

Horeca Registration System

Gedistribueerde Systemen 2

Robbe De Groeve

4ELICTsa

december 2020

Inhoud

Inleiding.....	5
1 Programma opbouw	5
2 SWOT analyse	5
2.1 <i>Sterktes.....</i>	<i>5</i>
2.2 <i>Zwaktes</i>	<i>6</i>
2.3 <i>Opportunities</i>	<i>6</i>
2.4 <i>Gevaren.....</i>	<i>6</i>

Inleiding

In dit verslag zal de implementatie van een voorgesteld horeca registratie systeem overlopen worden. Hierbij zal er vooral naar de opbouw van het programma dat dit systeem implementeert gekeken worden en zal deze implementatie dan ook geanalyseerd worden aan de hand van een SWOT analyse.

1 Programma opbouw

Voor de implementatie van dit horeca registratie systeem werd zoveel mogelijk modulair gewerkt. Het project wordt eerst verdeeld in twee delen met de server, common en client mappen. Deze server en client mappen worden dan nog eens verder verdeeld in de Registrar, Mixing proxy en Matching service deelprojecten bij de server en User, Facility en Practitioner bij de client. De structuur van de verschillende deelprojecten is zoveel mogelijk verdeeld in gelijkaardige Java packages:

- **Connection package:** Start de RMI server en client connecties en is verantwoordelijk voor het versturen van de data naar de juiste service
- **Controller package:** Bevat het grootste deel van de logica. Deze package verwerkt de data en stuurt deze dan door tussen de verschillende andere packages
- **GUI package:** Is verantwoordelijk voor de afbeelding van de data en de interactie met de gebruiker. Hiervoor wordt gebruik gemaakt van JavaFX.
- **Data package:** Zorgt voor de connectie met de Heroku PostgreSQL database en het versturen van de data heen en terug van de database.

De common map bevat bepaalde klassen waarvan zowel bij de server als de client applicaties gebruik van wordt gemaakt.

2 SWOT analyse

Sterktes <ul style="list-style-type: none">• Privacy gericht• Eenvoudig in gebruik	Zwaktes <ul style="list-style-type: none">• Geen authenticatiemethode geïmplementeerd
Opportunities <ul style="list-style-type: none">• Communicatie robuuster maken	Gevaren <ul style="list-style-type: none">• Oplossen/ontdekken van problemen• Communicatieproblemen

2.1 Sterktes

Het grootste voordeel van dit horeca registratiesysteem is de grotere privacy gerichtheid van dit gedistribueerd systeem in vergelijking tot andere gecentraliseerde uitvoeringen.

Als de drie server services geschieden blijven van elkaar kan de verspreiding van corona opgevolgd worden aan de hand van tokens, zonder dat deze tokens gelinkt kunnen worden aan de persoon die deze oorspronkelijk verstuurd heeft.

Een ander voordeel is de eenvoudige werking voor de eindgebruiker. Dit is in tegenstelling tot de actuele registratiemethode met pen en papier. Deze methode vraagt veel werk en tijd om de gegevens te verwerken en verhoogd de kans op fouten door bv: onleesbaar geschrift. Bij deze implementatie moet de horeca faciliteit enkel een QR code aanmaken en de gebruiker deze scannen om zijn bezoek aan de faciliteit te registreren, Alle andere stappen gebeuren in het systeem zelf. Hierdoor is de kans op fouten veel verminderd.

2.2 Zwaktes

In de huidige implementatie wordt er gewerkt met een identificatieparameter om de identiteit van de faciliteiten en gebruikers te achterhalen. Hierbij kunnen de gebruikers en faciliteiten zich identificeren aan de hand van respectievelijk hun Gsm-nummer en ondernemingsnummer. Door tijdsgebrek werd hiervoor wel geen authenticatiemethode aan de hand van bv: een wachtwoord geïmplementeerd en kunnen gebruikers zich hierdoor dus identificeren met een andere identificatieparameter. Een authenticatiemethode is iets wat zeker geïmplementeerd zal moeten worden voordat het programma in gebruik genomen kan worden.

2.3 Opportuniteiten

Buiten een authenticatiemethode die nog geïmplementeerd zal moeten worden zijn er ook nog een paar uitbreidingen mogelijk. Het grootste deel van deze uitbreidingen heeft te maken met de RMI verbindingen tussen de verschillende applicaties. In deze implementatie is er geen rekening gehouden met de verschillende problemen die zich binnen deze communicatiemethode kunnen voortdoen, zoals bijvoorbeeld een server die uitvalt. In dit geval zou bij de huidige implementatie het hele systeem terug heropgestart moeten worden. Bij een toekomstige uitbreiding zou het handig zijn mochten de andere services gewoon wachten tot de foutieve server heropgestart is om hierna automatisch terug contact te nemen en de werking van het systeem terug voort te zetten

2.4 Gevaren

Naast de voordelen die de gedistribueerde architectuur biedt, brengt deze methode ook enkele gevaren met zich mee. Eén gevaar hierbij is het onderhouden van de software. Bij de ontwikkeling kan het systeem makkelijk gesimuleerd worden met testdata, maar eens het systeem operationeel is zal het oplossen van bugs of zelfs het detecteren ervan moeilijk zijn aangezien deze voorgestelde methode ervanuit gaat dat de drie server applicaties niet met elkaar samenwerken. Hierdoor wordt het moeilijk om het afgelegde pad van de data te volgen en de aard van de fouten te achterhalen of zelfs te detecteren.

Ook de communicatie tussen de services kan gevaren met zich mee brengen. Door de gedistribueerde aard zijn er veel verschillende verbindingen nodig die opgesteld en beveiligd moeten worden tussen de client en de server en tussen de servers onderling. Aangezien de drie servers ook verschillende taken hebben maar toch afhankelijk zijn van elkaar zal bij het falen van één server de werking van het hele systeem belemmerd worden.

*De volledige implementatie van het horeca registratie systeem is terug te vinden op
GitHub: <https://github.com/RobbeRDG/HorecaRegistration>*

TECHNOLOGIECAMPUS GENT
Gebroeders de Smetstraat 1
9000 GENT, BELGIË
TEL. 09 265 86 10
WWW.KULEUVEN.BE

