

MAILING LIST

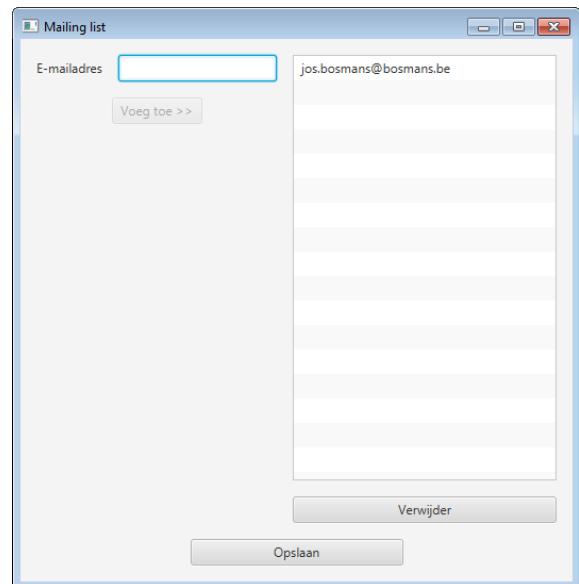
We bouwen een applicatie om een lijst van email adressen op te slaan. Opslaan doen we met de knop "Opslaan". Inladen gebeurt automatisch bij het opstarten van de applicatie.

Wanneer je applicatie voor het eerst start zal je de melding krijgen dat er een bestand niet gevonden kan worden!

(De GUI laat enkel eenvoudige email adressen toe)

De voornaamste klassen die we voor deze oefening nodig hebben zijn:

- [java.io.File](#)
- [java.io.PrintWriter](#)
- [java.util.Scanner](#)
- [java.io.IOException](#)
- [javafx.scene.control.Alert](#)



Raadpleeg in eerste instantie de [Java documentatie](#) als je ergens vast zit!

1 WIREFRAME

De GUI van deze oefening is reeds uitgewerkt. Het wireframe mag je als oefening maken met behulp van een tool naar keuze.

2 HOOFDSCHERM AANMAKEN – MVP

De *model* klasse is **MailingList**. Een andere belangrijke model klasse is **EmailAddress**. We vullen het model aan in punt 3.

De *view* klasse is **MailingListView**. Deze klasse is volledig uitgewerkt.

De *presenter* klasse is **Presenter**. Deze klasse is volledig uitgewerkt.

De **Main** klasse is gegeven.

3 MODEL

In de klasse **MailingList** werken we de methodes **loadFromDisk** en **saveToDisk** uit.

3.1 De methode loadFromDisk

- Maak een **Scanner** aan en geef **BESTAND** mee aan de constructor (zie bovenaan de klasse **MailingList**).
- **Zolang** de **Scanner** nog een volgende lijn heeft ([hasNextLine](#)) vraag je de volgende lijn op ([nextLine](#)) en maak je een nieuw **EmailAddress** object aan met deze lijn. Voeg elk aangemaakt **EmailAddress** object toe aan de verzameling genaamd **addresses**.
- Vang eventuele **IOExceptions** **niet** op! Deze worden door de view afgehandeld.

- Zorg er voor dat de **scanner** in alle gevallen correct wordt gesloten.
(Dit kan je doen met behulp van *try-with-resources* indien dit concept al behandeld is geweest in de les.)

3.2 De methode **saveToDisk**

- Maak een **PrintWriter** aan en geef **BESTAND** mee aan de constructor (zie bovenaan de klasse **MailingList**).
- **Voor elk** email adres in de verzameling genaamd **addresses** schrijf je een lijn weg met behulp van de methode **format**. Zorg er voor dat alle email adressen op een aparte lijn staan!
- Vang eventuele **IOExceptions** **niet** op! Deze worden door de view afgehandeld.
- Zorg er voor dat de **PrintWriter** in alle gevallen correct wordt gesloten.
(Dit kan je doen met behulp van *try-with-resources* indien dit concept al behandeld is geweest in de les.)

4 UI OPBOUWEN

De view is volledig uitgewerkt voor deze opdracht.

5 AFHANDELEN EVENTS

De presenter is volledig uitgewerkt voor deze opdracht.