**Voorstel**

**Samenvatting**

Op Trigger kun je foto’s posten. Je kunt andere gebruikers volgen en hun posts bekijken. Trigger doet in tegenstelling tot veel andere social media sites niet aan likes. Je kunt alleen ‘trigs’ uitdelen en dat doe je als je getriggerd bent door een post. Op de homepagina verschijnen getriggerde foto’s van gebruikers. Op de persoonlijke timeline zie je de posts van de gebruikers die je volgt. Je kunt op alle posts reageren met tekst of een GIF.

**Minimum viable product**

Minimum eisen van onze website zijn een pagina waar je kunt inloggen en de mogelijkheid hebt om een account aan te maken. Na het inloggen kom je op de homepagina met alle uploads. Er moet een begeleidende tekst bij de posts geplaatst kunnen worden en de posts moeten ‘getrigd’ kunnen worden. Je moet gebruikers kunnen volgen. Op alle foto’s kan worden gereageerd met tekst of een GIF. Ook moet je kunnen uitloggen.

**Afhankelijkheden**

## Databronnen: [http://api.giphy.com](http://api.giphy.com/)

## Externe componenten: Bootstrap

Concurrerende bestaande websites: Instagram en tumblr. Deze websites zijn voornamelijk gericht op likes, terwijl Trigger zich focust op dislikes. Daarmee zullen wij ons onderscheiden van de rest.

Moeilijkste delen: Het programmeren. De design en het idee lukken, maar hoe we dit moeten programmeren wordt lastig, aangezien we geen ervaring in het maken van websites hebben.

**Features**

1. Gebruikers kunnen een account aanmaken
2. Gebruikers kunnen inloggen
3. Gebruikers kunnen andere gebruikers volgen
4. Gebruikers kunnen op de homepagina posts zien
5. Gebruikers kunnen de posts van hun volgers zien op de volgers-feed
6. Gebruikers kunnen een foto posten met beschrijving
7. Gebruikers kunnen triggeren
8. Gebruikers kunnen reageren met tekst of een gif uit een online API
9. Gebruikers kunnen scrollen
10. Gebruikers kunnen uitloggen.

**Technisch ontwerp**

**Routes**

Voor onze functies hebben we inspiratie opgedaan uit de finance opdracht.

**Login**

De login-functie zorgt ervoor dat de gebruiker kan inloggen. Voor de code zullen we GET en POST gebruiken.

**Register**

Deze functie zorgt ervoor dat een nieuwe gebruiker zich kan aanmelden. Gebruikersnamen mogen niet overeenkomen. Voor deze functie zullen we gebruikmaken van GET en POST.

**Logout**

De log-out functie zorgt ervoor dat de gebruiker kan uitloggen. We zullen geen GET en POST gebruiken.

**Follow**

De follow-functie gaat ervoor zorgen dat je als gebruiker iemand kan volgen. Je moet op het plusje klikken op de homepage om iemand te volgen. Op de speciale volgerspagina kun je de posts van de gebruikers die je volgt zien. We zullen GET en POST gebruiken.

**Homepage**

Op de homepagina zal een overzicht zijn van allerlei posts van gebruikers. We zullen geen GET en POST gebruiken

**Post**

Deze functie zorgt ervoor dat je foto’s kunt posten. We zullen gebruik maken van POST.

**Comment**

Deze functie zorgt ervoor dat je kunt reageren op een post met tekst of een GIF. We zullen gebruik maken van POST.

**Trig**

De trig-functie zorgt ervoor dat je een foto kan triggeren. We zullen POST gebruiken.

**Followers\_feed**

Deze functie zorgt voor de volgpagina. Hierop zie je alle posts van de gebruikers die je volgt. We zullen GET gebruiken.

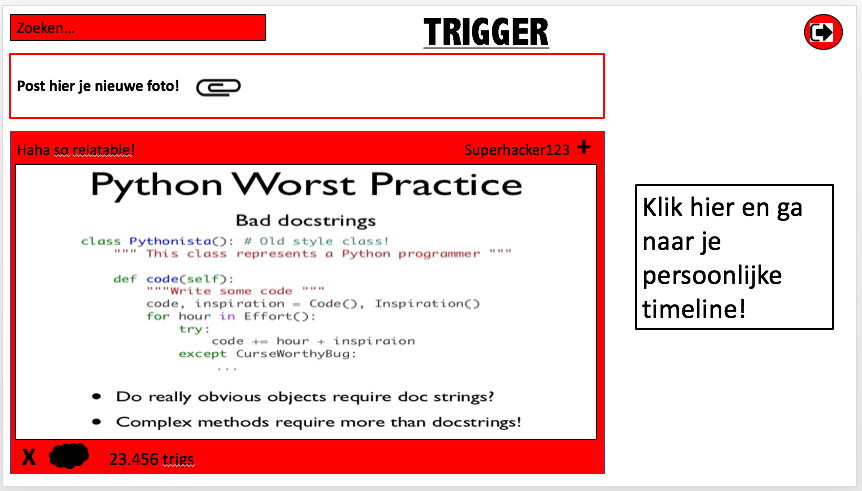
Gebruikelijke routes kunnen zijn:

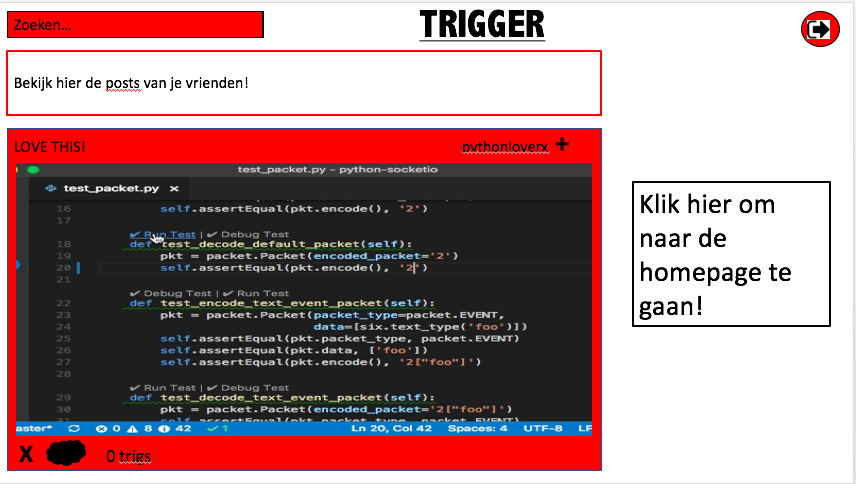
1. Register - log in - homepage - trig - comment - follow - followers\_feed - trig - comment - log out
2. Log in - followers\_feed - trig - comment - log out
3. Log in - homepage - trig - followers\_feed - trig - comment - log out

**Helper functies**

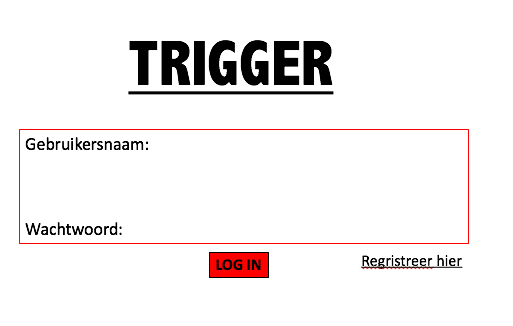
Functies als Apology en Login\_required zullen handig zijn voor onze site. Verder weten we nog niet precies hoe de site te werk gaat en is het moeilijk te benoemen welke Helper-functies we nodig gaan hebben.

**Schetsen**

****



****

****

**Plugins en frameworks**

Enkele plugins die we bij finance tegen zijn gekomen zullen ook hierbij handig zijn namelijk:

Flask, flash, redirect, render\_template, request, session, url\_for,Session

Deze zijn nodig om te navigeren op de site en te checken of een gebruiker wel is ingelogd.

-passlib.context import CryptContext

Zoals we in finance ook hebben gedaan, zo kunnen wachtwoorden als hashes worden opgeslagen.

-cs50 import SQL

We werken weer met databases met gebruikers en ook een database voor de fotos dus SQL zal ook nodig zijn.

Er zullen nog veel imports nodig zijn voor een photo feed, API reageren en triggen. Blijkt moeilijk te vinden hoe we dat moeten doen.