



[Bezoekers registratie Systeem]

REVISION	DATE	AUTHOR
1	29/08/2022	Angelo Dejaeghere
2	29/08/2022	Alien Carlier
3	30/08/2022	Angelo Dejaeghere

Functioneel

Doel van de applicatie

Voor een bedrijvenpark met een centrale receptie willen wij een applicatie laten maken die de bezoekers van het bedrijvenpark kan registreren. Dit is nodig in kader van de brandveiligheid, op deze manier weten we te allen tijde wie er aanwezig is op de site. Iedereen die op bezoek komt bij één van de bedrijven moet zich eerst aanmelden aan de receptie. Om dit zo vlug mogelijk te laten verlopen krijgen de bezoekers de mogelijkheid om zich aan te melden via een touchscreen die voorzien zal worden aan de inkom. Via het touchscreen kunnen ze hun eigen gegevens en de persoon waarmee ze een afspraak hebben ingeven. Bij het vertrek kunnen de bezoekers zich terug uitschrijven via hetzelfde touchscreen. Wij houden hierbij zeker rekening met de privacy van de bezoekers, Want het kan niet de bedoeling zijn dat een bezoeker op de hoogte is van de aanwezigheid van anderen.

[Nice to have]

Aangezien er ook parking aanwezig is willen we ook de parkeerplaatsen gaan beheren. De parking heeft een vast aantal parkeerplaatsen die door iedereen gebruikt kunnen worden. Een bedrijf dat op het bedrijvenpark gevestigd is kan een aantal parkeerplaatsen reserveren voor eigen gebruik. Deze parkeerplaatsen worden in een contract vastgelegd. Als een werknemer of bezoeker de parking oprijdt moet deze persoon zijn nummerplaat ingeven en aangeven voor welk bedrijf hij/zij komt. Zolang er plaats op de parking is mag iemand de parking oprijden. Indien er geen vrije plaatsen meer zijn moet dit worden aangegeven na het ingeven van de nummerplaat. De parking is volzet als alle niet gereserveerde plaatsen opgebruikt zijn. Er kunnen nog steeds vrije gereserveerde plaatsen zijn die een ander bedrijf toebehoren

Bij het buitenrijden moet terug de nummerplaat worden ingegeven en wordt er opnieuw een parkeerplaats vrijgegeven.

Opdeling van de applicatie

De applicatie zou uit 2 delen moeten bestaan, enerzijds een deel waar de bezoeker zich in en uit kan schrijven en anderzijds een deel om de gegevens te beheren maar ook de bezoekers op te zoeken. Het deel voor de bezoekers moet eenvoudig te bedienen zijn via een touchscreen, het andere een desktop applicatie zijn die door de receptiemedewerker gebruikt wordt.

[nice to have]

De applicatie voor de parking bestaat uit 2 delen, een deel voor het binnenrijden waar je een bedrijf kunt selecteren samen met het ingeven van een nummerplaat. het tweede deel dient voor het buitenrijden waar je enkel een nummerplaat kan ingeven. Het bekijken van het aantal wagens op de parking en de toebehorende parkeerplaatsen per bedrijf is geïntegreerd in de applicatie van de receptiemedewerker.

Algemene richtlijnen

- De applicaties moeten gebruiksvriendelijk worden opgemaakt
- Data wordt nooit echt verwijderd, maar wordt verborgen voor de gebruiker
- Lijst weergaves in de applicatie moeten gefilterd kunnen worden op alle zichtbare velden
- Vanuit een lijstweergave moet het mogelijk zijn om volgende acties uit te voeren
 - o Nieuw item toevoegen
 - o Detail weergave openen
 - o Bewerk weergave openen
 - o Verwijderen van een item
- Bij acties die data aanpassen moet er telkens een bevestiging gevraagd worden aan de gebruiker

Beschrijving per onderdeel

Gegevens die hieronder in het vet zijn aangeduid, zijn dingen die verplicht in te vullen zijn bij het aanmaken en/of het bewerken.

Beheer van bedrijven

De bedrijven op het bedrijvenpark worden bijgehouden met volgende gegevens:

- **Naam**
- **BTW nummer** (controle op geldigheid)
- Adres
- Telefoon
- **Email** (controle op geldigheid)

Per bedrijf worden ook de werknemers bijgehouden zodat deze te selecteren zijn in het aanmeldscherm. Per werknemer worden volgende gegeven bijgehouden:

- **Naam & voornaam**
- Email (controle op geldigheid)
- **Functie**

[nice to have]

Per bedrijf moet het contract voor de parking worden bijgehouden met volgende gegevens:

- **Bedrijf**
- **Startdatum contract**
- **Einddatum contract**
- **Aantal plaatsen**

Beheer van het bezoek

Voor een bezoeker moet volgende data worden bijgehouden:

- **Naam**
- **Email**
- **Bedrijf van de bezoeker**

Bij een bezoek moet volgende data bijgehouden worden:

- **Bezoeker**
- **Bedrijf** waar de bezoeker een afspraak heeft
- **Persoon** met wie de bezoeker een afspraak heeft
- **Starttijd** van het bezoek
- Eindtijd van het bezoek

Technisch

Technische randvoorwaarden

- De code wordt ontwikkeld met Visual Studio en bevat één solution met verschillende projecten
- Er wordt gebruik gemaakt van C# in een .NET Core omgeving
- Als databank wordt er gebruik gemaakt van SQL Server
- De technologie om met de databank te communiceren is ADO.NET
- De code base wordt beheerd in Github
- Voor de user interface wordt er gebruik gemaakt van WPF
- Configuratie info wordt in config bestanden beheerd
- Voor het schrijven van Unit Tests maken we gebruik van xUnit

Opmerkingen:

- Voorzie een flexibel ontwerp zodanig dat veranderingen geen al te grote impact hebben
- Zorg voor een gebruiksvriendelijke interface.
- Er worden geen extra packages gebruikt (indien nodig, moet dit eerst worden besproken en nadien goedgekeurd)

Werkwijze

Dit project wordt uitgevoerd door een team van programmeurs (in dit geval 2 of 3 personen) waarbij samenwerken een belangrijke rol speelt. Naast het schrijven van code zijn er nog een aantal andere taken (misschien niet de leukste) die moeten worden uitgevoerd. Het uitvoeren van een project is niet enkel de applicatie opleveren op een voorgestelde datum. In de eerste plaats moeten we er zeker van zijn dat hetgeen we opleveren wel degelijk is wat de klant wenst. Daarom zullen we op een 'agile'-achtige manier te werk gaan. Dat wil zeggen dat we een aantal tussentijdse opleveringen zullen doen waarbij we afchecken met de klant of dit is volgens de verwachtingen verloopt. Daarnaast moeten we ook rekening houden dat deze applicatie onderhouden moet worden (misschien niet door ons) en dat er dus voldoende documentatie beschikbaar is zodat een andere programmeur dit ook kan uitvoeren.

Om dit vlot te laten verlopen, definiëren we een aantal rollen en verantwoordelijkheden. Deze rollen zijn:

Organisatie-rol :

- Inplannen meetings
- Rapporteer over de taakverdeling
- Zorgt voor de opvolging van het project (ook rapportage)

Zorgt voor voldoende documentatie**Business-rol :**

- Klaart elke onduidelijkheid over de business uit aan het team
- Controleert op inhoudelijke correctheid
- Stelt (inhoudelijke) scenario's op die moeten worden getest
- Zorgt ervoor dat de klant duidelijk wordt geïnformeerd over de vooruitgang

Testing-rol :

- Zorgt dat er voor elke klasse de nodige unit tests zijn
- Zorgt ervoor dat alle unit tests steeds op groen staan (ook na aanpassingen)
- Geeft overzicht van wat er getest is
- Zorgt voor integratie tests
- Zorgt ervoor dat volledige applicatie is getest

Architectuur-rol :

- Is verantwoordelijk voor het ontwerp
- Maakt beslissingen over architectuur
- Documenteert en motiveert beslissingen
- Is verantwoordelijk voor de code base (Git)

Opmerking : als er bij verantwoordelijkheden staat “zorgt ervoor dat ...” wil dat niet per se zeggen dat de persoon met deze rol ook deze taak moet uitvoeren. Deze persoon moet ervoor zorgen dat de taak wordt uitgevoerd.

Voorbeeld: Bij de organisatie-rol staat er dat deze persoon ervoor zorgt dat er voldoende documentatie is. Dit wil niet zeggen dat deze persoon alle documentatie moet schrijven. Dit betekent wel dat deze persoon er moet op toezien dat elke klasse is gedocumenteerd (op een eenduidige manier), dat overzicht schema's van de klassen (rol architectuur) en testen (rol testing) op één plaats beschikbaar zijn en dat de documentatie inhoudelijk voldoet.

Git/Github

De code moet beheerd worden in Git, en dit is een taak van elk lid van het team. Dit wil zeggen dat iedereen een basiskennis moet hebben aangaande het gebruik van Git/Github. Onderstaande links wijzen naar een aantal tutorials die daarbij kunnen helpen.

Opmerking: wanneer er aan verschillende taken tegelijkertijd wordt gewerkt door verschillende teamleden is het belangrijk dat we elkaar niet 'storen', branches/pull requests zouden daarbij zeker kunnen helpen.