Event Tracker

Specificaties

Evert De Schryver

Schooljaar: 2018 – 2019 semester 2  
Opdrachtgever: Sea Shepherd Conservation Society

Inhoudstabel

[Analyse 3](#_Toc3368488)

[Probleemdomein 3](#_Toc3368489)

[Procesdiagram 4](#_Toc3368490)

[Use Case Diagram 5](#_Toc3368491)

[Use cases Overzicht 6](#_Toc3368492)

[User Interface Mockups 7](#_Toc3368493)

[Login Screen 7](#_Toc3368494)

[Forgot Password 8](#_Toc3368495)

[Reset password 9](#_Toc3368496)

[Change password 10](#_Toc3368497)

[View events 11](#_Toc3368498)

[Event Details 12](#_Toc3368499)

[Create/Edit Event 13](#_Toc3368500)

[Account Overview 14](#_Toc3368501)

[Create/Edit Account 15](#_Toc3368502)

[Storyboards 16](#_Toc3368503)

[Login Storyboard 16](#_Toc3368504)

[Forgot Password Storyboard 17](#_Toc3368505)

[Change Password 18](#_Toc3368506)

[Sidebar Nevigation 19](#_Toc3368507)

[Event Management 20](#_Toc3368508)

[accountmanagement 21](#_Toc3368509)

[Activity Diagrams 22](#_Toc3368510)

[Login 22](#_Toc3368511)

[Reset password 23](#_Toc3368512)

[Change password 24](#_Toc3368513)

[View upcoming events 25](#_Toc3368514)

[View my events 25](#_Toc3368515)

[View event details 26](#_Toc3368516)

[(Un)subscribe 27](#_Toc3368517)

[Create event 28](#_Toc3368518)

[Edit event 29](#_Toc3368519)

[Cancel event 30](#_Toc3368520)

[Delete event 31](#_Toc3368521)

[Remove participant 32](#_Toc3368522)

[Create account 33](#_Toc3368523)

[Edit account 34](#_Toc3368524)

[Delete account 35](#_Toc3368525)

[Ontwikkelingstack 36](#_Toc3368526)

# Analyse

## Probleemdomein

Event Tracker is een ‘responsive mobile first’ webapplicatie waarmee een gebruiker, afhankelijk van zijn niveau van autorisatie, evenementen kan creëren, beheren en verwijderen. Alle gebruikers kunnen zich voor een evenement inschrijven. Door zich in te schrijven, geven zij aan te zullen participeren in de organisatie van het evenement.

Een evenement wordt aangemaakt door een gebruiker met de daartoe voorziene autoriteit (een superuser) en beschikt over volgende zaken:

* Een naam
* Een start- en einddatum
* Een start- en einduur per dag
* Een locatie bestaande uit een stad en een provincie
* Een korte omschrijving
* Een bepaald aantal gezochte participanten
* Een lijst van reeds ingeschreven gebruikers

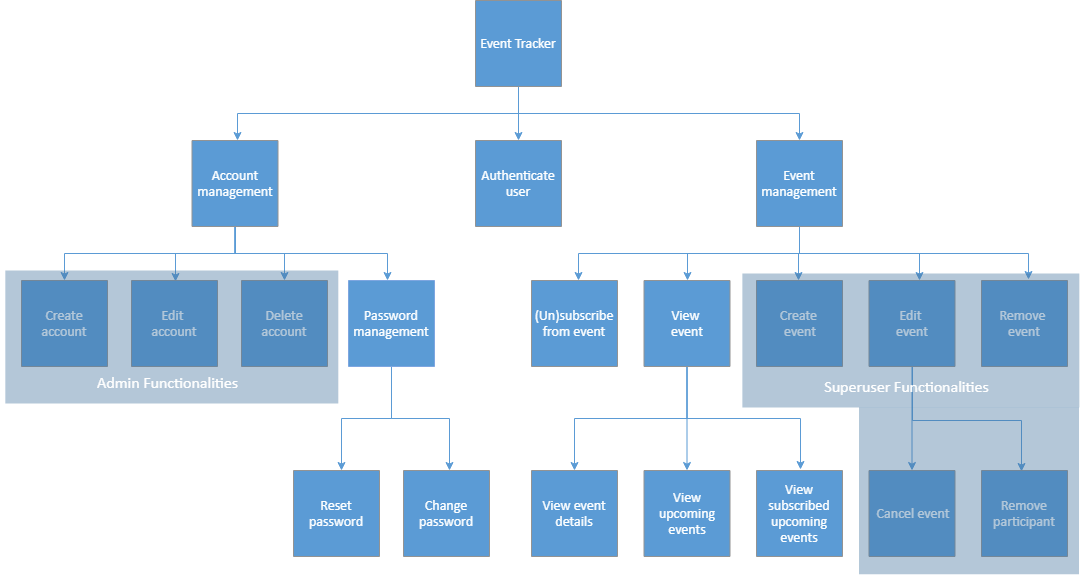
Alle gebruikers kunnen zich voor elk evenement in- of uitschrijven. Ook al is het gevraagd aantal participanten van een evenement bereikt, dan nog kan een gebruiker zich inschrijven voor het evenement. Er staat geen maximum op het aantal participanten. De applicatie zal wel visueel aanduiden wanneer het gevraagde participantenaantal is bereikt en wanneer niet. Een evenement kan door een superuser geannuleerd worden. Ook dan zal de applicatie visueel aanduiden dat het evenement geannuleerd is. Een gebruiker kan zich in dit geval niet meer inschrijven voor het evenement. Ook kan een superuser de details van een geannuleerd evenement niet langer aanpassen. Een geannuleerd evenement kan door een superuser terug beschikbaar worden gesteld.

Gebruikers moeten zich inloggen alvorens ze gebruik kunnen maken van de applicatie. Zij kunnen zelf geen gebruikersprofiel aanmaken. Dit gebeurt door een administrator. Een gebruikersprofiel beschikt over een gebruikersnaam, paswoord, emailadres en gebruikersrol. Een gebruiker van de applicatie kan één van volgende rollen bekleden: basic user, superuser of administrator.

* Basic user:
  + Zij kunnen zich inloggen
  + Zij kunnen een overzicht bekijken van alle toekomstige evenementen
  + Zij kunnen een overzicht bekijken voor alle toekomstige evenementen waarop zij zich hebben ingeschreven
  + Zij kunnen de details van evenementen bekijken
  + Zij kunnen zich in- of uitschrijven voor evenementen
  + Zij kunnen het paswoord gelinkt aan hun profiel wijzigen of resetten
* Superuser:
  + Zij kunnen alles wat een basic user kan
  + Zij kunnen evenementen creëren, aanpassen en verwijderen
  + Zij kunnen voor elk evenement deelnemers uitschrijven
  + Zij kunnen evenementen annuleren
* Administrator:
  + Zij kunnen alles wat superusers kunnen
  + Zij kunnen gebruikersprofielen creëren, aanpassen en verwijderen

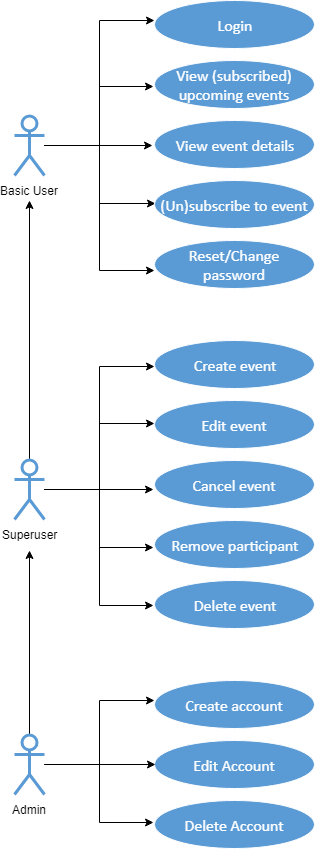
## Procesdiagram

Onderstaand diagram schets een beeld van alle processen die plaats vinden binnen de context van de applicatie.



## Use Case Diagram

Onderstaand diagram schets een beeld van alle use cases die plaats vinden binnen de context van de applicatie:



## Use cases Overzicht

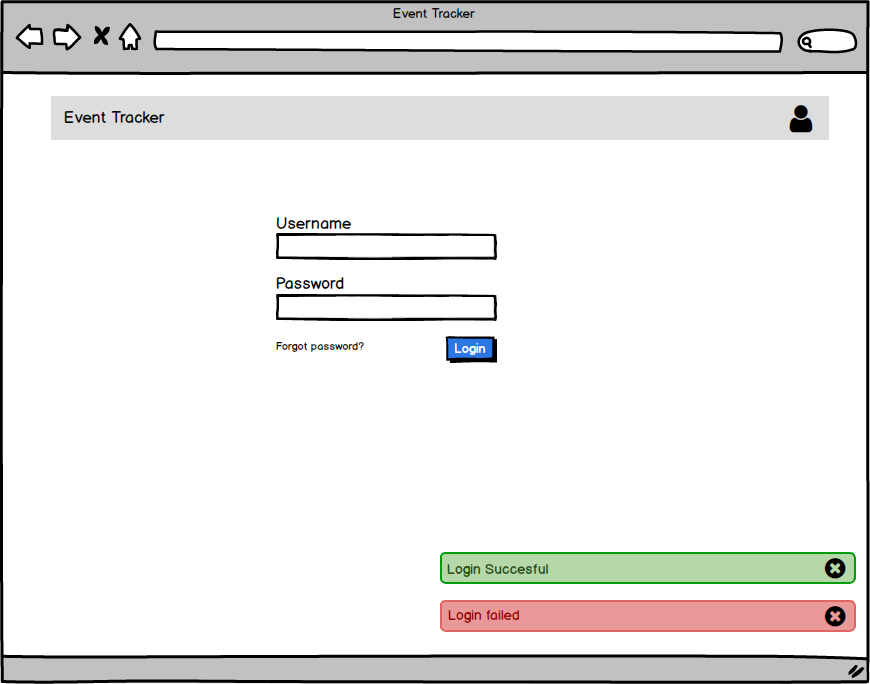
Onderstaand overzicht schets een beeld van alle use cases die plaats vinden binnen de context van de applicatie.

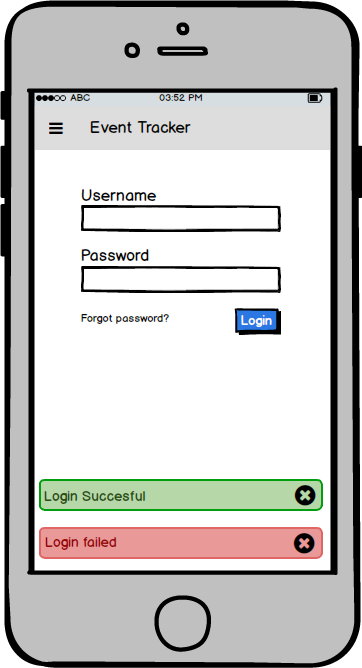
|  |  |  |
| --- | --- | --- |
| Als een… | Zou ik graag… | Zodat ik… |
| Basic user | Inloggen | Gebruik kan maken van de applicatie |
| Basic user | Een lijst krijgen van alle toekomstige evenementen | Kan zien welke evenementen er gepland staan |
| Basic user | Een lijst krijgen van alle toekomstige evenementen waarvoor ik me heb ingeschreven | Een overzicht heb van alle toekomstige evenementen waar mijn aanwezigheid wordt verwacht |
| Basic user | De details van een evenement bekijken | Alle nodige info heb omtrent het evenement |
| Basic user | Mezelf inschrijven voor een evenement | Mijn aanwezigheid kenbaar kan maken |
| Basic user | Mezelf uitschrijven voor een evenement | Mijn afwezigheid kenbaar kan maken |
| Basic user | Mijn paswoord wijzigen | Goede beveiligingsnormen kan hanteren |
| Basic user | Mijn paswoord resetten | Een nieuw paswoord kan instellen indien ik het vergeten ben |
| Superuser | Evenementen aanmaken | Nieuwe evenementen beschikbaar kan stellen voor gebruikers |
| Superuser | Evenementen aanpassen | De praktische info van een evenement kan wijzigen |
| Superuser | Evenementen verwijderen | Onnodige evenementen kan verwijderen |
| Superuser | Evenementen annuleren | Gebruikers op de hoogte kan stellen wanneer een evenement niet doorgaat |
| Admin | Gebruikersprofielen aanmaken | Aan gebruikers de mogelijkheid geef gebruik te maken van de applicatie |
| Admin | Gebruikersprofielen verwijderen | Bepaalde gebruikers de toegang tot de applicatie kan ontzeggen |
| Admin | Gebruikersprofielen aanpassen | De gebruikersrol, email of naam van een gebruiker kan wijzigen |

## User Interface Mockups

##### Login Screen

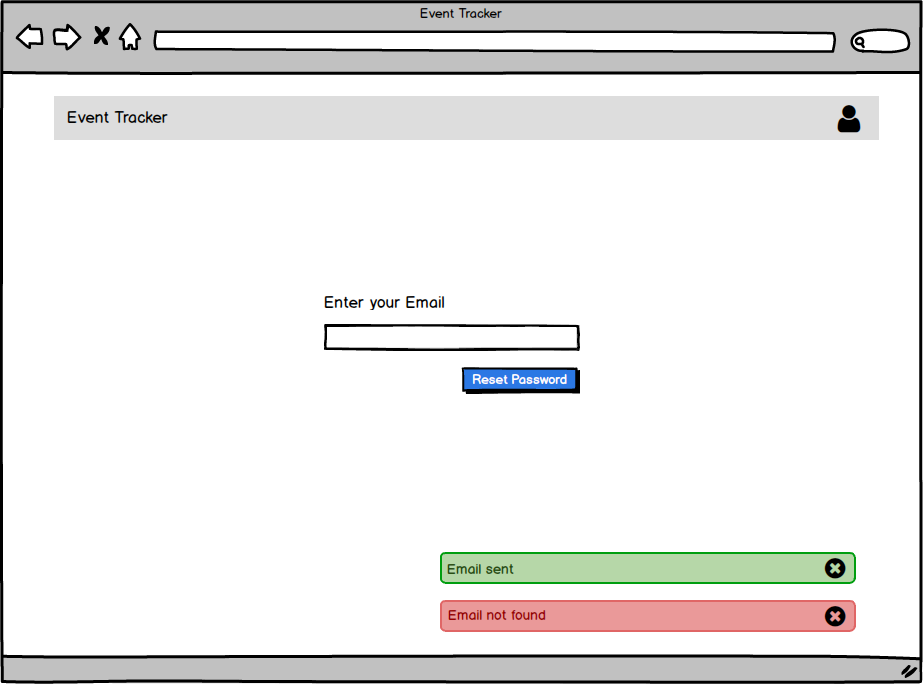
Het login scherm is beschikbaar voor alle gebruikers.





##### Forgot Password

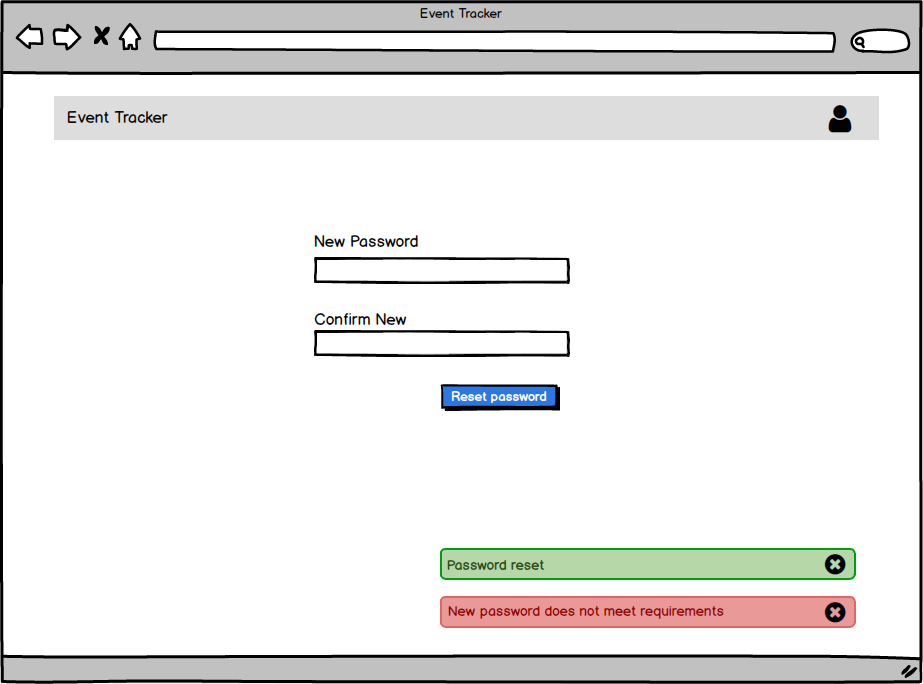
Het ‘Forgot password’ scherm is beschikbaar voor alle gebruikers.

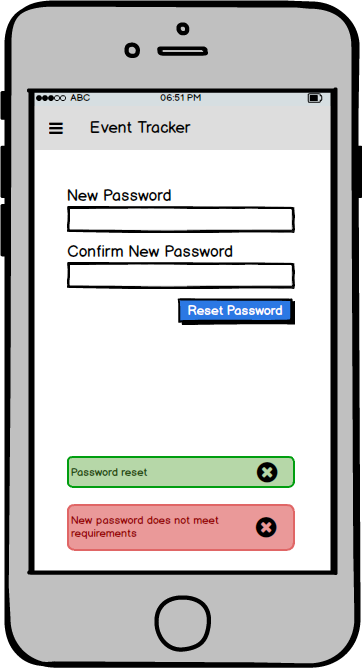




##### Reset password

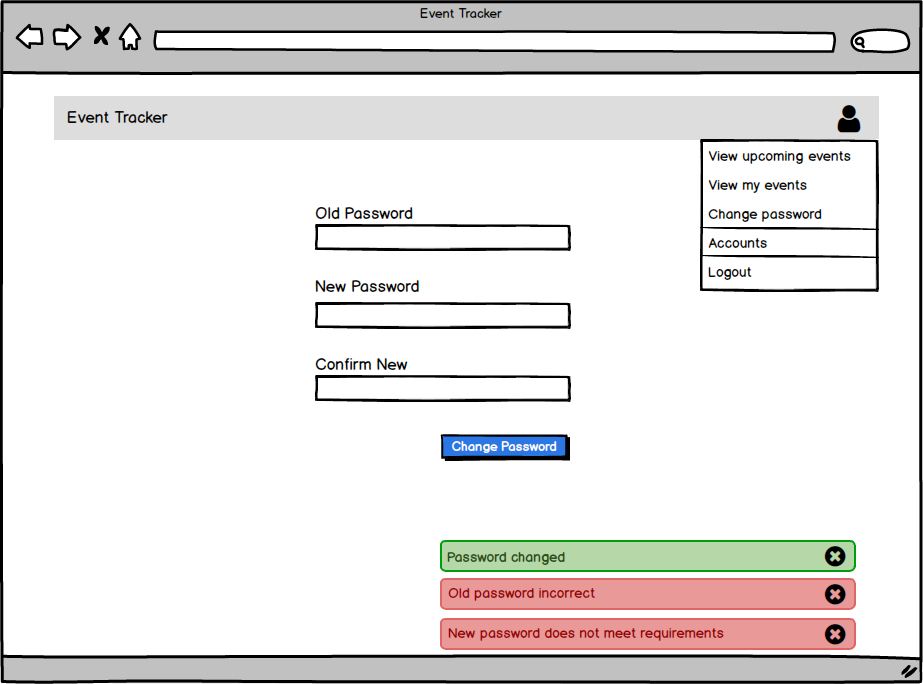
Het reset password scherm is toegankelijk voor alle gebruikers. Dit scherm kan enkel bereikt worden via een door de applicatie gegenereerde URL die per mail naar de user wordt verstuurd nadat die een geldige email heeft ingevoerd in het ‘Forgot Password’ scherm.

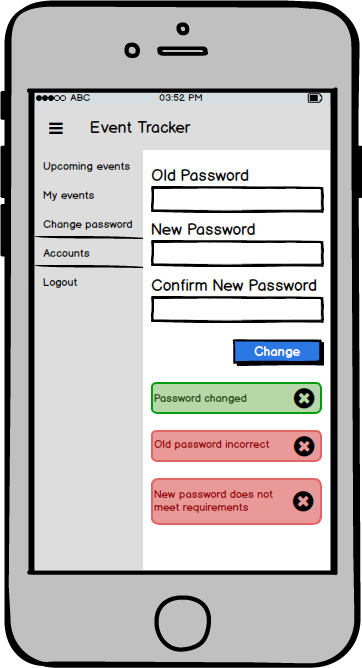




##### Change password

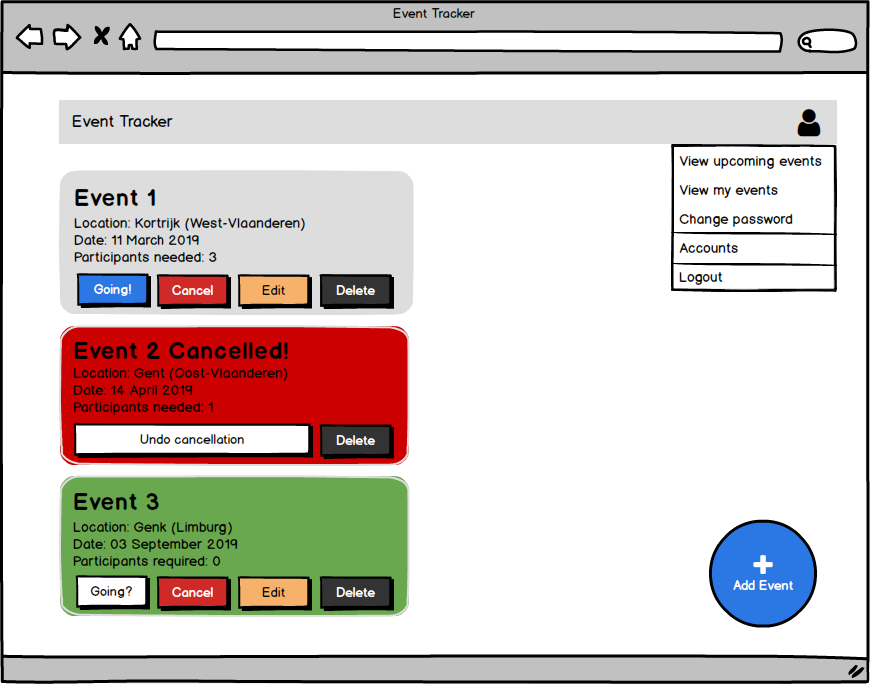
* Het ‘Change password’ scherm is beschikbaar voor alle gebruikers.
* De link ‘Accounts’ in het dropdown menu is enkel zichtbaar voor administrators.

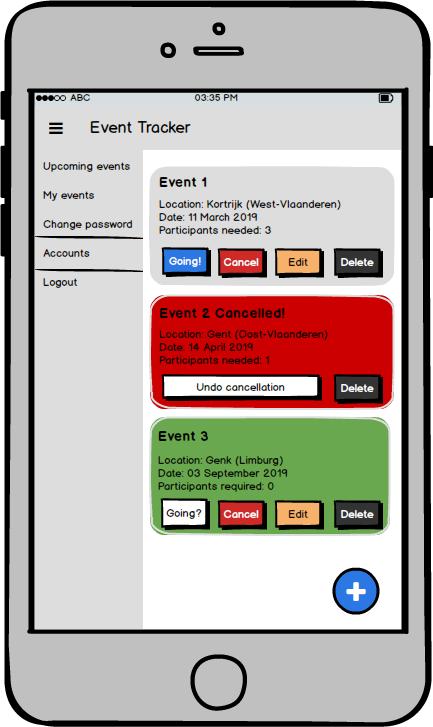




##### View events

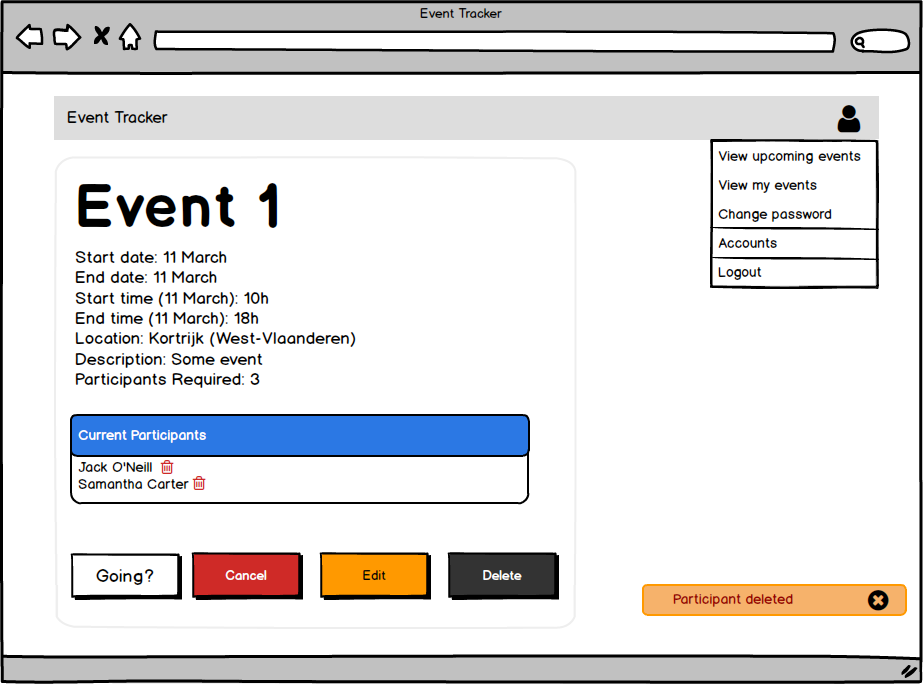
* Het ‘View Events’ scherm is beschikbaar voor alle gebruikers.
* De link ‘Accounts’ in het dropdown menu is enkel zichtbaar voor administrators.
* Deze lay-out wordt gehanteerd bij zowel de ‘view upcoming events’ als bij de ‘My events’ actie.
* De buttons ‘Add event’, ‘Cancel’, ‘Edit’ & ‘Undo cancellation’ zijn enkel zichtbaar voor superusers & administrators

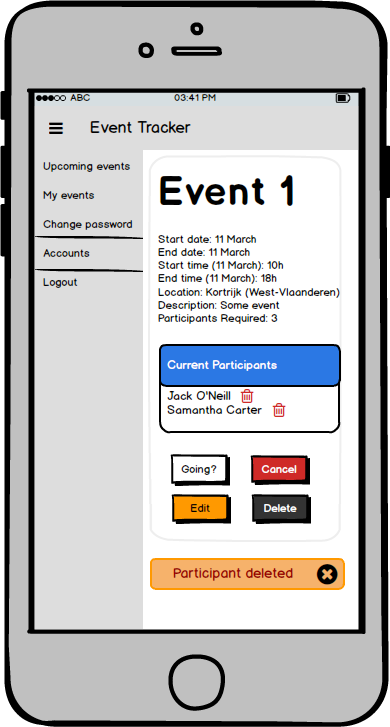




##### Event Details

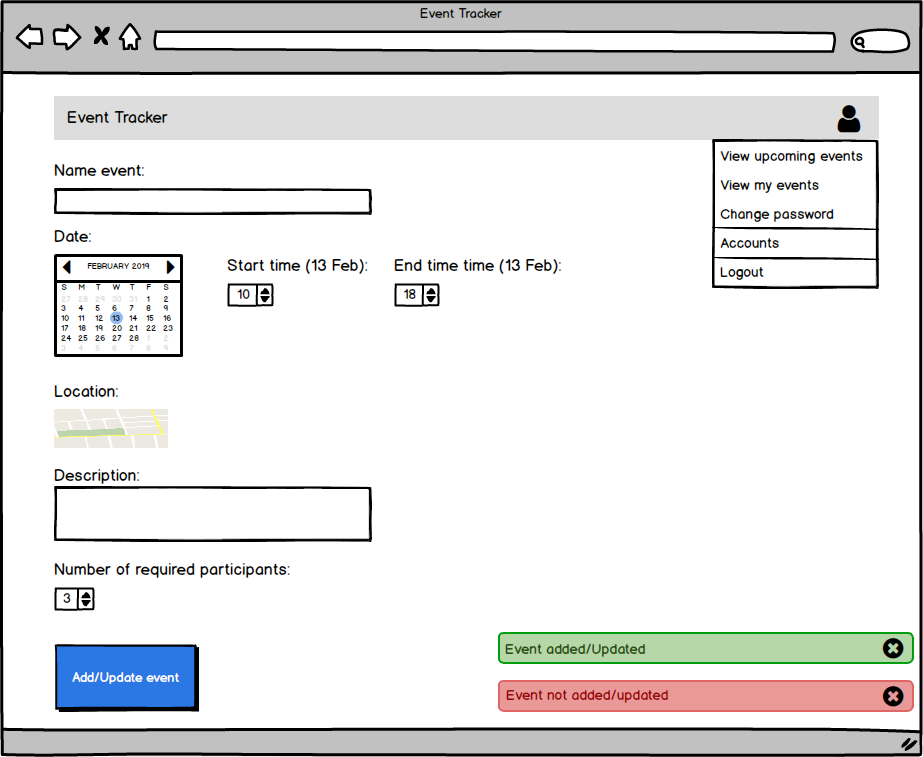
* Het ‘Event Details’ scherm is beschikbaar voor alle gebruikers.
* De link ‘Accounts’ in het dropdown menu is enkel zichtbaar voor administrators.
* De buttons ‘Cancel’, ‘Edit’ en de prullenmand iconen zijn enkel beschikbaar voor superusers en administrators

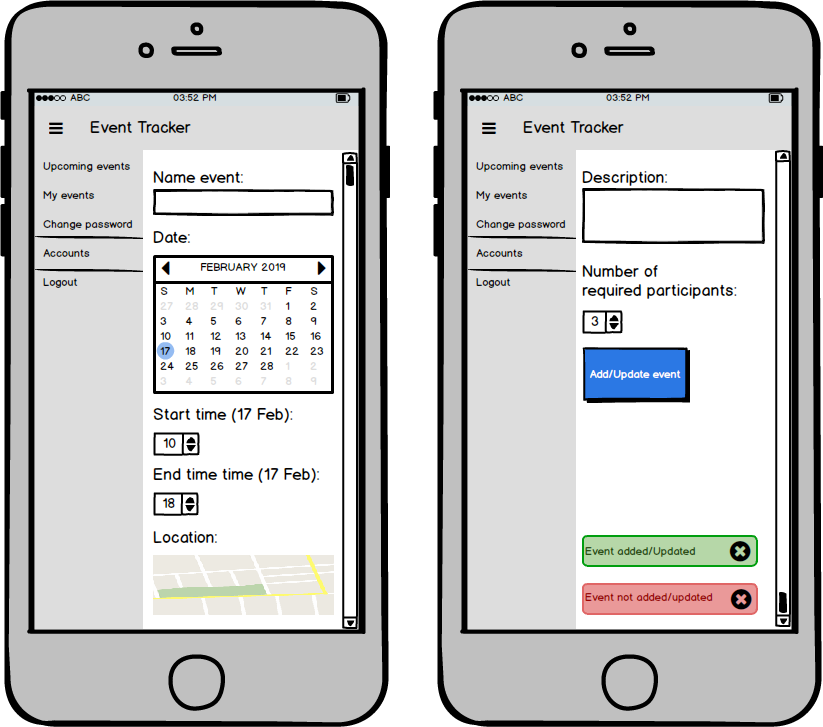




##### Create/Edit Event

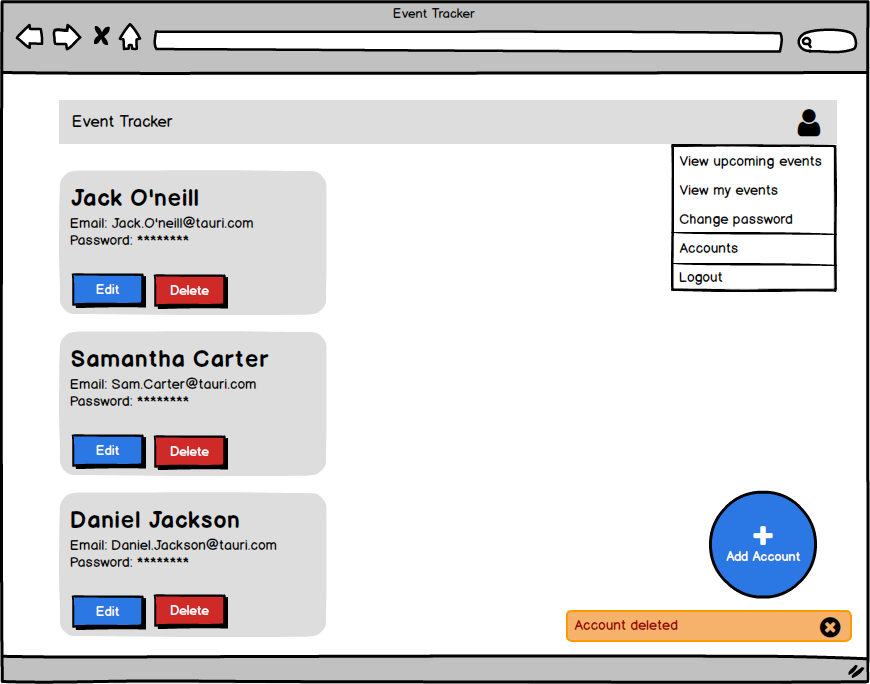
* Het ‘Create/Edit Event’ scherm is enkel beschikbaar voor superusers en administrators.
* De link ‘Accounts’ in het dropdown menu is enkel zichtbaar voor administrators.
* De location input linkt naar de Google Maps API.
* In de Date Picker kunnen meerdere opeenvolgende data worden aangeduid.
* Per aangeduide datum zal er een instantie van de ‘Start time’ en ‘End time’ controls verschijnen.
* Bij een nieuw evenement zijn alle inputvelden ingevuld met hun default waardes.
* Bij het opdateren van een evenement zijn alle inputvelden ingevuld met de waarden die op dat moment waren ingesteld voor het evenement in kwestie.

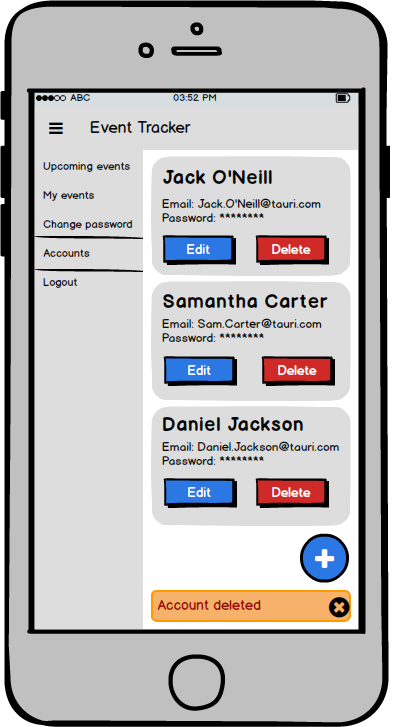




##### Account Overview

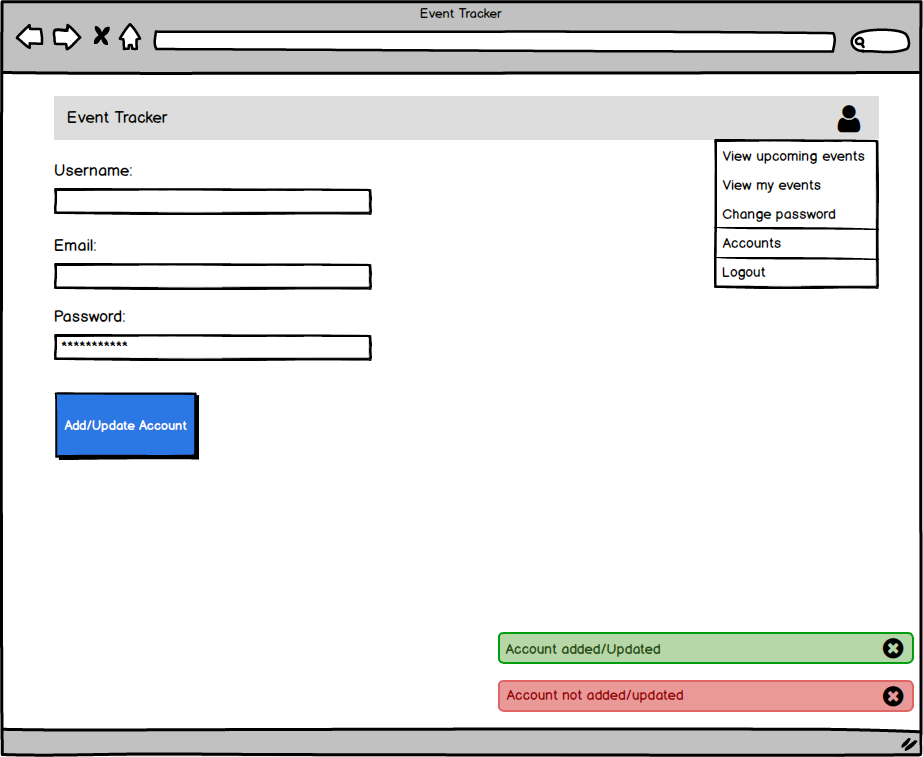
* Het ‘Account Overview’ scherm is enkel beschikbaar voor administrators.
* De link ‘Accounts’ in het dropdown menu is enkel zichtbaar voor administrators.

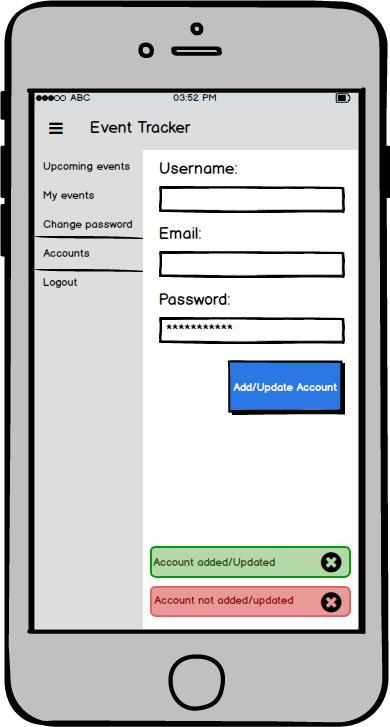




##### Create/Edit Account

* Het ‘Create/Edit Account’ scherm is enkel beschikbaar voor administrators.
* Bij een nieuwe account zijn alle inputvelden ingevuld met hun default waardes.
* Bij het opdateren van een account zijn alle inputvelden ingevuld met de waarden die op dat moment waren ingesteld voor het evenement in kwestie.

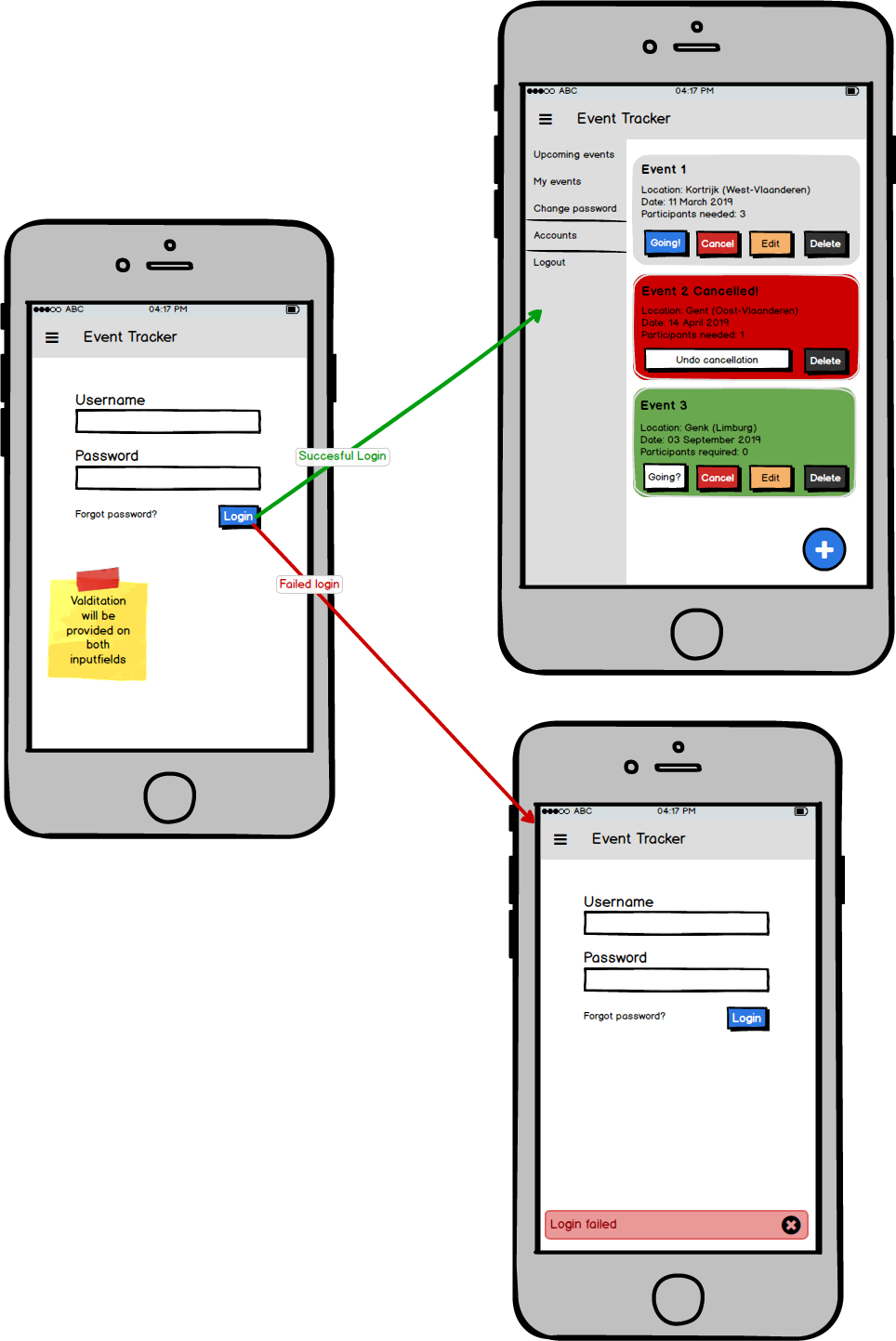




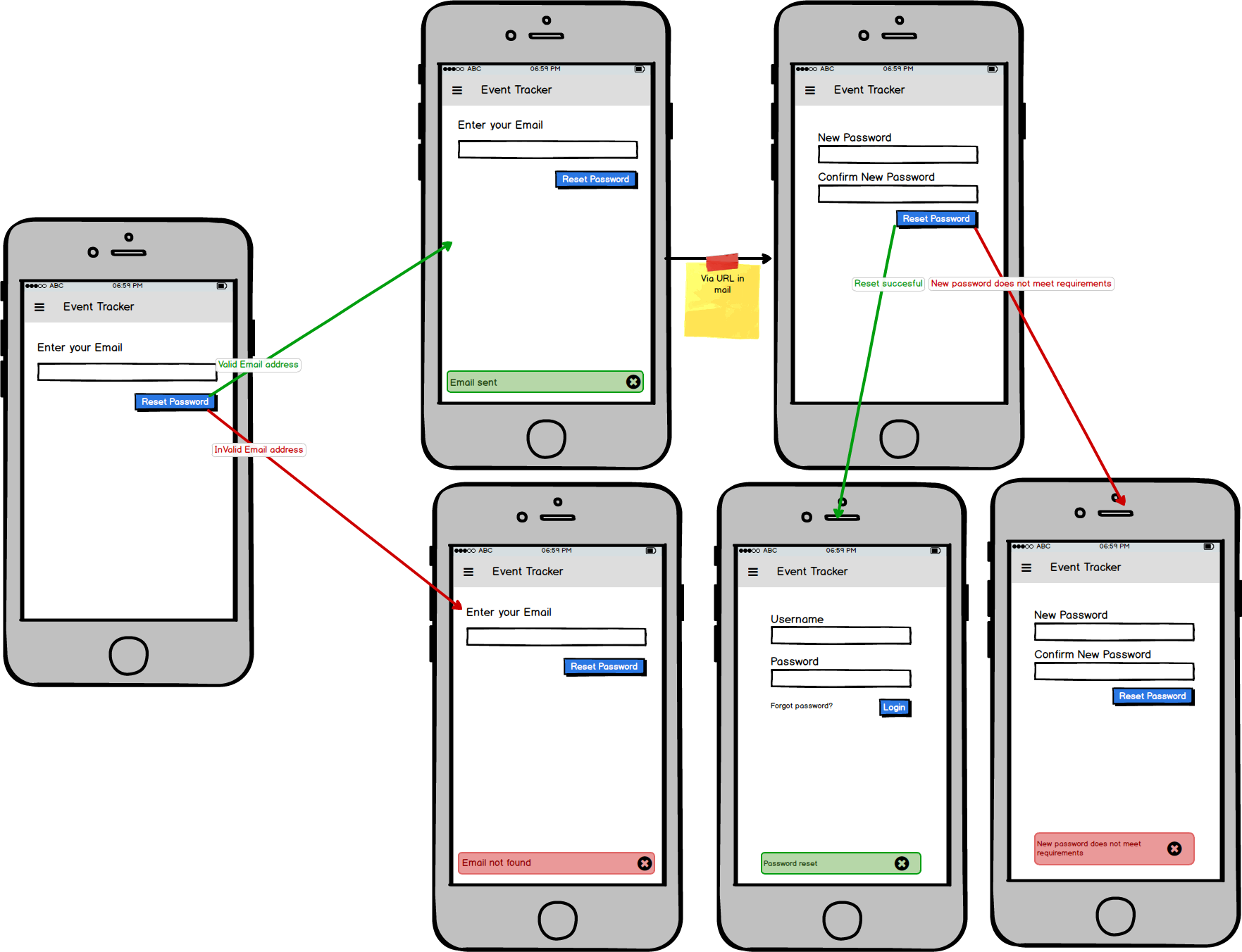
## Storyboards

Op vlak van navigatie is er geen wezenlijk verschil tussen de mobiele UIs en desktop UIs. Om het simpel te houden, wordt bijgevolg enkel de mobiele UI gebruikt bij onderstaande storyboards. Tevens bevatten alle UIs in onderstaande storyboards alle functionaliteiten. Sommige controls of UIs zijn dus niet toegangkelijk voor alle gebruikers. Dit is waar nodig gespecifieerd.

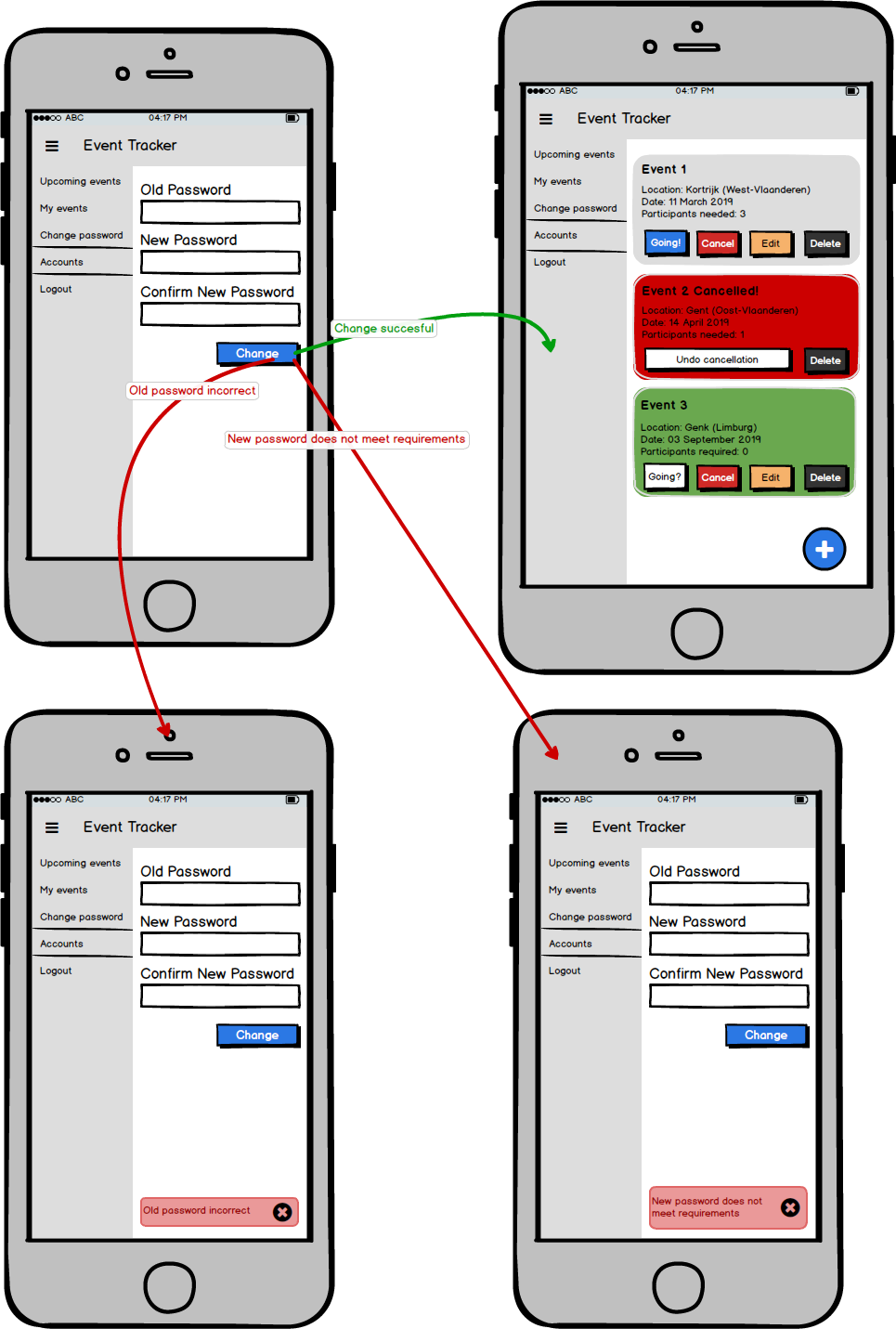
##### Login Storyboard



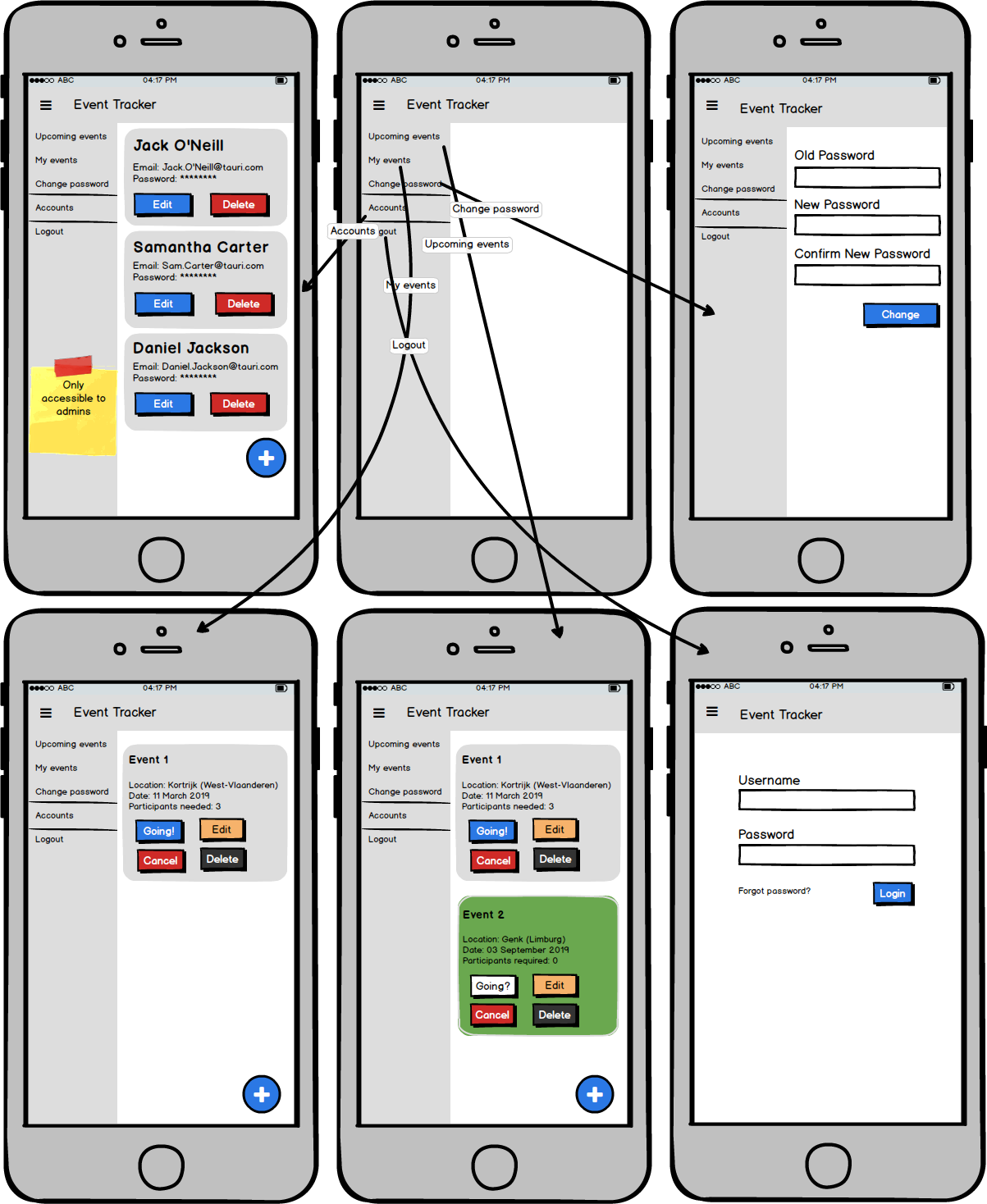
##### Forgot Password Storyboard



##### Change Password

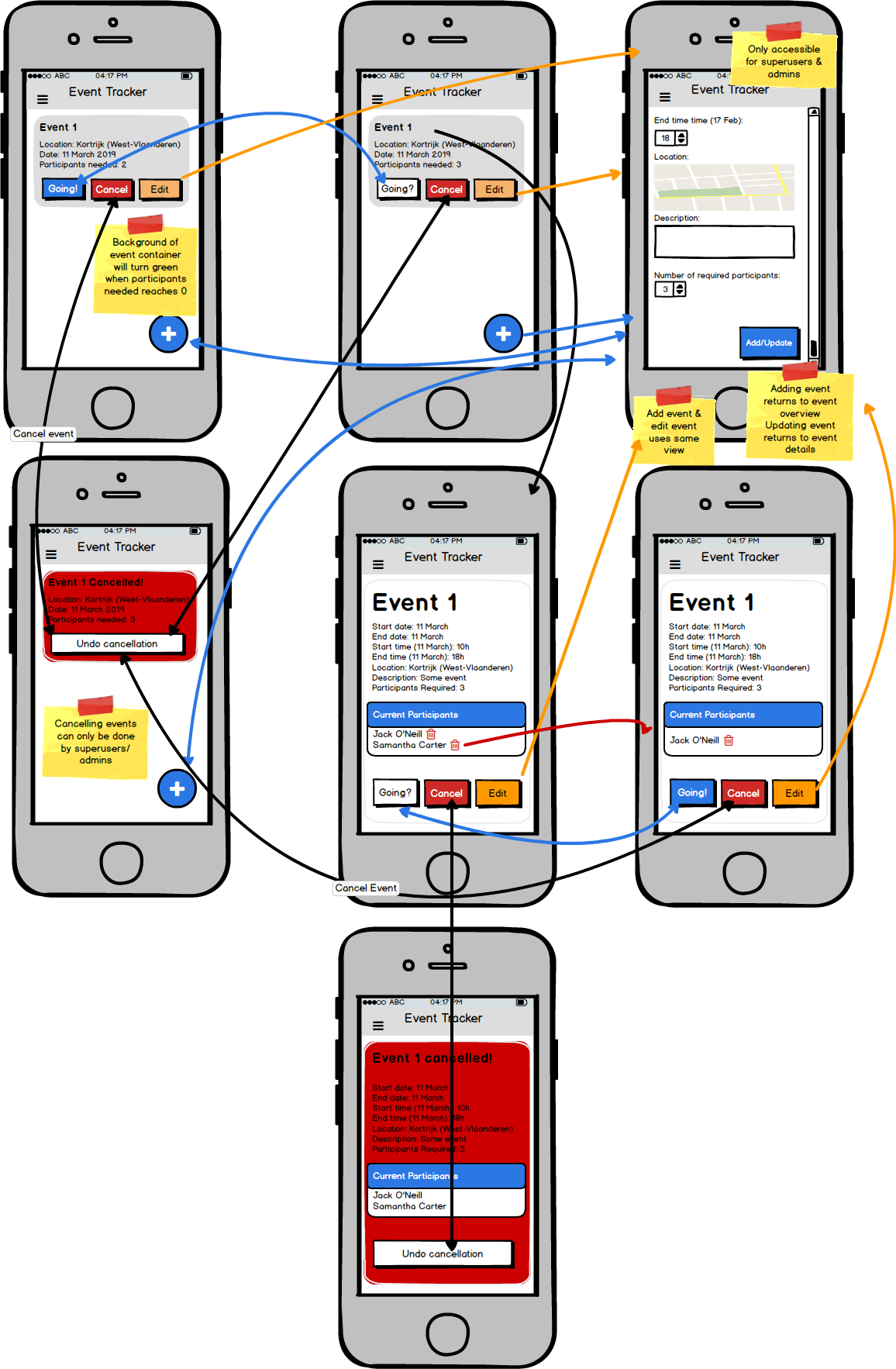


##### Sidebar Nevigation

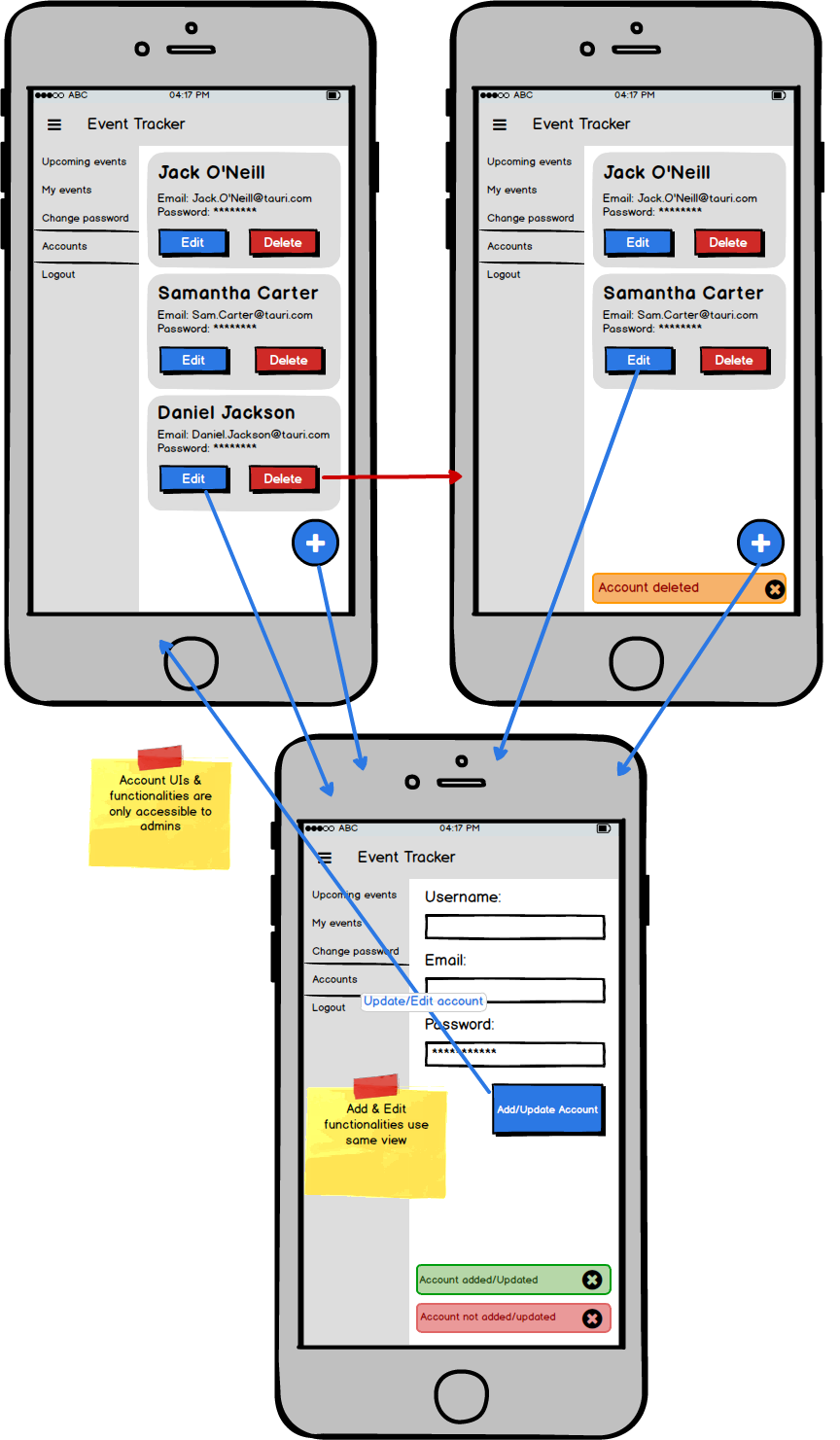


##### Event Management

Gezien onderstaand storyboard reeds veel interacties kent, is de ‘Delete event’ functionaliteit hier niet opgenomen. De gebruiker wordt na het verwijderen van een evenement simpelweg teruggestuurd naar een opgedateerd overzicht van de nog resterende evenementen.

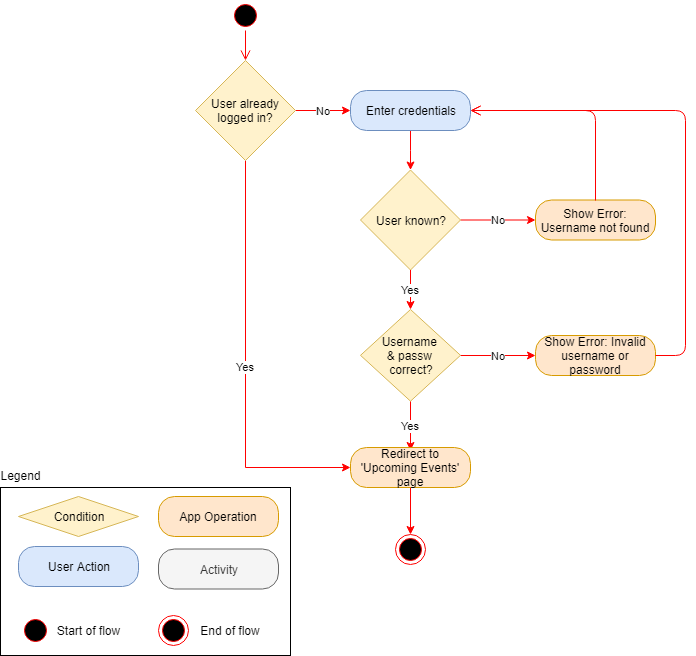


##### accountmanagement

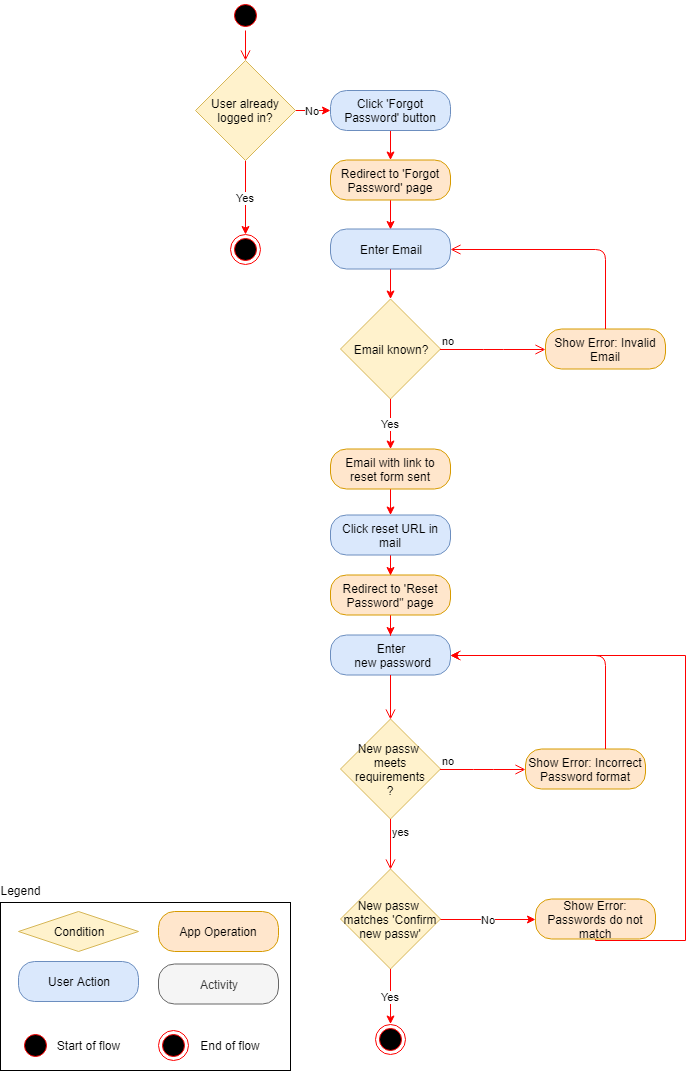


## Activity Diagrams

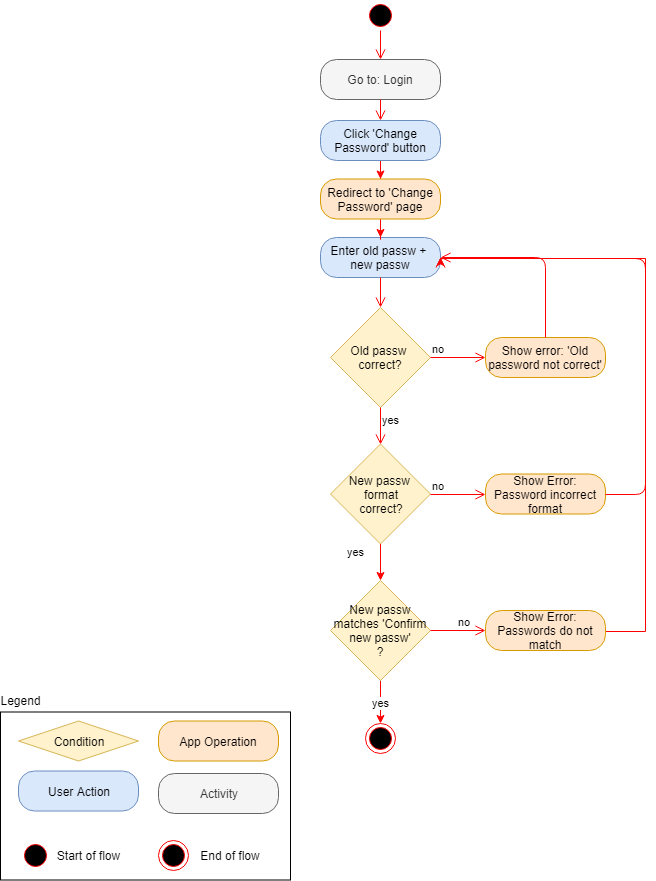
##### Login



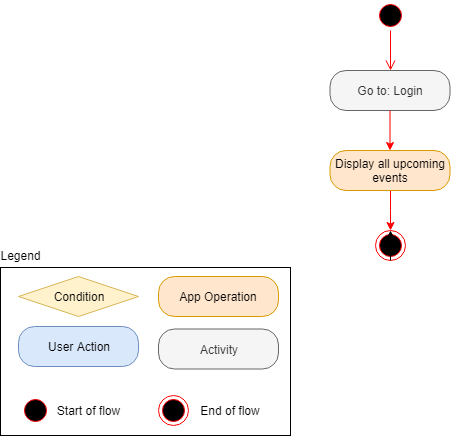
##### Reset password



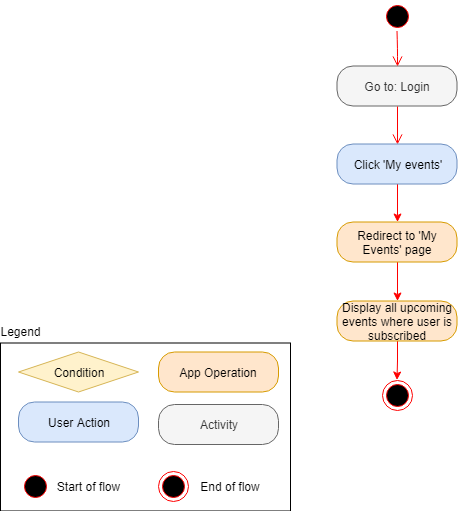
##### Change password



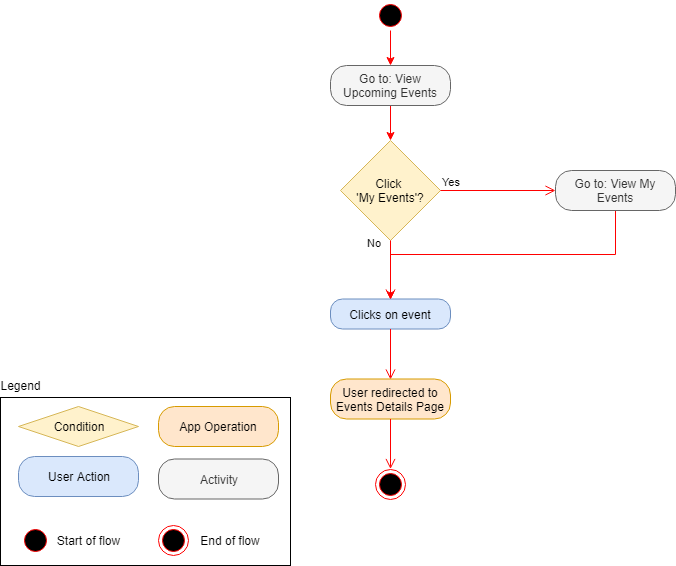
##### View upcoming events



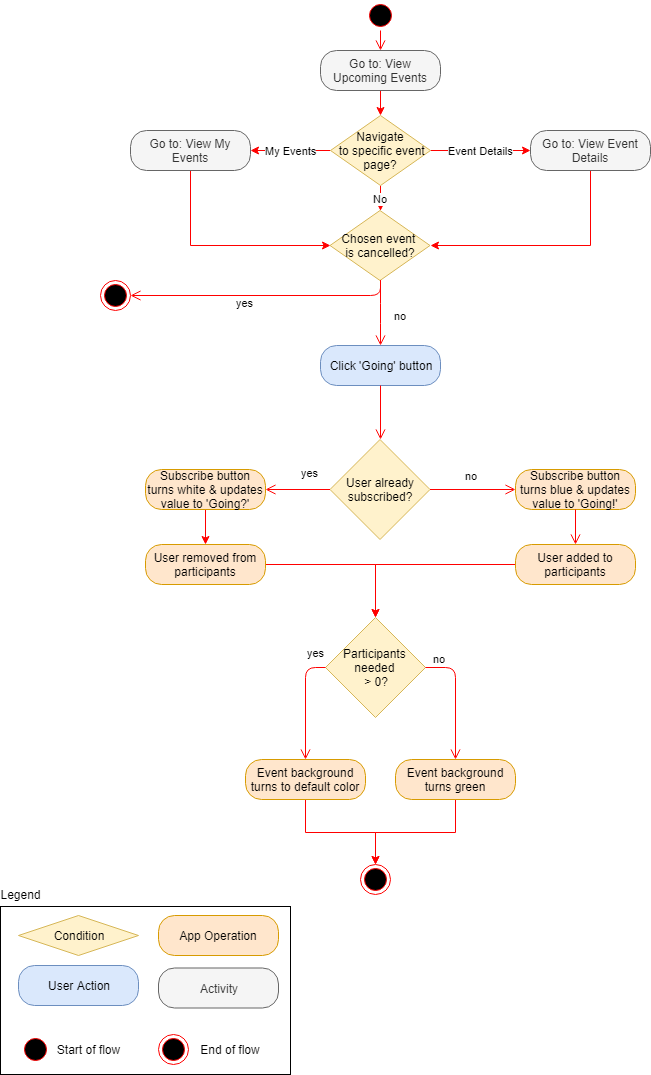
##### View my events



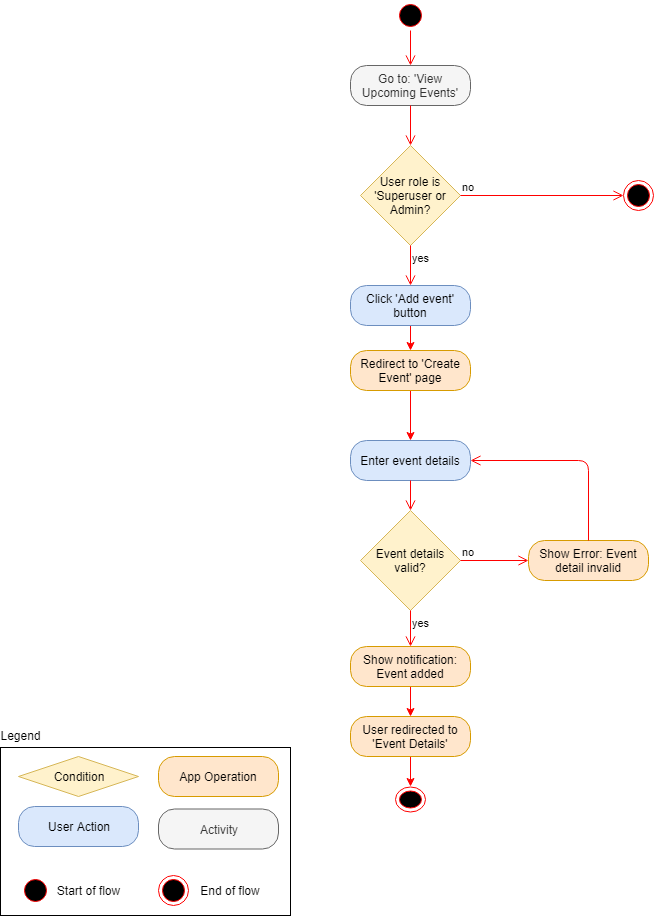
##### View event details



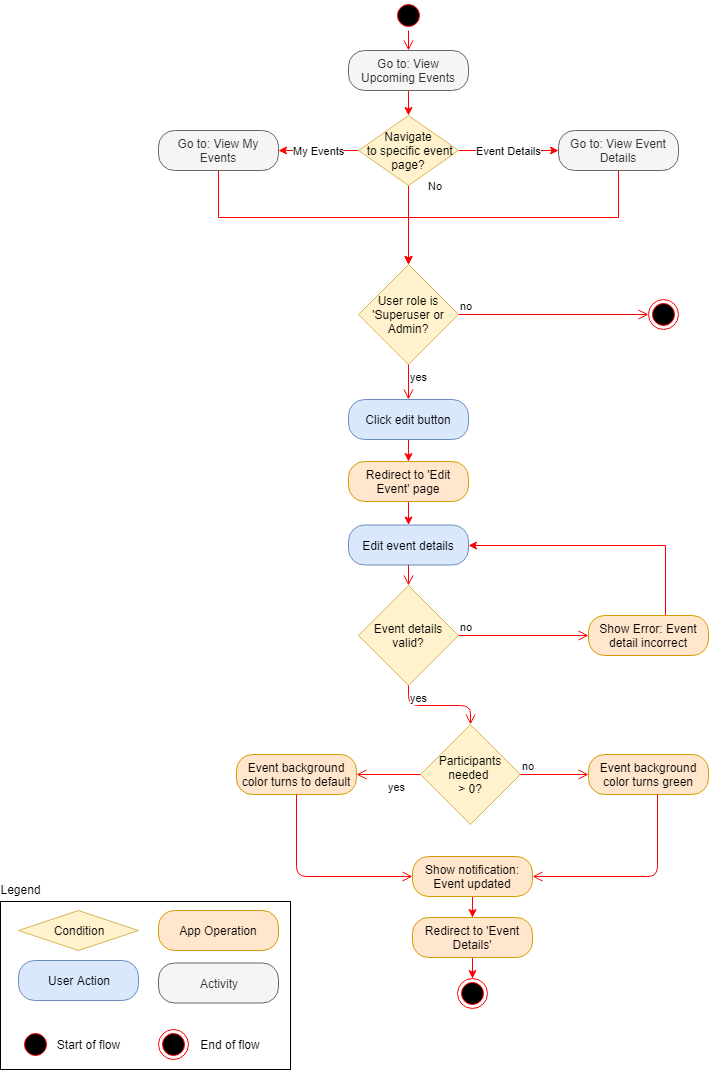
##### (Un)subscribe



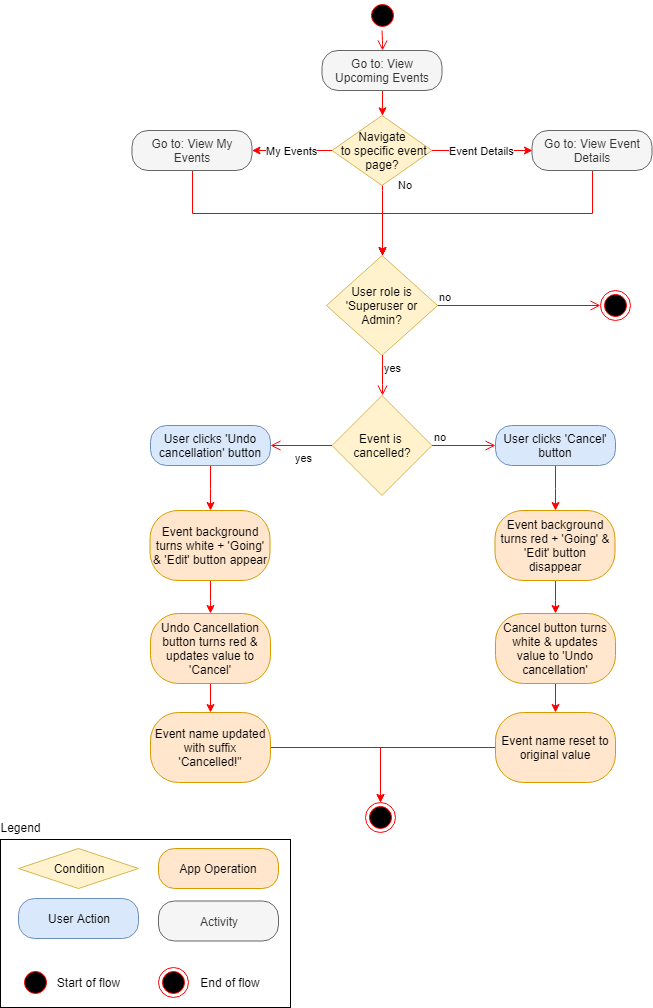
##### Create event



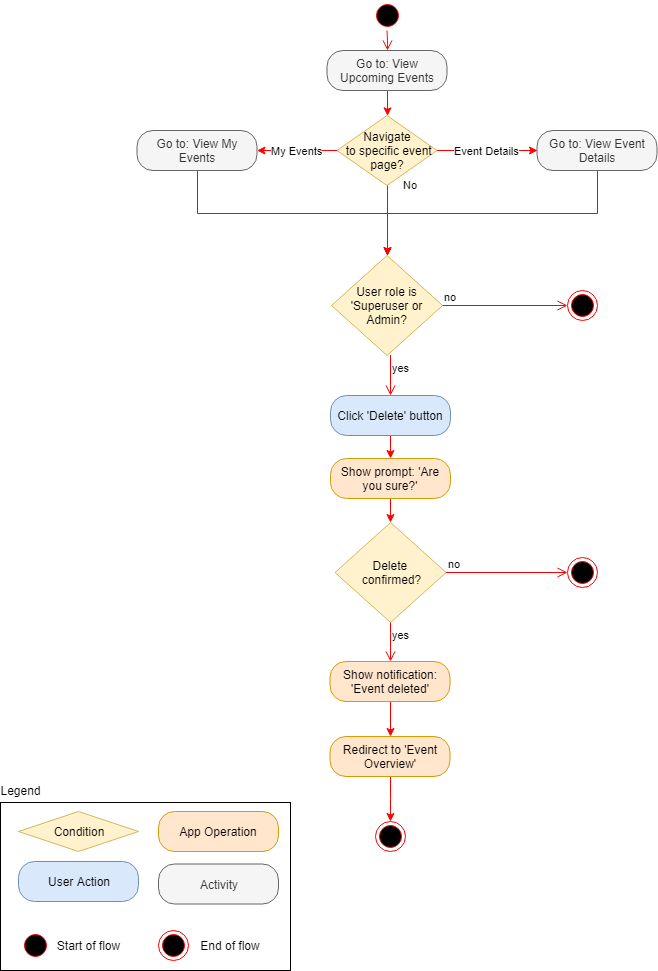
##### Edit event



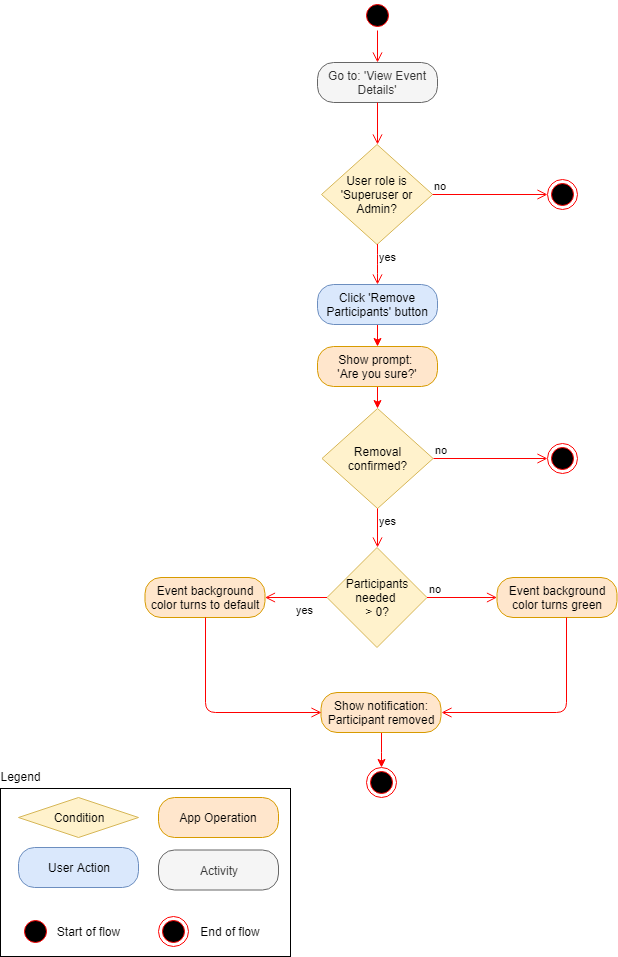
##### Cancel event



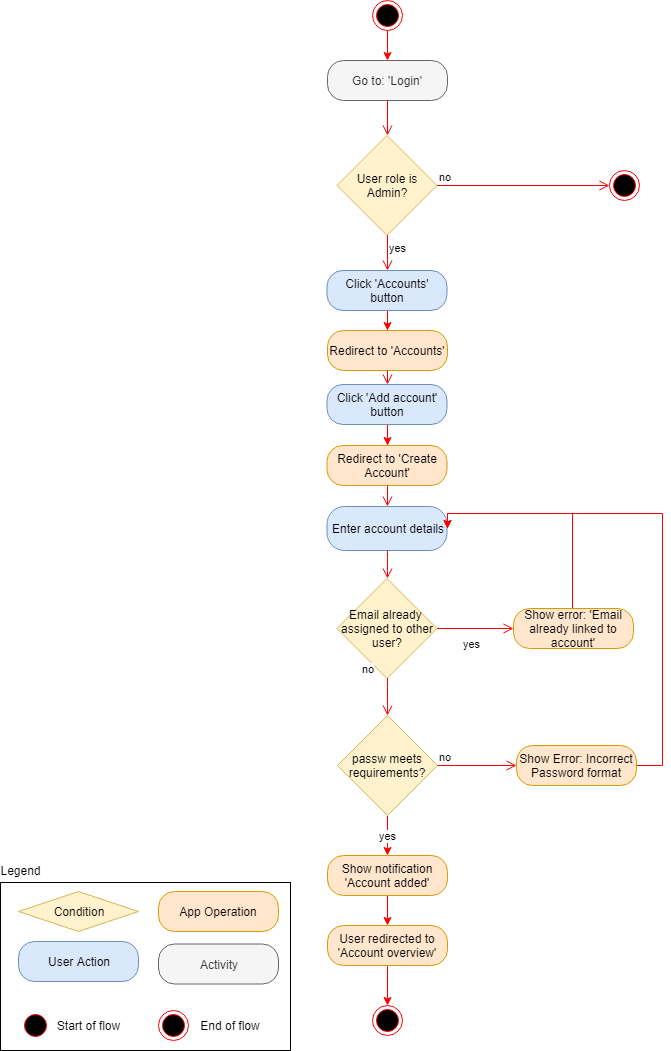
##### Delete event



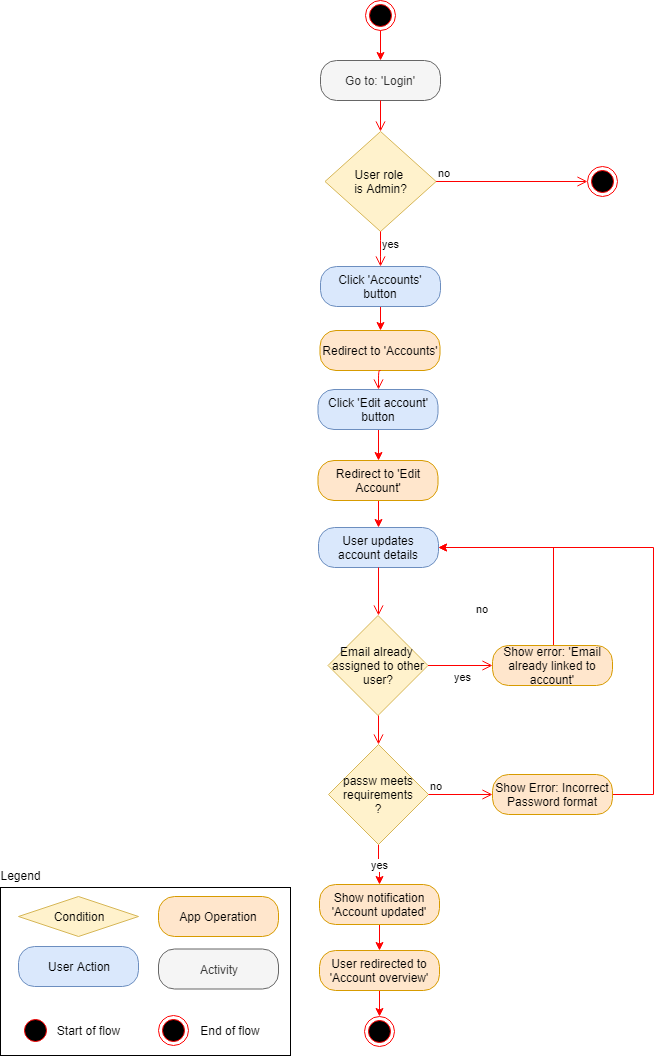
##### Remove participant



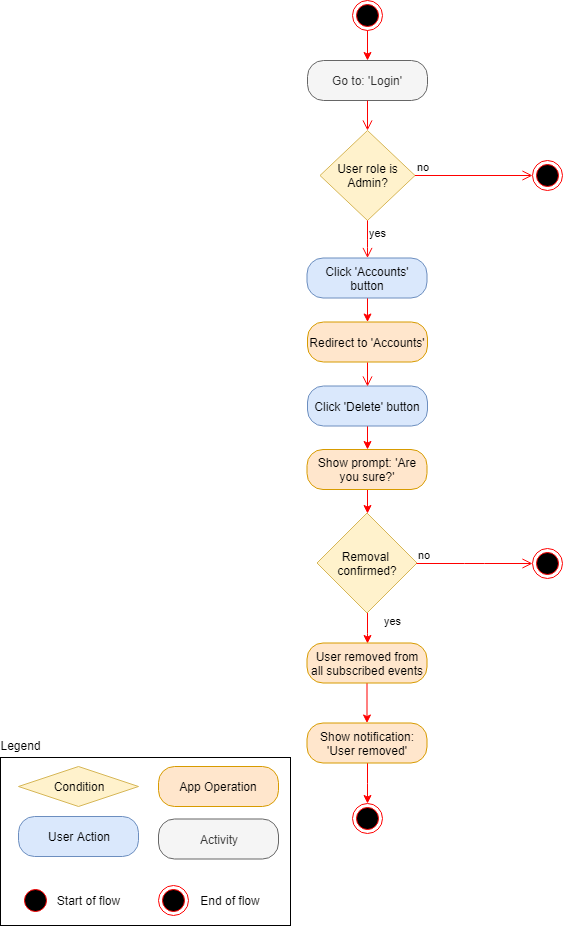
##### Create account



##### Edit account



##### Delete account

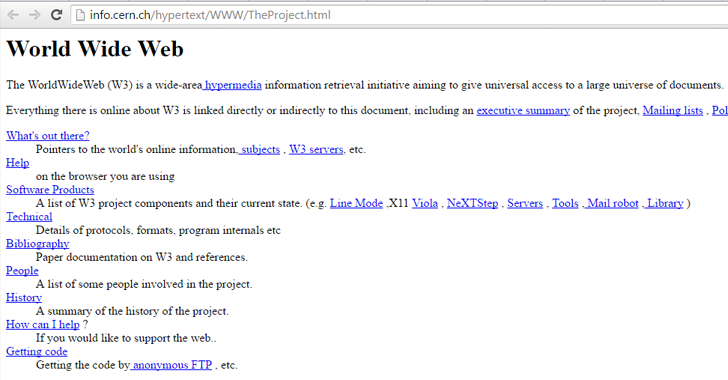


## Ontwikkelingstack

* Doelplatform: Cross Platform
  + OS & Device independent
  + Doel is een zo breed mogelijk publiek aan te spreken met een zo beperkt mogelijke code base
  + Focus op mobile-friendly user experience door gebruik van Bootstrap 4 (Mobile First)
* Technologieën:
  + Front end:
    - HTML 5
    - CSS 3
    - SaSS (Nieuw: niet gezien tijdens opleiding)
    - JavaScript
    - Jquery
    - Bootstrap 4 (Nieuw: mobile first)
  + Back end:
    - ASP.NET Core MVC (Nieuw: Core)
    - ASP.NET Core Identity (Nieuw: Core)
    - Entity Framework Core (Nieuw: Core)
    - Google Maps API
* Tools
  + IDE: MS Visual Studio 2017 Community Edition
  + UI Mockups & Storyboards: Balsamiq Mockups
  + Diagrams: Draw.io

## Technologische uiteenzetting: Mobile First

Op de dag van dit schrijven, vierde het wereldwijde web zijn 30e verjaardag. Bij zijn ontstaan werden we verwelkomd in een wereld met ogenschijnlijk geen limiet op zijn eigen potentieel. Maar de eerste webpagina’s zagen er niet bepaald sexy uit.



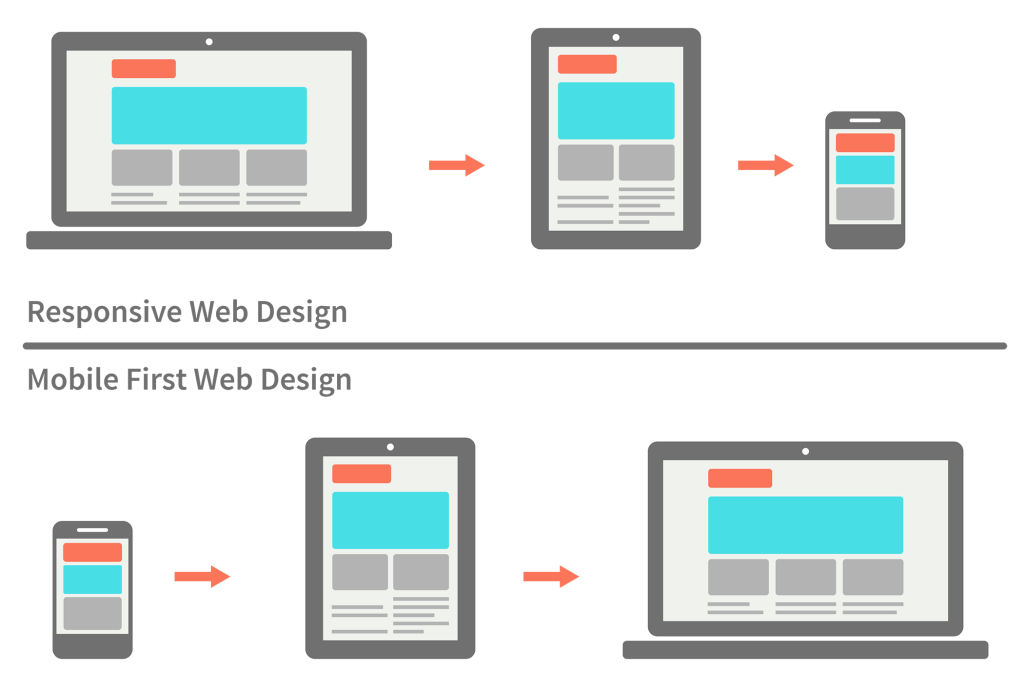
De vraag naar gebruiksvriendelijke en mooie websites kwam er toen steeds meer mensen hun weg vonden naar het wereldwijde web. Dit creëerde nieuwe uitdagingen. Naast de inhoud werd nu ook de lay-out belangrijk. En er veranderde nog meer. De manier waarop we vandaag websites bezoeken, is niet dezelfde als 30 jaar geleden. Waar websites vroeger voornamelijk via desktops of laptops werden bezocht, zien we vandaag hoe smartphones en tablets deze rol overnemen.

Als antwoord op deze trends, is men via heel wat tussenhaltes gekomen tot ‘responsive webdesign’. Websites oogden nu fris en modern en waren vlot toegankelijk vanop ieder device. Maar er was ruimte voor verbetering. Men ging er namelijk van uit dat websites vooral via een device met groot scherm werden bezocht. De lay-out van een webpagina werd bijgevolg berekend op de dimensies van een groot scherm. Pas als de website zou detecteren dat ze werd geraadpleegd via een klein scherm, zou de lay-out worden herberekend op basis van de dimensies van dit scherm.

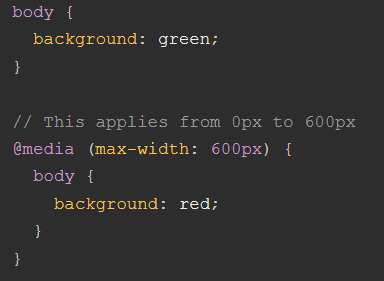
Een jammere zaak omwille van twee redenen:

1. Door de opkomst van smartphones, worden steeds meer websites bezocht door middel van devices met klein scherm.
2. Smartphones of tablets hebben doorgaans minder rekenkracht als desktops of laptops.

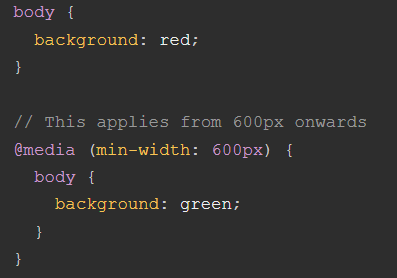
Een lay-out voor een groot scherm wordt dus vaak onnodig berekend. Als antwoord hierop, ontstond de ‘Mobile First’ filosofie waarbij devices met kleiner scherm net bevoordeeld worden. De lay-out wordt in eerste instantie berekend op kleinere schermen. Pas als de website detecteert dat een device met groot scherm de website raadpleegt, wordt de lay-out herberekend naar een groter formaat. Hierbij kan gebruik gemaakt worden van de doorgaans grotere rekenkracht van een device met groot scherm. Men draait de boel dus om.



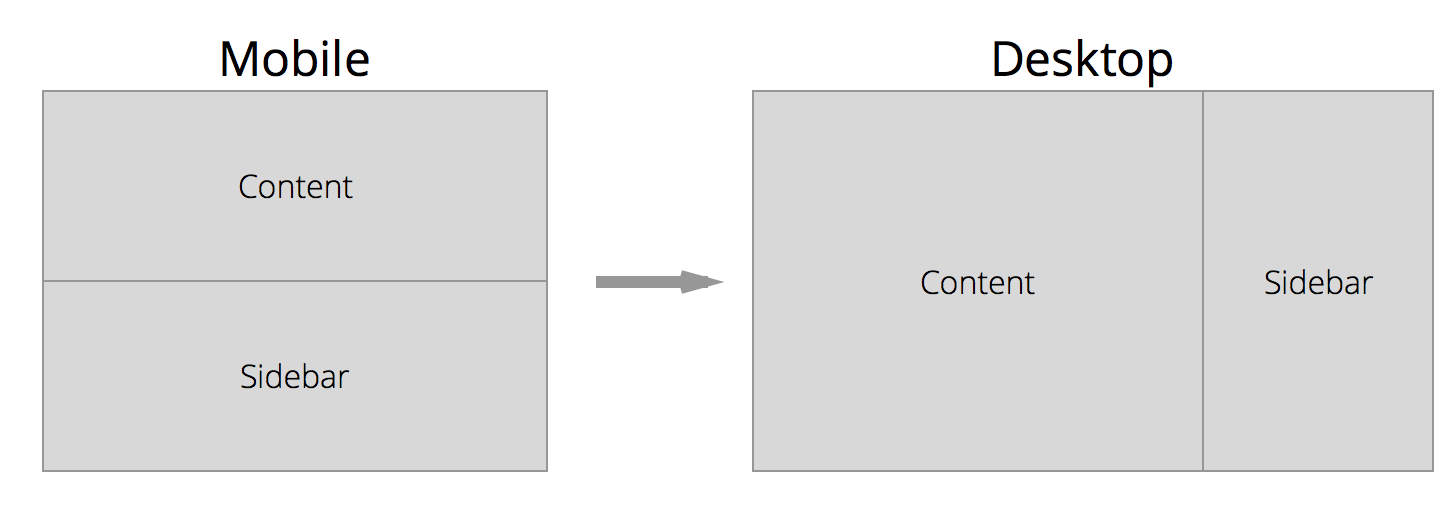
Als we vervolgens in de technische details duiken, moeten we het eerst hebben over media queries. Aan de hand van een media query wordt bepaald welke CSS gebruikt wordt afhankelijk van de dimensies van het scherm. Zo zal de achtergrond van een website door onderstaande code rood kleuren indien deze bezocht wordt door een scherm dat smaller is dan 600px.



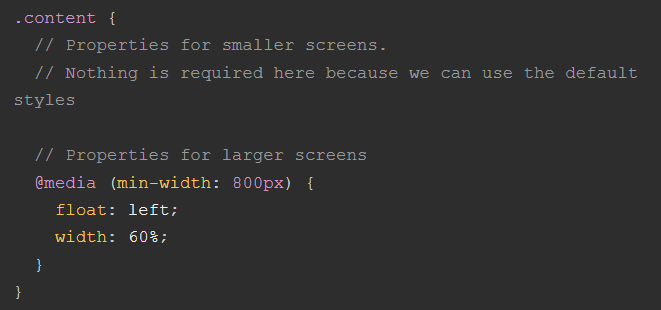
Dit is een voorbeeld van ‘Desktop First’. Bij ‘Mobile First’ keert men dit proces om. De CSS zal eerst de lay-out voor een kleiner scherm berekenen. Pas bij een breedte van 600px zal een andere lay-out berekend worden.



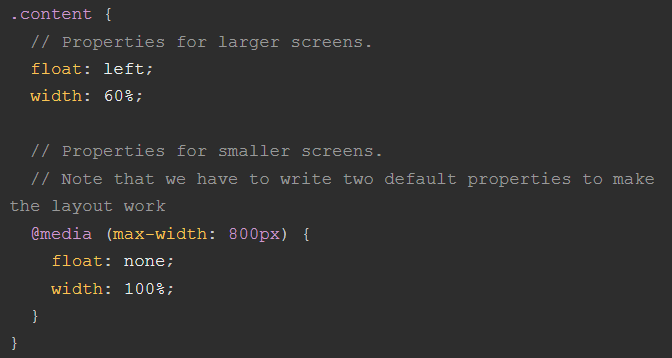
‘Mobile First’ biedt ook een voordeel voor de ontwikkelaar. Het is namelijk zo dat men bij het ontwikkelen van een mobiele user interface sneller gebruik kan maken van de default waardes van bepaalde elementen. Stel dat we een website maken waar er naast de hoofdinhoud ook een sidebar is. Het is op een kleiner scherm interessanter om deze sidebar onder de hoofdcontent weer te geven:



In het geval van een kleiner scherm gebruikt de sidebar 100% van de schermbreedte. In het geval van een groter scherm, komt de sidebar naast de hoofdinhoud te staan en zal deze pakweg 40% van het scherm innemen. Bij een ‘Mobile First’ aanpak kunnen we profiteren van de default waarde van de width property van een <div>:



Bij ‘Desktop First’ ontstaat er al sneller een situatie waar we een default waarde moeten ‘resetten’:



De ‘Mobile First’ approach bespaart ons hier 2 lijnen code. Als we deze performantie winst kunnen doortrekken naar een complexe website, kunnen we ook hier concluderen dat ‘Mobile First’ een mooi voordeel biedt.

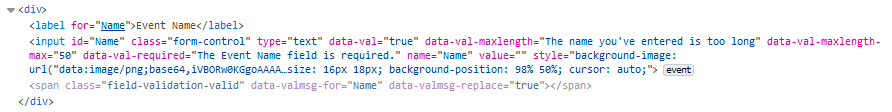
## Technologische obstakels tijdens development

##### The magic string conundrum

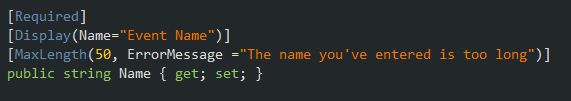
Een bijzonder complexe regio van de EventTracker applicatie is het toevoegen van nieuwe evenementen. Een evenement heeft enkele makkelijk te implementeren input velden zoals de naam van het evenement of het aantal gezochte participanten. De nieuwe MVC TagHelpers, met dank aan .Net Core, zorgen hier voor een doodeenvoudige implementering:



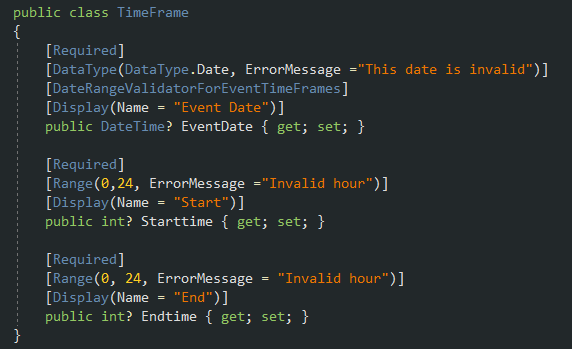
Deze TagHelpers genereren tijdens het compileren de nodige HTML voor een succesvolle modelbinding:



Hier wordt niet alleen het “Name” attribuut gegenereerd, hetgeen nodig is voor model binding, ook de “Data Annotations” waarmee de properties van het model zijn gedecoreerd, zijn omgezet naar de nodige attributen voor validatie.



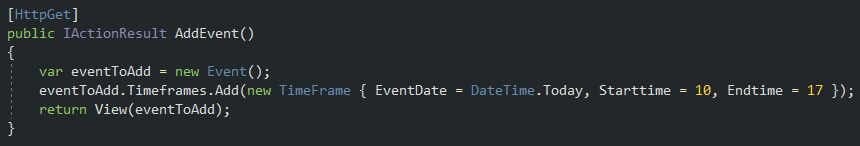
Maar niet alle properties vallen zo makkelijk te implementeren. Zo zijn er de ‘Timeframes’ van een evenement. Deze property is een “List” van het complexe type “Timeframe”. Één enkele timeframe bestaat uit een datum, start- en einduur. Elke property van een “Timeframe” is ook voorzien van enkele “Data Annotations”, waarvan zelfs 1 custom-made, namelijk [DateRangeValidatorForEventTimeFrames]:



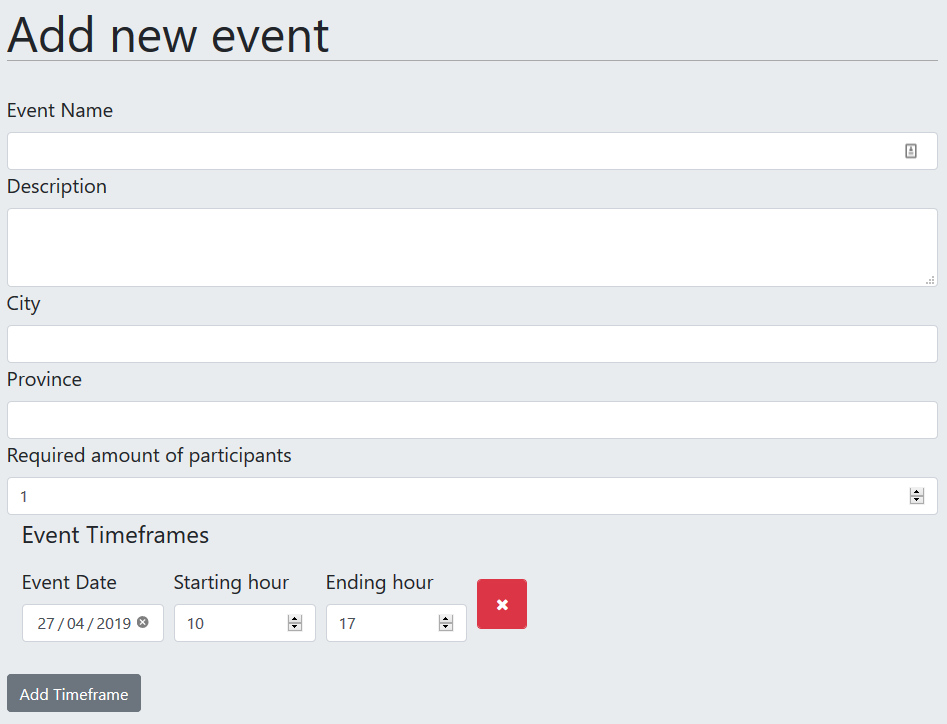
Na enkele consultaties bij onze goede vriend Google, werd duidelijk dat ook deze implementatie, mits wat extra denk- en codeerwerk, best doenbaar was. Er moet slechts één enkele variabele meegegeven worden in onze view:



Uiteraard moet de View wel voorzien zijn van een leeg “Event” waarvan reeds één “Timeframe” is geïnstantieerd. Perfect om enkele default waardes aan te leveren:



Dit alles resulteert in onderstaande view:

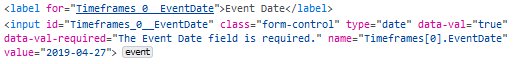


Tot nu toe loopt alles goed. Maar er is een venijnige voorwaarde verbonden met het creëren van nieuwe evenementen. Het is namelijk nooit op voorhand geweten hoeveel timeframes een evenement zal hebben. Vaak is er dat slechts één, soms zijn er dat drie of vier.

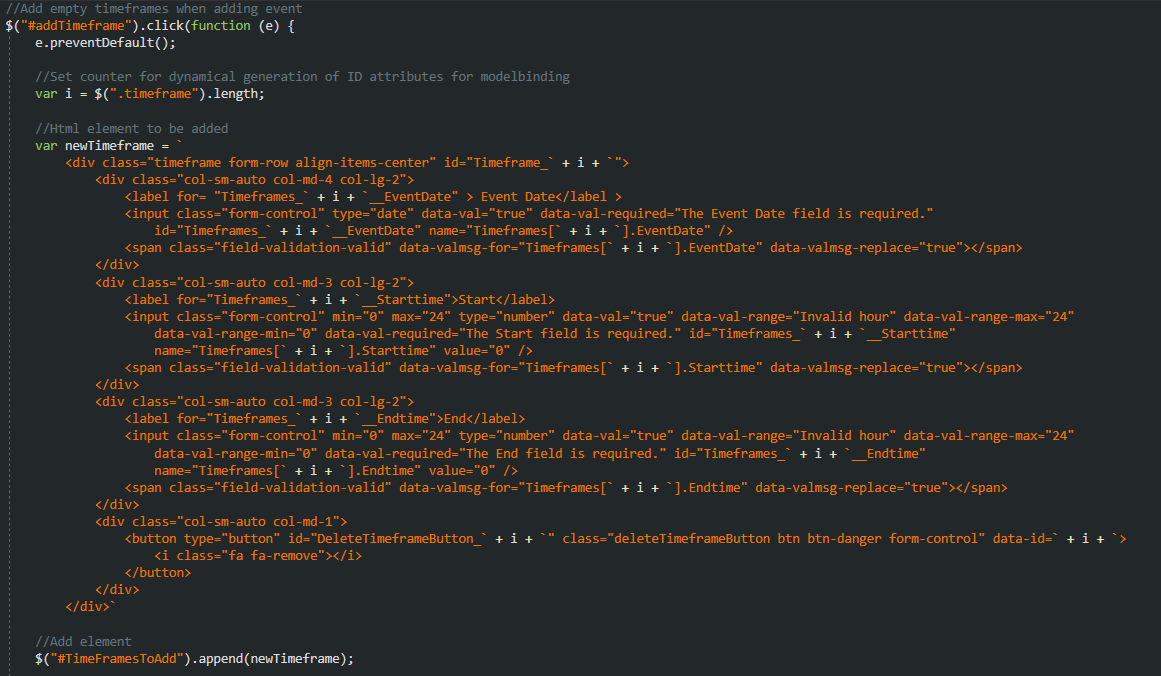
Gezien we niet op voorhand weten over hoeveel “Timeframes” een evenement zal beschikken, moeten we een gebruiker voorzien van de functionaliteit om naar believen “Timeframes” toe te voegen. Dit kan hij met de “Add Timeframe” knop. Dit genereert de nodige HTML waardoor de gebruiker zijn input kan geven.

Maar daar loopt het natuurlijk grondig mis op vlak van de nodige broodnodige validatie & modelbinding attributen die moeten ingevuld zijn. Naar welke “Timeframe” in de collectie van “Timeframes” moeten de nieuw gegenereerde input fields binden? Als we dit niet voorzien in de “Name” attributen van de input Velden, krijgen we slechts 1 “Timeframe” terug van de view, namelijk onze default “Timeframe” waarvoor een TagHelper de HTML heeft gegenereerd.

Een mogelijke oplossing hiervoor is om de door de TagHelper gegenereerde HTML te bestuderen en na te gaan of we deze HTML at runtime kunnen genereren. Deze HTML ziet er als volgt uit:



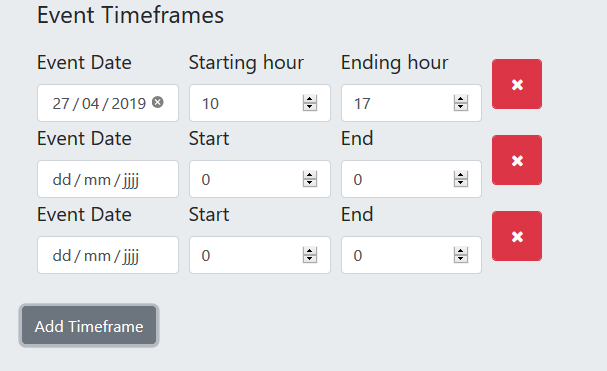
De sleutel tot het modelbinding zit hem in het “Name” attribuut. Daar zien we “Timeframes\_0\_EventDate” staat. Voor de goede verstaander: “Timeframes\_[index]\_EventDate”. Iets wat we perfect met wat sluw JavaScript werk kunnen voorzien:



Dankzij bovenstaande code gebeuren volgende dingen bij het drukken op de “Add Timeframe” knop:

* We tellen hoeveel <div> elementen er reeds zijn met de class “timeframe”. Het resultaat hiervan wordt onze index. Deze staat sowieso op 1 gezien we reeds 1 “Timeframe” hebben gegenereerd via de TagHelper.
* We creëeren de nodige HTML voorzien van alle attributen voor modelbinding en validatie. Hierbij gebruiken we onze index om het “Name” attribuut van de correcte waarde te voorzien.
* We voorzien een button om de gecreëerde HTML terug te verwijderen indien nodig.
* We plakken de nieuwe HTML achteraan de reeds voorziene HTML.

Et voila!



Maar is dit de beste oplossing voor dit probleem? Het antwoord daarop laat ik in het midden. Het was alleszins een moeizame klus om dit voor elkaar te krijgen, om nog maar te zwijgen van de afgrijselijke ‘magic string’ die wordt gebruikt om de nodige HTML te genereren.

Op Stack Overflow vinden we een alternatief waarbij de gebruiker via een aparte post scope voor “Timeframes” aan de controller duidelijk kan maken hoeveel “Timeframes” hij zou willen. Daardoor kan er in het teruggestuurde Model gebruik gemaakt worden van de nodige TagHelpers. Maar of de gebruiker het ziet zitten om daar telkens een post te moeten doen naar de server, is nog maar de vraag.

Bij de oplossing die ik heb voorzien is er quasi geen laadtijd. De JavaScript code loopt uiteraard op de client. De magic string bestaat tevens alleen maar uit attributen en elementen die tot de standaard behoort. Ook al zou men beslissen om de benaming van deze attributen of elementen te wijzigen, hetgeen me zou verbazen, vermoed ik dat de huidige browsers de ‘oude’ attributen nog jarenlang zullen ondersteunen. Ik zie dus weinig risico tot een brekende code.

En toch blijft het pijn doen aan de ogen, zo’n magic string...

##### Overposting: Een mogelijks kwetsbaarheid?