



FUNCTIONEEL ONTWERP

Internet of Things: GitHub

Shared Service Center
Scalda Vlissingen
Edisonweg 4a

| | |
|---------------|--------------------|
| Projectleider | : Cihan Toksöz |
| Projectleden | : Maikel Evegroen |
| Datum | : 20 november 2019 |
| Klas | : ICO43C |

INHOUDSOPGAVEN

| | | |
|---|--------------------------------------------------------------------|---|
| 1 | INLEIDING | 3 |
| 2 | BESCHRIJVING VAN DE MOGELIJKE FUNCTIONALITEITEN NIEUWE SYSTEEM ... | 4 |
| | GITHUB | 4 |
| | GIT | 4 |
| | PARALLELE TIJDLIJNEN | 4 |
| | BRANCHES IN GIT | 4 |
| | PULL REQUEST | 4 |
| | ISSUE TRACKER | 5 |
| | OPENSOURCESOFTWARE | 5 |
| | BRONCODE | 5 |
| 3 | MOSCOW-ANALYSE | 6 |
| 4 | BESCHRIJVING VAN DE GEKOZEN OPLOSSING | 6 |
| 5 | ONTWERP NIEUWE OMGEVING | 7 |
| 6 | BESCHRIJVING KOSTEN IMOLEMENTATIE TOTAAL OMGEVING | 8 |
| 7 | ORGANISATORISCHE CONSEQUENTIES | 8 |
| 8 | GEBRUIKTE BRONNEN | 8 |

1 INLEIDING

Binnen in het Shared Service Center is er behoefte aan een gelijkwaardige document ter behoefte van vooruitgang van de informatievoorziening aan deelnemende leden en externe klanten. Deze behoefte is ontstaan uit de overdraagbaarheid van projecten aan projectleden.

In dit functioneel ontwerp zijn de functionaliteiten beschreven van de GitHub tool, die gebruikt gaat worden in de testomgeving als versiebeheersysteem. Het functioneel ontwerp toont een beeld van de gewenste eisen en behoeften van de opdrachtgever. De functionaliteiten, gewenste eisen en behoeften, worden later in het document aangevuld met een MoSCoW analyse waarin een prioriteitstelling is toegepast van concreet noodzakelijke functionaliteiten tot eventuele mogelijke functionaliteiten.

2 BESCHRIJVING VAN DE MOGELIJKE FUNCTIONALITEITEN NIEUWE SYSTEEM

GITHUB

De opdrachtgever heeft de opdracht gegeven om een GitHub omgeving te ontwerpen. GitHub is een website waarop software kan geplaatst worden. GitHub is gebouwd rond het Git-versiebeheersysteem, waardoor GitHub alle mogelijkheden van Git en eigen toevoegingen aanbiedt. Het beschikt onder ander over toegangscontrole en verschillende samenwerkingsfuncties, zoals issue tracker, een forum voor het aanvragen van functies, takenlijsten en wiki's voor ieder project. Op GitHub staan veel opensourcesoftware, omdat bij openbare repositories de broncode kan worden ingekeken voor derden.

GIT

Git is een vrij verdeeld versiebeheersysteem. Het wordt ook wel een softwarebroncode-managamentproject genoemd. De nadruk ligt op snelheid.

Iedere Git-werkmap bevat de volledige pakketbron, softwarebron, repository of kortweg repo met een compleet historisch overzicht en volledige trackingcapaciteiten. Repo is een opslagplaats van software en de daarmee gerelateerde informatie zoals afhankelijkheden. De pakketbron wordt meestal aan het internet verbonden door de bestanden op ene server te plaatsen. Via een speciaal programma, een pakkerbeheerder, kan de software dan gedownload worden naar de computer van de eindgebruiker. De pakketbron kan zowel samengebrachte software bevatten als broncode die nog samengevoegd moet worden.

PARALLELE TIJDLIJNEN

Versiebeheer kan helpen om versies van je bestanden bij te houden. Als zodanig werkt het eigenlijk als een uitgebreide UNDO functie. UNDO functie is een interactietechniek die in veel computerprogramma's wordt geïmplementeerd. Hiermee wis je de laatste wijziging die in het document is aangebracht, en wordt deze teruggezet naar een oudere status. Alles wat er gedaan is kun je weer ongedaan maken en je kunt opnieuw beginnen vanaf een eerdere versie. Versiebeheerprogramma's als GIT kunnen veel meer. Ze kunnen namelijk parallelle tijdlijnen ondersteunen.

BRANCHES IN GIT

Voordat er begonnen kan worden met het maken van wijzigingen en het delen van codes, is het eerst handig over de interne structuur die Git bijhoudt. Git biedt een heleboel mogelijkheden om verschillende 'takken' bij te houden. In Git heet dit een branch. Net zoals er bij een boom een hoofdstam is waar takken afsplitsen, heeft Git dat ook met alle verschillende versies. Er is een hoofdtak, die vaak master genoemd wordt. Naast deze hoofdtak kun je zijtakken maken. De wijzigingen op een zijtak blijven dan apart van de hoofdtak, totdat er besloten wordt om deze erin terug toe te voegen. Dit heeft als voordeel dat je onafhankelijk van ander wijzigingen nieuwe dingen kunt ontwikkelen of testen. Dit is vooral handig als je met meerdere mensen tegelijk aan een project werkt, zodat er wel ergens ontwikkeld kan worden, zonder dat er daarbij mensen in de weg zitten.

PULL REQUEST

Meestal wil de gebruiker dat zijn werk terecht komt in de code basis van het oorspronkelijke programma. De veranderde versie van de code kan kenbaar gemaakt worden door middel van een 'pull request'. Dit is niks anders dan "Ik heb deze verandering gemaakt en je mag het hier ophalen als je het kunt gebruiken". De code kan opgehaald worden ('pullen') en dan vervolgens samenvoegen ('mergen') met de programma code.

ISSUE TRACKER

Een issue tracker is een computerprogramma dat gebruikt wordt om fouten in software vast te leggen en het proces om deze op te lossen te begeleiden. Een issue tracker ondersteunt software. Om dit te kunnen doen, biedt een issue tracker onder meer de volgende functies:

- Vastleggen van een issue
- Toewijzen van een issue aan een persoon
- Commentaar toevoegen aan een issue
- Toewijzen van een issue aan een ander persoon
- Een issue sluiten, dat wil zeggen vastleggen dat dit issue is opgelost

De meeste moderne issue trackers zijn webapplicaties, zodat een issue tracker centraal op een server kan worden geïnstalleerd, waarna alle leden van het team deze kunnen gebruiken via een webbrowser. Over het algemeen gebruikt een issue tracker e-mail om een teamlid te attenderen op nieuwe issues, toegewezen issues en wijzigingen in issues.

Een issue tracker kan door een projectleider worden gebruikt om werk te verdelen over de leden van een team. Binnen de issue tracker wordt dan vastgelegd wie waaraan werkt. Meestal is er binnen een issue tracker ook een mogelijkheid om aan te geven hoelang het gaat duren om een issue op te lossen. Dit kan worden gebruikt voor het maken en bijhouden van een planning

OPENSOURCESOFTWARE

Opensourcesoftware (soms ook openbronsoftware) is computerprogrammatuur waarvan de gebruiker de licentie heeft om naast gebruiker, ook ontwikkelaar te zijn. Als gevolg hiervan is het vrijgeven van de broncode onderdeel van de licentie. Dit geeft gebruikers de mogelijkheid om de software te bestuderen, aan te passen, te verbeteren, te verspreiden of verkopen. De ontwikkeling van opensourcesoftware komt vaak tot stand op publiekelijke en gemeenschappelijke wijze, door samenwerking van zowel individuele programmeurs als overheden en bedrijven. Opensourcesoftware is tevens de meest prominente ontwikkeling in de opensourcebeweging.

BRONCODE

Programma's worden eerst in een andere taal gemaakt, waarna ze worden vertaald naar een voor de processor begrijpelijke taal. Het 'programma' in de eerste, voor de maker begrijpelijke taal, heet de code. Is een synoniem van Code.

3 MOSCOW-ANALYSE

| Onderdeel | Criteria |
|---------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Functionaliteiten 'Must have' | |
| Mergen | Samenvoeging van codes. |
| Issue tracker | Fouten in software vastleggen door middel van een computerprogramma. |
| Pull request | Het doorsturen en ophalen van de codes. |
| Wiki's | Wiki's bieden een plaats voor de pakketbronnen om de routekaart van het project op te stellen, de huidige status te tonen en de software samen beter te documenteren. |
| Alerts track | Beveiligingsproblemen die van toepassing zijn op de afhankelijkheden van de gegevensopslagruimte. |
| Functionaliteiten 'Should have' | |
| GitKraken | GitKraken is een Git GUI-client voor Windows, Mac en Linux. Het helpt ontwikkelaars productiever en efficiënter te worden met GIT. |
| Functionaliteiten 'Could have' | |
| | |
| Functionaliteiten 'Won't have' | |
| | |
| Dient de klant zelf per medewerker te verstrekken | |

4 BESCHRIJVING VAN DE GEKOZEN OPLOSSING

Voordelen:

Algemeen

- Je kunt er voor kiezen om jouw project voor jezelf te houden
- Projecten kunnen veilig worden bewerkt, er wordt tijdens het bewerken een kopie gemaakt van het originele
- Je kunt openbaar een probleem rapporteren, een idee publiceren of vragen stellen
- Je kunt de revisie nummers teruglezen, dit zijn de codes die staan voor een wijziging die worden verbonden aan een persoon en tijdstip
- Gratis toegang voor studenten

Nadelen:

Algemeen

- Soms ontstaan er problemen met de klant en werkgever, als bijvoorbeeld een particuliere server als host dient voor de geschreven code

5 ONTWERP NIEUWE OMGEVING

In de screenshot is een voorbeeld te zien van een project op GitHub waarin alle bestanden staan met documentatie over het project, problemen, werkverzoeken en eventuele bronnen. Onze testomgeving moet er uiteindelijk ook zo uit zien. Voor de testomgeving zullen wij het project van Ronald van Sikkelerus nemen die zich verdiept in de multisensors. Door het project op te pakken van Ronald, kunnen we kijken en vervolgens laten zien dat het werkend zal zijn en voort kunnen zetten in een mogelijke uitbreiding voor heel het Shared Service Center.

The screenshot shows the GitHub repository page for 'aws-samples / headpose-estimator-apache-mxnet'. At the top, there are buttons for 'Watch' (1), 'Star' (8), and 'Fork' (3). Below these are tabs for 'Code', 'Issues' (0), 'Pull requests' (0), 'Projects' (0), and 'Insights'. The repository description states: 'Head Pose estimator using Apache MXNet. HeadPose_ResNet50_Tutorial.ipynb helps you to walk through an entire work flow of developing a CNN model from the scratch including data augmentation, fine-tuning, training, saving check-point model artifacts, validation and Inference.' Below the description, statistics show 60 commits, 1 branch, 0 releases, 2 contributors, and the Apache-2.0 license. A 'Find file' button and a 'Clone or download' button are present. The file list shows various files and folders with their commit messages and dates. The latest commit is 520e11a on Jun 18.

| File/Folder | Commit Message | Time Ago |
|----------------------------------------|-----------------------------------------------|--------------|
| .github | Creating initial file from template | 7 months ago |
| HeadPose_SageMaker_PythonSDK | Add files via upload | 2 months ago |
| testIMs | Add files via upload | 2 months ago |
| CODE_OF_CONDUCT.md | Creating initial file from template | 7 months ago |
| CONTRIBUTING.md | Creating initial file from template | 7 months ago |
| HeadPose_ResNet50_Tutorial.ipynb | Update HeadPose_ResNet50_Tutorial.ipynb | 6 months ago |
| HeadPose_ResNet50_Tutorial_Gluon.ipynb | Update HeadPose_ResNet50_Tutorial_Gluon.ipynb | 6 months ago |
| LICENSE | Creating initial file from template | 7 months ago |
| NOTICE | Creating initial file from template | 7 months ago |
| README.md | Update README.md | 2 months ago |
| preprocessingDataset_py2.py | Add files via upload | 6 months ago |

6 BESCHRIJVING KOSTEN IMOLEMENTATIE TOTAAL OMGEVING

Om aan te tonen hoeveel tijd dit project in beslag zal nemen tevens als hoeveel de aan te schaffen middelen zullen kosten hebben wij als projectgroep een schema opgesteld.

| Tijdsbesteding, Project veilige GitHub | | | | |
|----------------------------------------|-----------------|---------------------------|-----------------|-------------------|
| Product | Aantal personen | Tijd in uren | Totaal per week | Totaal van sprint |
| Documentatie | 2 | 4x 2 projecturen per week | 8 uur | 32 uur |
| Testomgeving | 2 | 4x 2 projecturen per week | 8 uur | 32 uur |
| Totaal | | | | 64 uur |

Om GitHub te gebruiken moet er een account aangemaakt worden. Doordat wij als studenten inloggen, kunnen wij een gratis versie nemen van GitHub. Er zijn mogelijk geen kosten verbonden aan het project. Zodra er kosten worden geconstateerd, zal het besproken worden met de projectcoach en de opdrachtgever.

| Product kosten | | |
|----------------|--------|------------------|
| Product | Aantal | Verwachte kosten |
| | | |
| | | |
| | | |
| Totaal | | € |

7 ORGANISATORISCHE CONSEQUENTIES

Wij als organisatie zullen straks gebruiken maken van de nieuwe GitHub omgeving. De GitHub omgeving zal op elk device bereikbaar moeten zijn. Zodra het verbonden is met het netwerk, is het te bereiken voor elk eindgebruiker die er gebruik van moet maken binnen de organisatie. Dit zorgt er voor dat de gebruikers verder kunnen aan het werk waar ze mee bezig waren. De gebruikers zullen moeten leren werken met de nieuwe omgeving. Aan de hand van een handleiding, zullen de gebruikers geleid worden voor de nieuwe omgeving van GitHub. De projectleiders zullen procedure van Github toegewezen krijgen, waarin staat hoe ze te werk kunnen gaan in de GitHub omgeving.

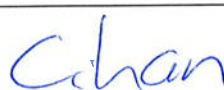

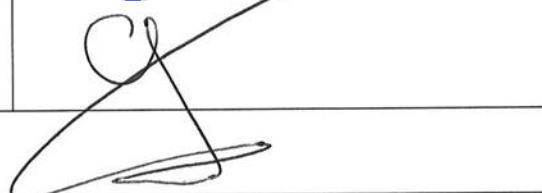
8 GEBRUIKTE BRONNEN

Binnen dit project hebben wij gebruik gemaakt van een aantal webbronnen. Deze webbronnen zijn als volgt:

GitHub Education | <https://education.github.com/students>

GitHub Help | <https://help.github.com/en>

Quora | <https://www.quora.com/What-are-the-pros-and-cons-of-GitHub-versus-Bitbucket>

| | |
|-----------------------------------------------------|------------------------------------------------------------------------------------|
| Handtekeningen | |
| Student: <i>Naam : Cihan Toksöz</i> |  |
| Projectcoach: <i>Naam : Jeroen van Os</i> |  |
| Opdrachtgever: <i>Naam : Kees Krijger</i> |  |
| Datum: 20 november 2019 | |