2024 - 2025

|  |
| --- |
|  |
| Vergelijkend Onderzoek voor Wachtlijstbeheer |
|  |

Jonas Quintiens  
Student Bachelor in de Toegepaste Informatica – Applicatieontwikkeling

Inhoudsopgave

[Inleiding 2](#_Toc195550081)

[Doelstelling 2](#_Toc195550082)

[Conclusie 4](#_Toc195550083)

# Inleiding

Bij het ontwikkelen van een wachtlijstfunctionaliteit in een bestaande applicatie is het belangrijk om een methode te kiezen die zowel eerlijk als efficiënt is voor het beheren van inschrijvingen wanneer de capaciteit beperkt is. In dit onderzoek vergelijken we vier verschillende benaderingen om een wachtlijst te beheren, elk met hun eigen voor- en nadelen.

# Doelstelling

Het doel is om de meest geschikte wachtlijstmethode te kiezen op basis van de specifieke noden van de organisatie. De gekozen methode moet zowel technisch haalbaar zijn als een goede gebruikerservaring bieden. De evaluatie gebeurt aan de hand van zes criteria, met behulp van de Weighted Ranking Method (WRM).

De Te Vergelijken Benaderingen

**FIFO (First In, First Out)**  
→ Wie zich eerst aanmeldt, krijgt als eerste de kans om in te schrijven bij een vrijgekomen plaats.

**Prioriteitswachtlijst**  
→ Gebruikers krijgen een score of label (bv. VIP, trouwe klant), en worden op basis daarvan hoger in de lijst geplaatst.

**Loting**  
→ Bij vrijkomende plaatsen wordt willekeurig iemand uit de wachtlijst gekozen.

**Batch invites / Golven**  
→ Gebruikers worden in groepen uitgenodigd, bv. 5 per keer, zodat meerdere mensen tegelijk kunnen inschrijven.

Vergelijkingscriteria

|  |  |  |
| --- | --- | --- |
| Criterium | Gewicht (%) | Uitleg |
| Eerlijkheid | 25% | Hoe objectief en transparant is het systeem voor gebruikers? |
| Beheercomplexiteit | 20% | Hoe moeilijk is het voor beheerders om te configureren en op te volgen? |
| Gebruikerservaring | 15% | Hoe duidelijk en frustratievrij is het voor gebruikers? |
| Technische complexiteit | 15% | Hoe moeilijk is de implementatie technisch gezien? |
| Flexibiliteit | 15% | Kan je makkelijk iemand manueel uitnodigen, aanpassen of verwijderen? |
| Risico op lege plaatsen | 10% | Hoe groot is de kans dat iemand niet snel reageert en plaats onbezet blijft? |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Criterium | Gewicht (%) | FIFO (Score) | Prioriteit (Score) | Loting (Score) | Batch/Golf (Score) |
| Eerlijkheid | 25% | 10 \* 25% = 2.5 | 1 \* 25% = 0.25 | 7 \* 25% = 1.75 | 8 \* 25% = 2 |
| Beheercomplexiteit | 20% | 10 \* 20% = 2 | 5 \* 20% =  1 | 9 \* 20% = 1.8 | 7 \* 20% = 1.4 |
| Gebruikerservaring | 15% | 8 \* 15% =  1.2 | 6 \* 15% = 0.9 | 6 \* 15% = 0.9 | 9 \* 15% = 1.35 |
| Technische complexiteit | 15% | 10 \* 15% = 1.5 | 1 \* 15% = 0.15 | 5 \* 15% = 0.75 | 5 \* 15% = 0.75 |
| Flexibiliteit | 15% | 4 \* 15% = 0.6 | 10 \* 15% = 1.5 | 1 \* 15% = 0.15 | 5 \* 15% = 0.75 |
| Risico op lege plaatsen | 10% | 8 \* 10% = 0.8 | 7 \* 10% = 0.7 | 4 \* 10% = 0.4 | 10 \* 10% = 1 |
|  |  |  |  |  |  |
| Totaal |  | 8.6 | 4.5 | 5.75 | 7.25 |

# Conclusie

Uit de Weighted Ranking Method (WRM) blijkt De FIFO-benadering blijkt de beste keuze voor het beheren van een wachtlijst vanwege de volgende voordelen:

* Eenvoudige implementatie: De "wie eerst komt, wie eerst maalt"-methode is eenvoudig te begrijpen en snel te implementeren.
* Transparantie: Het biedt een eerlijke en duidelijke volgorde, wat de gebruikerservaring ten goede komt.
* Lage initiële complexiteit: De basisfunctionaliteit is gemakkelijk te configureren, wat het systeem snel operationeel maakt zonder ingewikkelde regels.
* Schaalbaarheid: FIFO kan eenvoudig worden uitgebreid met extra functionaliteit, zoals prioriteitsniveaus of dynamische aanpassingen, zonder dat het de bestaande structuur verstopt of moeilijk maakt.
* Makkelijker beheer: Beheerders kunnen snel en effectief de wachtlijst beheren zonder dat er veel technische kennis vereist is.

In vergelijking met alternatieven zoals prioriteitswachtlijsten of loting biedt FIFO een robuuste en flexibele oplossing die geschikt is voor de meeste gevallen, vooral wanneer snelheid en eenvoud belangrijk zijn.