Ontwerp Document

Naam: Steef v.d. Weijdeven

Project: Portfolio Desktop

Recentste aanpassing: 09/06/2020

Contents

[Inleiding 3](#_Toc42550260)

[Wireframe 4](#_Toc42550261)

[Database model 5](#_Toc42550262)

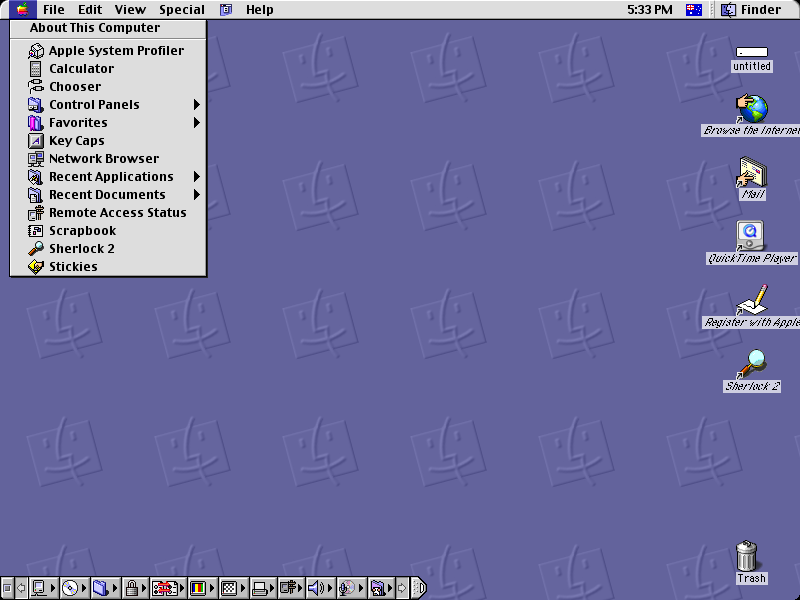
[Klassen diagram 6](#_Toc42550263)

# Inleiding

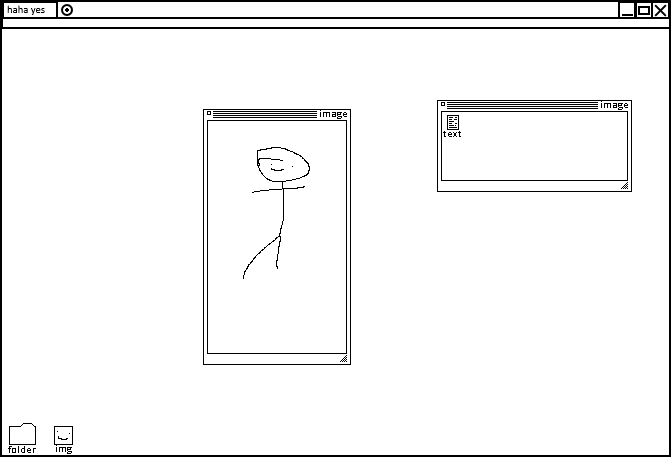
Mijn persoonlijke project is een portfolio website gemaakt in de stijl van een ouderwetse desktop (mac os 9 als hoofd inspiratie)[1]

Er zullen meerdere dingen mogelijk zijn op de pagina naast het lezen van mijn portfolio:

* Tekenen door middel van jspaint (zal later visueel aangepast worden om thematisch te passen)
* Muziekspeler door middel van webamp (zal net zoals jspaint aangepast worden om thematisch te passen)

 [1]Mac OS 9

# Wireframe



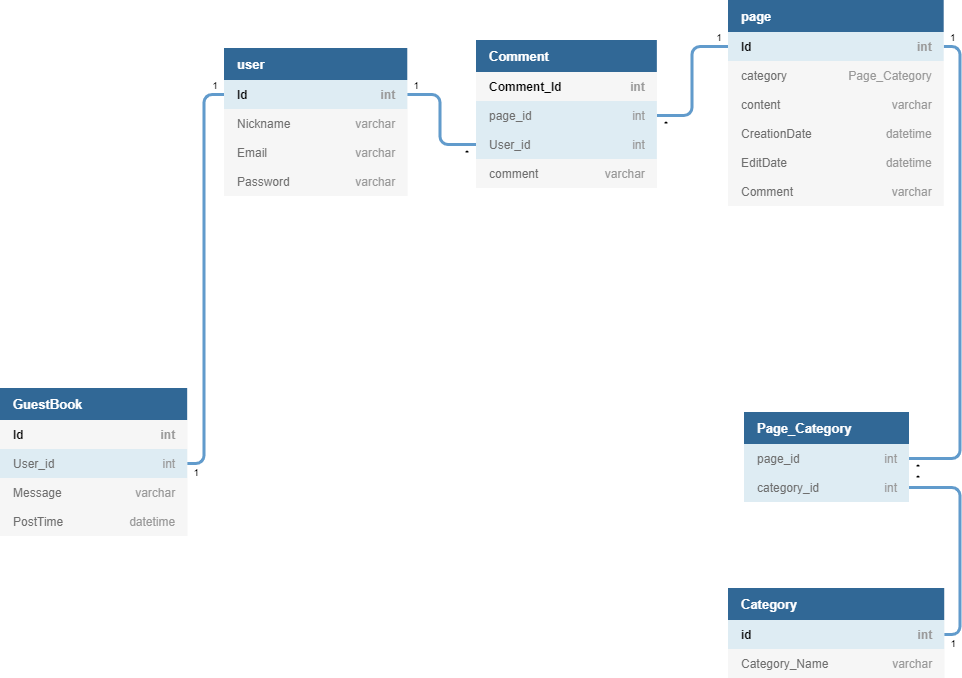
De website is op een visueel niveau erg lijkend op een desktop vanuit de jaren 80/90 met een grote inspiratie vanuit mac os9.

Er zijn meerdere iconen linksonder op de pagina. Deze iconen, waneer geklikt, openen een window die een bepaald iets laten zien:

* Folder: wanneer geopend, een window wordt getoond met meerdere bestanden erin die op zichzelf ook geopend kunnen worden.
* Afbeelding: wanneer geopend, een window wordt getoond waarin de afbeelding die alleen als preview in het icoon als de gehele afbeelding getoond wordt.
* Audio: wanneer geopend, een muziekspeler window wordt geopend en begint met het afspelen van het audiobestand. In het geval dat de speler al open is wordt het spelende bestand afgebroken en vervangen met het nieuwe bestand.
* Video: wanneer geopend, op dezelfde manier als de audiobestanden, wordt er een videospeler geopend en wordt de video getoond. Dezelfde regels zijn van toepassing als bij de muziekspeler.
* Internet browser: wanneer geopend, een internet browser wordt getoond die dezelfde requirements volgt als internet browsers van die tijd (www. En http/https moeten erin bijvoorbeeld)

Elke window kan gesleept worden en gesloten worden.

# Database model



Op de website kun je een account aanmaken, waarmee je toegang krijgt tot meerdere dingen, je kan comments achterlaten op de portfolio pagina’s en in een gastboek schrijven. Dit zullen andere gebruikers ook zien als zij op de pagina komen (onafhankelijk van of zij ingelogd zijn of niet)

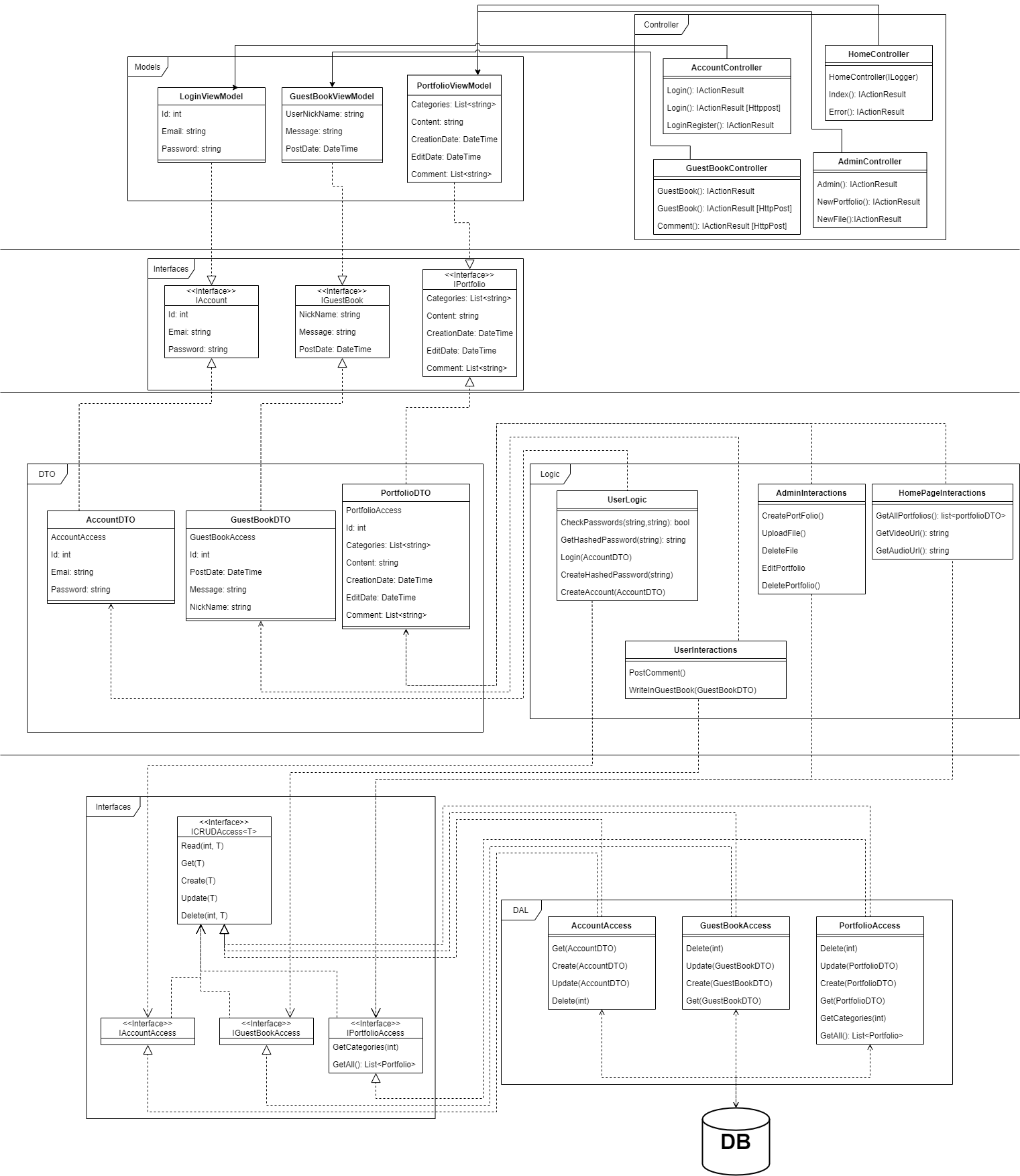
Als de gebruiker een account aanmaakt moet diegene een email adres, wachtwoord en Gebruikersnaam invullen. De gebruikersnaam zal gebruikt worden bij zowel de comments op de portfolio pagina’s en in het gastboek.

Een gebruiker kan meerdere malen in het gastboek schrijven (bijvoorbeeld als iemand meerdere malen naar hetzelfde hotel gaat en daar in het gastboek schrijft) waarbij er een nieuw item wordt toegevoegd aan de database “GuestBook” met een uniek Id, de gebruikersnaam, wat voor bericht ze hebben geschreven en wanneer ze dit hebben gedaan.

Een gebruiker kan ook op portfolio pagina’s comments achterlaten om bijvoorbeeld een vraag te stellen over het onderwerp van die pagina, hierbij wordt er een nieuw item toegevoegd aan de Comment tabel. Hierbij wordt de gebruikersnaam gebruikt samen met wat de gebruiker getypt heeft, verder wordt ook het id van de pagina erbij gevoegd om zo de correcte comments op de correcte paginas te zetten.

Elke pagina heeft zijn eigen “content” (de tekst en media op de pagina) samen met één of meerdere categorieën, een creatie datum, de laatste aanpassing’ s datum en de comments. De categorieën hebben een eigen id en naam die ervoor zorgt dat een gebruiker op categorie kan sorteren om zo onderwerp gerelateerde portfolio pagina’s te kunnen vinden en lezen.

# Klassen diagram



Ik heb zowel een DTO als viewmodel met de reden dat de DAL laag en de view laag apart blijven en dat ik niet vast zit met de viewmodel resultaat, deze worden aan elkaar gelijk gesteld door middel van een interface die ertussenin zit. Dit was ook mogelijk door het gebruik maken van iets zoals een automapper. Dit heb ik niet gedaan omdat het in het geheel voor meer moeite en grotere code zal zorgen naar mate ik bijvoorbeeld een portfolio pagina vanuit de front-end aan wil maken.

De logica laag is met de DAL laag verbonden doormiddel van interfaces om er zo voor te zorgen dat de logica onafhankelijk is van de database zelf, hierdoor kan de database vervangen worden met een andere zonder dat de gehele code daarvoor hoeft aangepast te worden.