Microsoft Azure to host our database and website.

2 API's

2.1 Cloudinary

Cloudinary is a cloud api to store and retrieve images aswell as performing transformations on retrieving. We use the cloudinary api to store and display pictures for both users and events. Whenever an image gets uploaded it returns a public key, this key we store for every instance of a user or event. This public key we can use afterwards to request the image and display it on our page.

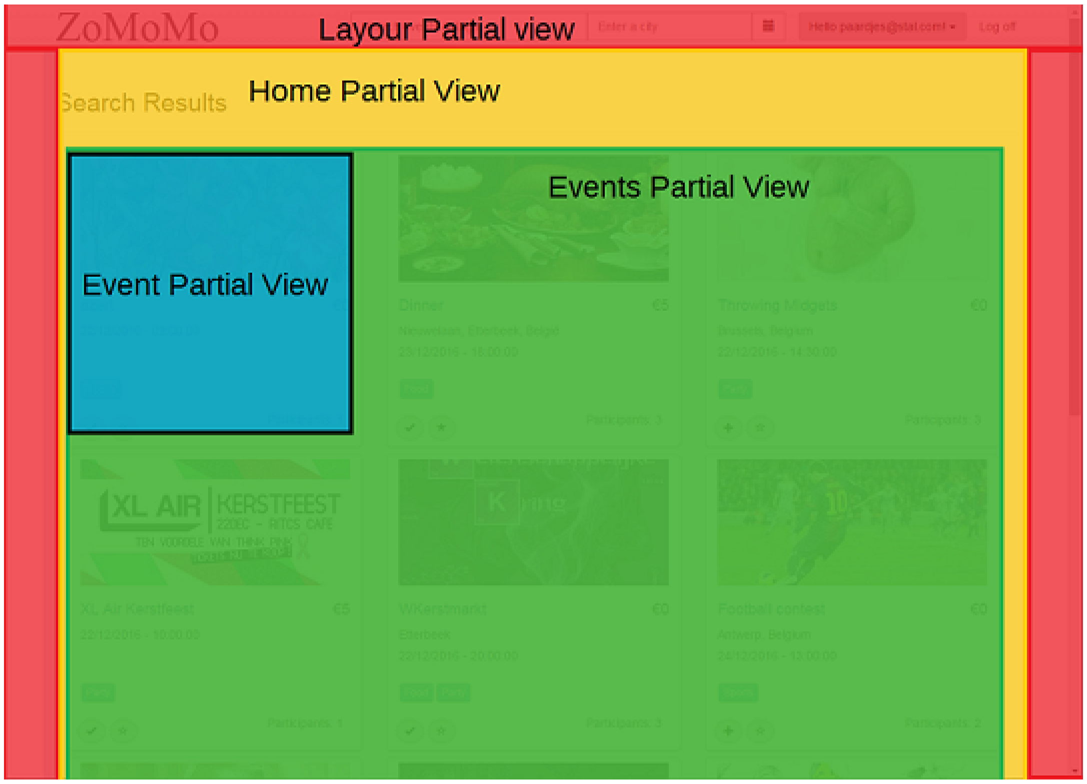
2.2 Google

We use the google api in 2 ways, we use it to display a map of the event location on the details page of that events. And we also use this api to generate a list of potential results upon typing in a text box that is defined to fill in a location.

3 Home

The home page is the heart of our web appliation, it gives the user a listed version of the available events corresponding to the search conditions you may have set. We wanted to make this accesible from anywhere so that no matter on what page the user is, he will be able to quickly search events without losing your current page. The illustration below will provide you with more insight in how we do this.

3.1 Partial View



We tried to have as little redundency as possible in our views, this we achieved by using a lot of abstraction in the way we designed our views. The layout partial view is the one where the navigation bar, script/css imports and shared functionality is written. this Partial view captures all the other views. In the layout partial view is also some ajax implemented to be able to access search results from any page. When we go to the home page, the Home partial view gets loaded, with an instance of an EventViewmodel, which contains

all the events at that time. Then it calls the Events partial view with the list of events, which in hes turn calls a partial view to draw a single event and append these together. This requires a single event and the correct instance of the association if the user owns one with that event. This is to correctly display the button based on if the user is going or not.
When ever something is written in the upper search bar, the events partial view gets called again but with the result of the querry formed by the search form.

(note: The dateTimePicker is set to the current day by default if you are not on the home-page. We didn't want this to happen, it is a small bug. Anyway, you can set the timespan manually!)

4 Account

The user is able to search for events but when he wants to create an event he first has to log in. This can either be done by registering on our site or logging in via facebook.
Registration into the website is only possible if the registration form is filled correctly. The requirements for the password are thr following:
For the password :

- Passwords must have at least one special character.
- Passwords must have at least one lowercase ('a'-'z').
- Passwords must have at least one uppercase ('A'-'Z').
- Password must have at least six characters.

Apart from password validation we require that the e-mail address is a valid instance of an e-mail, require their name and a small description. And we give the option to add a profile picture.
Each user is able to look up his information and edit it by using action links respectively called My profile and Settings. Under "My profile" we can find information about the user's description but also about his/her events history (see section 5).

4.1 AccountController

Account Controller is the controller that will handle all the server request involving the user. returning a view for the registration, profile, settings.
When we make a new project, dot net framework provides us with a lot of prepared code, so we just have to focus on the extension of the code. In our case we received a project that allows individual user authentication. Whenever a user is registered, the user is saved into the table that dot net provides. We decided to keep using this ability.
Their table contains the e-mail, a string ID and a password that has been hashed. Our table contains more information about an user. Now every time a user registers, it is first registered in their table and if everything went without errors we can easily conclude that we also can store the additional user information in our table.
Apart from their log-in table we have our own table that contains extra information we want to store about an account and every entity in our table shares the primary key with the corresponding id in the login table.

4.1.1 Profile action

This action will render a view that contains the current user information but also his event history. To generate this view we require the user information and logs from the database. These logs we use to get the history of that user.
In this action we made multiple requests to the database to fetch all the events belonging to the the current user. And classified the events into three categories:

- events organised
- events that the current user participates
- events that the user is interested

There are 3 buttons on this page that allow you to dynamically choose which of these lists you would like to see. This is accomplished by using Ajax with a content placeholder.
When the current user accesses his/her profile page, he also get the chance to edit his events or view the event page. Whenever another user access the current user profile page, this ability is disabled ofcourse.

4.1.2 settings action

Whenever a user decides to click on the actionlink "settings" he will be redirected to the view that allows the user to change his information. The user is allowed to change his first name, last name, date of birth, profile description and profile picture.
Here again for the profile picture we use the external service cloudinary.

5 Logbook

Logbook is an entity we designed to associate a single user to an event, here we store the user id, event id as well as 3 booleans variables named going, interested, organise. The booleans are used to determine the current relation the user has to an event and update it in case this changes.

- organise boolean: If this is set on true then the user is the one that created the event.
- going boolean: if this is set on true then the user will participate to this event.
- interested: If this is set on true then the user is interested in the event.

This logbook is supposed to represent the history of a user. The history of a current user can be seen by every user on the website.
At any particular time the user can edit his organised events and also decide to change his mind about going to an event.

6 Events

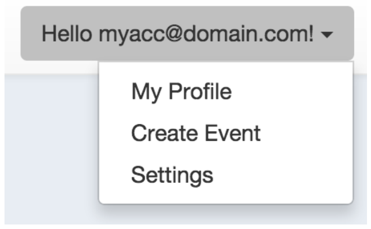
6.1 Database

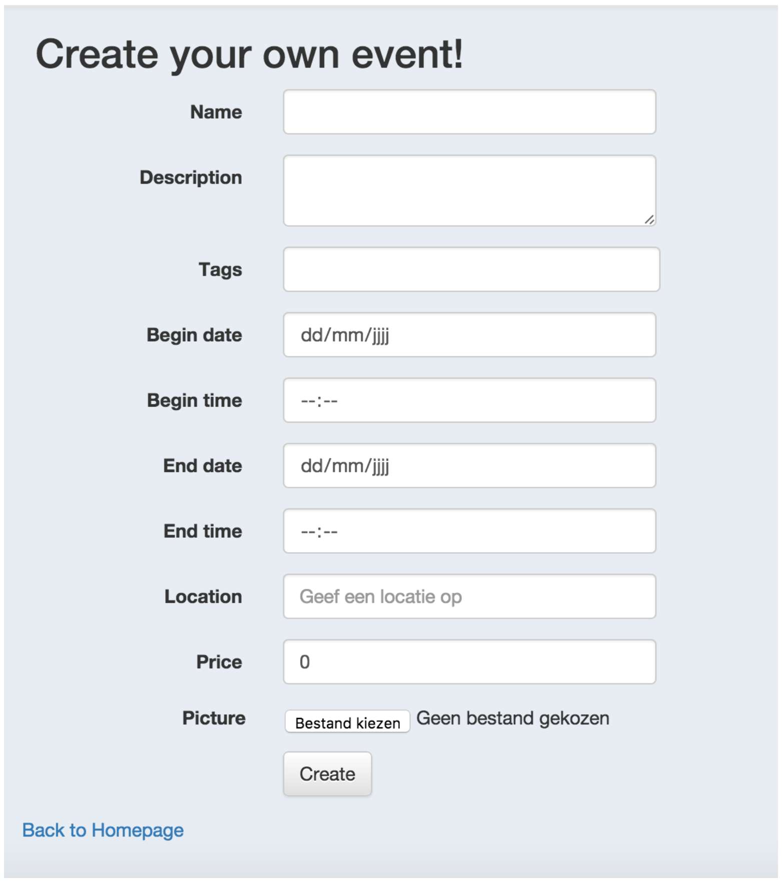
var	type	info
EventID	int	identity(1,1), primary key
EventName	varchar	max-length = 20
EventDescription	varchar	max-length = 300
EventBeginDate	DateTime	format = dd/MM/jjjj
EventEndDate	DateTime	format = dd/MM/jjjj
EventLocation	nvarchar	max-length = 100
EventPrice	int	default = 0
EventPicture	nchar	default image
EventParticipants	int	default = 1 (host)

6.2 Create Events

6.2.1 Website

When logged in, a user can create his own event.
Just a simple click on create event in the profile dropdown navigation bar (which will redirect you to www.zomomoo.azurewebsites.net/Event/CreateEvent) and some form fill-ins, will do.

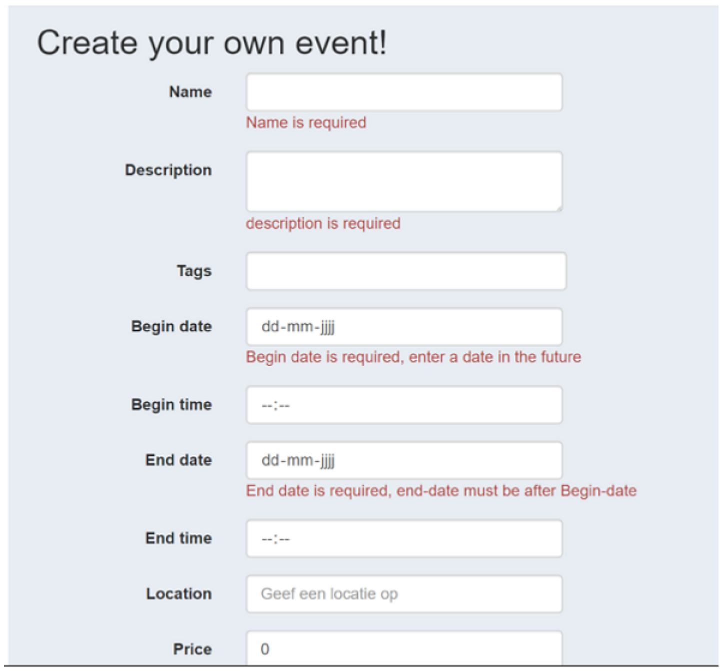



Create your own event!

Name	
Description	
Tags	
Begin date	dd/mm/jjjj
Begin time	--:--
End date	dd/mm/jjjj
End time	--:--
Location	Geef een locatie op
Price	0
Picture	Bestand kiezen Geen bestand gekozen

1. REQUIRED
    - EVENT NAME
        (a) the user cannot input more than 20 characters in this field
    - EVENT DESCRIPTION
        (a) the user cannot input more than 300 characters in this field
    - EVENT TAGS
        (a) the user cannot input more than 5 different tags in this field
    - EVENT BEGIN DATE + TIME
        (a) begin date cannot be in the past
    - EVENT END DATE + TIME
        (a) end date is later than begin date
    - EVENT LOCATION
        (a) uses the google maps API to autocomple location
    - EVENT PRICE
        (a) default value is 0 when not set
2. OPTIONAL
    - EVENT PICTURE
        (a) uses the Cloudinary API

If one of the input fields is not entered correctly, one or multiple errors will appear with specified instructions for the user.



Create your own event!

Name	
	Name is required
Description	
	description is required
Tags	
Begin date	dd-mm-jjjj
	Begin date is required, enter a date in the future
Begin time	--:--
End date	dd-mm-jjjj
	End date is required, end-date must be after Begin-date
End time	--:--
Location	Geef een locatie op
Price	0

6.2.2 EventControler

When the form is correctly filled in, the model is send to the EventControler. There, a function "FileUpload" will be called which takes one parameter: the model. It updates the database from the model and redirects the user to his newly made event.

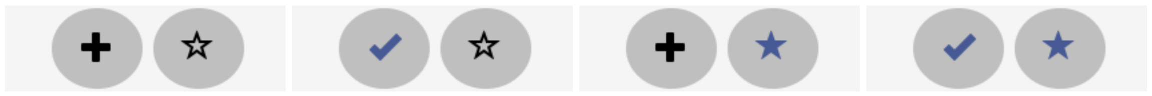
6.3 EventDetails

6.3.1 Website

A user can click on an eventName. This name contains a link that will redirect the user to the following page "www.zomomoo.azurewebsites.net/Event/EventDetails/id". On this page, all the details of the event will be displayed.
These are:

1. Picture
2. Tags
3. Name
4. Host(link to profile)
5. #Participants
6. #Intereseds
7. Begin-date & -time
8. End-date & -time
9. Google map with marker on location
10. Description

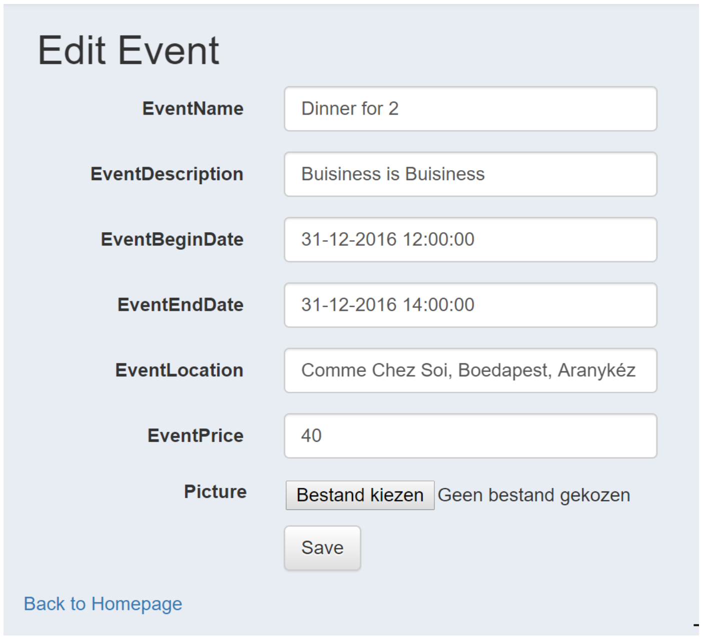
It is also possible for a user to participate or set up as interested for an event. The corresponding glyphicon will change when clicked and the logbook will be updated.



At the end of the page, the user can return to the homepage by clicking on "Home". An additional "Edit" link will be displayed on the EventDetails page if the user is the host of the event.

6.4 EditEvent

6.4.1 Website



Edit Event

EventName	Dinner for 2
EventDescription	Buisiness is Buisiness
EventBeginDate	31-12-2016 12:00:00
EventEndDate	31-12-2016 14:00:00
EventLocation	Comme Chez Soi, Boedapest, Aranykéz
EventPrice	40
Picture	Bestand kiezen Geen bestand gekozen

A user can edit his own event by clicking ”edit” at the bottom of the EventDetails-page. It will redirect the user to (www.zomomoo.azurewebsites/Event/EditEvent/id). This page needs an authentification request before opening, so only the host/creator of the event can access this page. All predefined fields will be automaticly filled in with the corresponding values. The user can modify these and save his changes. At the bottom, the user can find a ”Back to Homepage” link witch will return the user to the homepage.

6.4.2 EventControler

The existing event can be found in the database and be modified. After the changes are made, the user will be redirected to the modified event page (EventDetails).

6.5 Mobile

Bootstrap makes mobile websites clean. Just minimize your browser and see the results.