Technisch ontwerp

Door: Jordan Groothuizen

# Inhoud

### Blz. 1 – Titel

### Blz. 2 – Inhoud

### Blz. 3 – Data flow Diagram

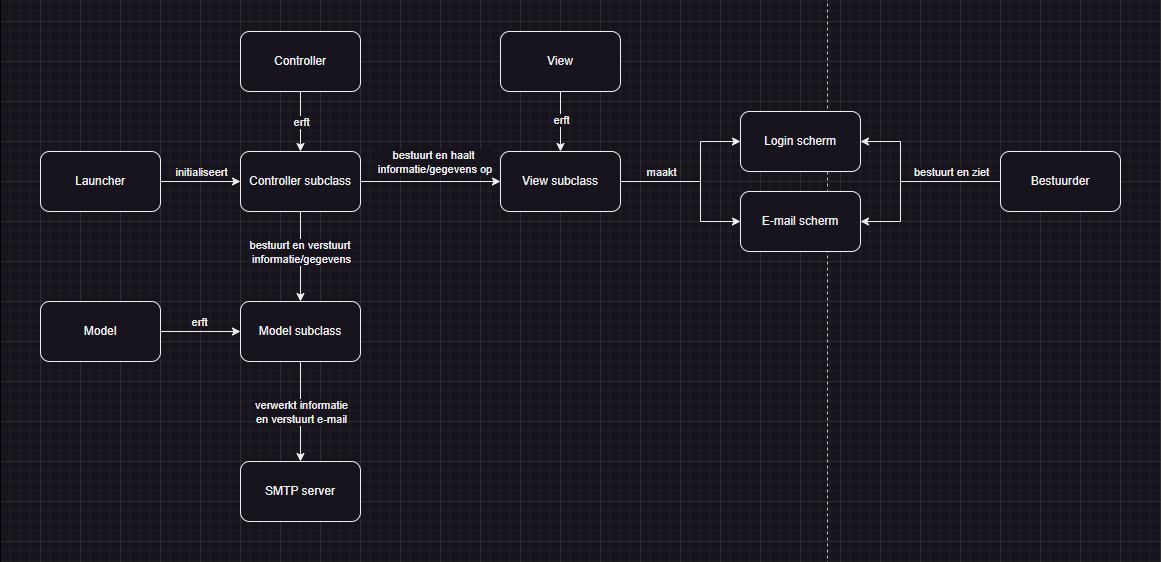
### – Uitleg van de classes

### Blz. 4 – Toekomstige plannen

### Blz. 4 – Belangrijke punten

### Blz. 5 – geraadpleegde bronnen

# Data Flow Diagram



## Uitleg van de classes

### Launcher

Initialiseert een controller subclass, en doet in dit geval verder niks.

### Controller

Maakt, Overheerst en bestuurt de view en model classes. De controller kan ook compleet zelfstandig werken via de console of in de achtergrond.

### View

Erft de JFrame class van java.swing en creëert een venster. De view krijgt instructies van de controller class.

### model

Slaat gegevens op en verwerkt ze. In dit geval worden de gegevens gebruikt om een e-mail te sturen via een SMTP server (Simple Mail Transfer Protocol server). De model krijgt instructies van de controller class.

# Toekomstige plannen

Er kan gemakkelijk controller, view of model classes toegevoegd worden. Misschien als er een manier is om e-mails op te halen van een e-mail adres, kan er een lijst gemaakt worden met alle ontvangen e-mails op dat adres.

# Belangrijke Punten

Om het programma te testen heb ik het open source programma Papercut gebruikt. Met Papercut kan je e-mails sturen via de localhost (127.0.0.1) en het simuleert dan hoe de e-mail er uit zal zien op een officiële e-mail server.

# Geraadpleegde bronnen

### - w3schools.com

### - docs.oracle.com

### - stackoverflow.com