

MVCLibraryApp

DOCUMENTATIE VOOR HET ONTWIKKELEN VAN DE
MVCLIBRARYAPP APPLICATIE

Robin Kantier, Quinten Schaap (studenten)
WINDESHEIM ALMERE, ADSD1

Inhoud

1.0	Missie	2
1.1	Achtergrond	2
1.2	Doelgroep	2
1.3	Stakeholders	2
2.	Visie.....	3
2.1	Randvoorwaarden	3
2.1.1	Functioneel.....	3
2.1.2	Technisch	4
2.2	Vraagstelling.....	4
2.3	Probleemstelling	5
2.3.1	afbakening.....	5
2.3.2	Binnen de scope.....	5
2.3.3	buiten de scope	5
2.4	Doelstellingen	5
3.	Use cases	6
4.	ontwerp.....	7
4.1	Activiteitendiagrammen	7
4.1.1	Bezoeker	8
4.1.2	Medewerker	8
4.1.3	Beheerder	9
4.2	Sequentiediagrammen	10
4.3	ERD diagram	11
4.4	Klassendiagram.....	12
4.5	Wireframe.....	13
4.5.1	Homepage	13
4.5.2	LoginPage	14
4.5.3	Registratie Pagina.....	15
5.	Resultaten.....	17
6.	Kwaliteitscriteria	17
7.	Acceptatiecriteria	17

1.0 Missie

De missie van dit project is het creëren van een functionele bibliotheek app waarin een gebruiker een boek kan lenen, terugbrengen en een boete kan krijgen bij achterstallig retourneren. Gebruik makend van .net 7 en een C# webapp met het MVC pattern.

1.1 Achtergrond

Voor Software Development 2 worden de studenten uitgedaagd om een functionele web applicatie te bouwen, met als doel een app waarin gebruikers gemakkelijk boeken kunnen inzien en reserveren, medewerkers items kunnen uitlenen en innemen, en nieuwe boeken in het systeem kunnen zetten, en rechten voor het aanpassen van gebruikers. Beheerders dienen alles te kunnen inzien, en veranderen of blokkeren.

1.2 Doelgroep

De doelgroep voor dit project is de gebruikers van de bibliotheken, zowel als bibliotheken om dit product te implementeren.

1.3 Stakeholders

De stakeholders van dit project zijn:

- Hogeschool Windesheim Flevoland (80%*)
De opdracht is vergeven door het Windesheim, en is uitgevoerd ter behalen van de opdracht van periode 4. Alle rechten van het opgeleverde product zijn vrij om gebruikt te worden door het Windesheim voor eventuele verlening van diensten in sectoren die toepasselijk zijn op dit product.
- Robin Kantier (10%) (Student-Developer)
- Quinten Schaap (10%)(Student-Developer)

**Aandeelpercentage project LLibraryApp*

2. Visie

De visie van de LibraryApp is om het proces van het bouwen van een webapplicatie door studenten te laten doorlopen, en om een geautomatiseerd systeem op te zetten voor het beheren items van de bibliotheek.

2.1 Randvoorwaarden

De volgende randvoorwaarden zijn gegeven in het opdrachtsformulier:

De applicatie moet te bouwen en runnen zijn zonder ingreep van de docenten.

De applicatie moet voldoen aan alle functionele en technische eisen. Namelijk:

2.1.1 Functioneel

Bezoekers mogen:

- Items en Authors inzien en zoeken op: titel, auteur, locatie, etc.
- Filteren op: auteur, jaar van uitgave, locatie, etc.
- Items reserveren.
- Eigen leningen en reserveringen beheren.

Medewerkers mogen:

- Alles wat een Bezoeker mag.
- Items uitlenen en innemen.
- Items en Authors aanmaken en bewerken.
- Alle leningen en reserveringen beheren.
- Volledige CRUD op Users met het type/level/rechten van Bezoeker.
- Openstaande posten (facturen/openstaand bedrag) inzien.
- Betalingen verwerken (niet de betaalmethoden zelf, maar puur de registratie → er is betaald, betaalmethoden vallen buiten de scope van deze opdracht).
- Bezoeker accounts blokkeren.
- Locaties aan de Items toekennen.
- Kan abonnementen toevoegen en annuleren.

Beheerders mogen:

- Alles wat een Medewerker mag.
- Items en Authors verwijderen.
- Volledige CRUD op Users met alle type/level/rechten.
- Alle accounts blokkeren.
- Locaties toevoegen (denk aan Locatie Noord Kast 11 plank 3).

2.1.2 Technisch

- Maak een ASP.NET Core Web App (Model-View-Controller) en pas het MVC design pattern toe.
- Gebruik .NET 7.0
- Gebruik https i.p.v. http
- Gebruik de onderstaande connection string waarbij S0000000 vervangen wordt door het bovenste student nummer in Program.cs
`"Server=(localdb)\\mssqllocaldb;Database=BibliotheekS0000000;Trusted_Connection=True;MultipleActiveResultSets=true"`
- Volg de werkwijze uit de colleges en pas deze, en eventuele uitbreidingen, in dezelfde gedachten toe bij het ontwikkelen van deze applicatie.
- De applicatie gebruikt een database met Entity Framework.
- Gebruik geen "Razor pages", maar wel "Razor".
- Er zijn Migrations: De database moet door een ander te installeren zijn door Update-Database uit te voeren.
- Er is Seeding-Data (niet test1, test2 etc., maar wel enigszins logische data) en minimaal 100 records per tabel (abonnementen en locaties uitgezonderd). Maak gebruik van een Faker.
- Alle code moet voldoen aan de Microsoft coding conventions.
<https://learn.microsoft.com/en-us/dotnet/csharp/fundamentals/codingstyle/coding-conventions> Met uitzondering van: Use parentheses to make clauses in an expression apparent. (Niet verplicht om het zo te doen, maar mag wel)
- Voorkom magic numbers (hardcoded numbers) en gebruik in plaats daarvan constanten of Enums
- Enum types gebruiken waar logisch.
- Voorzie een veilig inlogsysteem gebaseerd op username/e-mail en password. Gebruik hiervoor de ingebouwde "Authentication Type" met de optie "Individual accounts".

2.2 Vraagstelling

1. Wat zijn effectieve manieren om rechten te verlenen op verschillende geauthenticeerde accounts, en welke libraries zijn hiervoor te gebruiken?
2. Hoe zorg je voor een HTTPS verbinding, en pas je dit toe in je MVC webapp?
3. Wat is razor, en hoe past het project team dit effectief toe.?
4. Welke vereisten staan er in de coding conventions, en hoe is dit van toepassing op het project?
5. Wat zijn de voordelen van .net 7?

2.3 Probleemstelling

Het huidige bibliotheekstelsel mist een geïntegreerd platform dat in staat is om tegemoet te komen aan de behoeften van zowel bezoekers als personeel. Deze applicatie moet het mogelijk maken om boeken en andere media te zoeken en te reserveren, leningen en reserveringen te beheren, en betalingen te verwerken voor diverse kosten zoals abonnementen, uitgeleende materialen, reserveringskosten, boetes, etc. Het stelsel moet ook een rolgebaseerde toegangscontrole implementeren om de toegang en acties van gebruikers te beperken op basis van hun rol.

2.3.1 afbakening

Dit project zal zich alleen richten op de ontwikkeling van de applicatie en zal niet betrokken zijn bij de implementatie of het onderhoud van de server- en netwerkinfrastructuur die nodig is om de applicatie te hosten.

2.3.2 Binnen de scope

1. Ontwikkeling van de ASP.NET Core Web App (MVC) applicatie.
2. Implementatie van de functies zoals vermeld in de projectvereisten.
3. Testen van de applicatie voor gebruik.
4. Documentatie van het ontwikkelingsproces en de uiteindelijke applicatie.

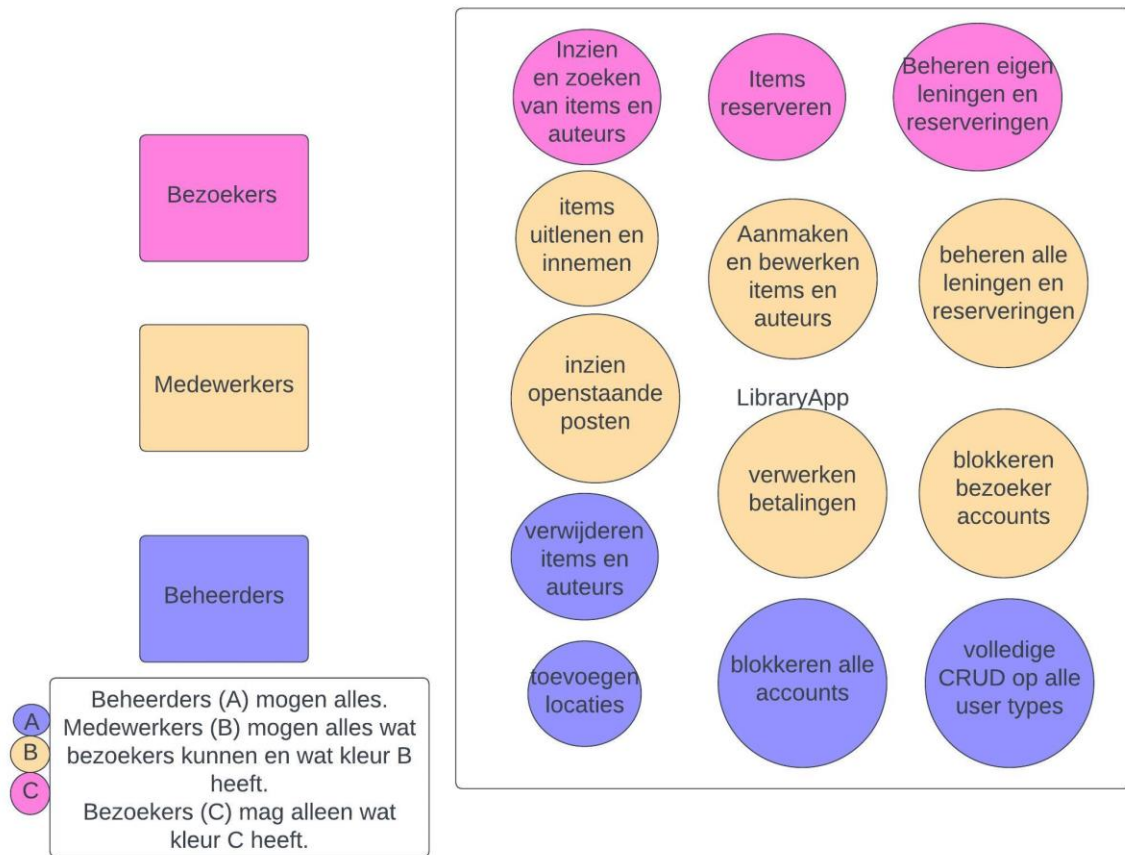
2.3.3 buiten de scope

1. Het installeren en onderhouden van de serverinfrastructuur.
2. Training van bibliotheekmedewerkers en gebruikers in het gebruik van het stelsel.
3. Langdurig onderhoud en updates na de inleverdatum van het project.

2.4 Doelstellingen

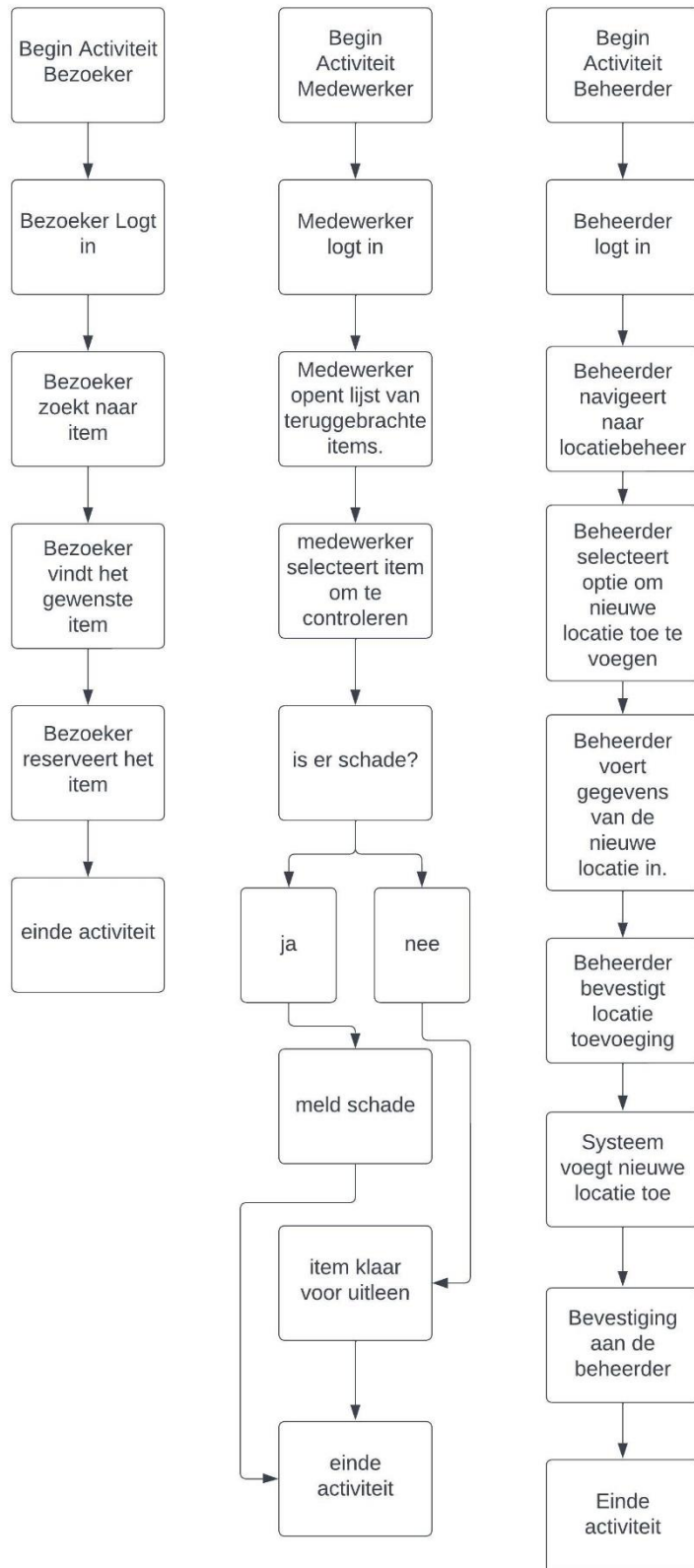
1. Het ontwikkelen van een gebruiksvriendelijke bibliotheekbeheerapplicatie.
2. De applicatie moet de gebruikers in staat stellen om materialen te zoeken, reserveringen te maken en hun leningen en reserveringen te beheren.
3. De applicatie moet het personeel in staat stellen om alle bibliotheekgegevens te beheren, waaronder uitleengegegevens en inventarisbeheer.
4. De applicatie moet een veilig inlogsysteem hebben en een role-based access control systeem.
5. De applicatie moet voldoen aan alle technische en functionele vereisten zoals vermeld in de projectopdracht.
6. Het succesvol opleveren van de applicatie voor de gestelde deadline.

3. Use cases



4. ontwerp

4.1 Activiteitendiagrammen



4.1.1 Bezoeker

1. Begin Activiteit: startpunt van het diagram.
2. Bezoeker Logt In: Dit is de eerste stap die de Bezoeker uitvoert. In UML wordt dit weergegeven als een rechthoek met afgeronde randen (een activiteit).
3. Bezoeker Zoekt naar een Item: De volgende stap voor de Bezoeker is het zoeken naar een specifiek Item.
4. Bezoeker Vindt het Gewenste Item: De Bezoeker vindt het Item dat hij/zij wil lenen.
5. Bezoeker Reserveert het Item: De Bezoeker reserveert het Item via de interface van de applicatie.
6. Systeem Bevestigt de Reservering: Het systeem bevestigt de reservering en informeert de Bezoeker. Dit kan bijvoorbeeld een bericht zijn op de website of een e-mail naar de Bezoeker.
7. Einde Activiteit: eindpunt van het diagram.

4.1.2 Medewerker

1. Begin Activiteit: Startpunt van het diagram.
2. Medewerker Logt In: De Medewerker logt in op het systeem.
3. Medewerker Opent Lijst van Teruggebrachte Items: De Medewerker navigeert naar de lijst van recent teruggebrachte Items.
4. Medewerker Selecteert Item om te Controleren: De Medewerker kiest een Item uit de lijst om te controleren op eventuele schade.
5. Medewerker Controleert Item: De Medewerker inspecteert het Item op fysieke schade.
6. Is er Schade?: Een beslissingspunt (diamant) waar de stroom van controle splitst.

Ja: Als er schade is, gaat de Medewerker naar stap 7.

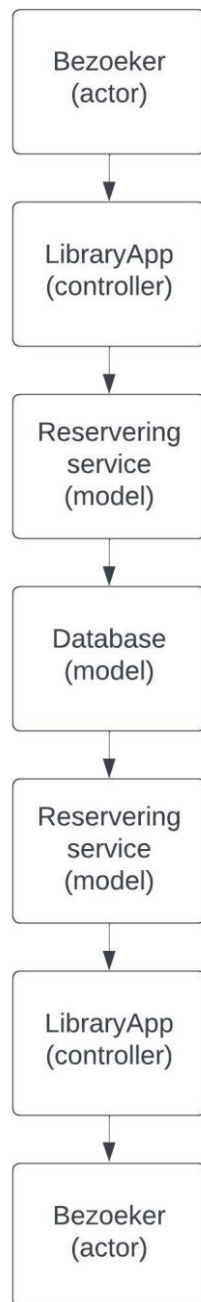
Nee: Als er geen schade is, gaat de Medewerker naar stap 8

7. Meld Schade: Als er schade is, noteert de Medewerker de schade in het systeem en informeert de Bezoeker over eventuele boetes. Ga daarna naar stap 9.
8. Item is Klaar voor Uitleen: Als er geen schade is, update de Medewerker de status van het Item in het systeem om aan te geven dat het klaar is om opnieuw uitgeleend te worden. Ga daarna naar stap 9.
9. Einde Activiteit: Eindpunt van het diagram.

4.1.3 Beheerder

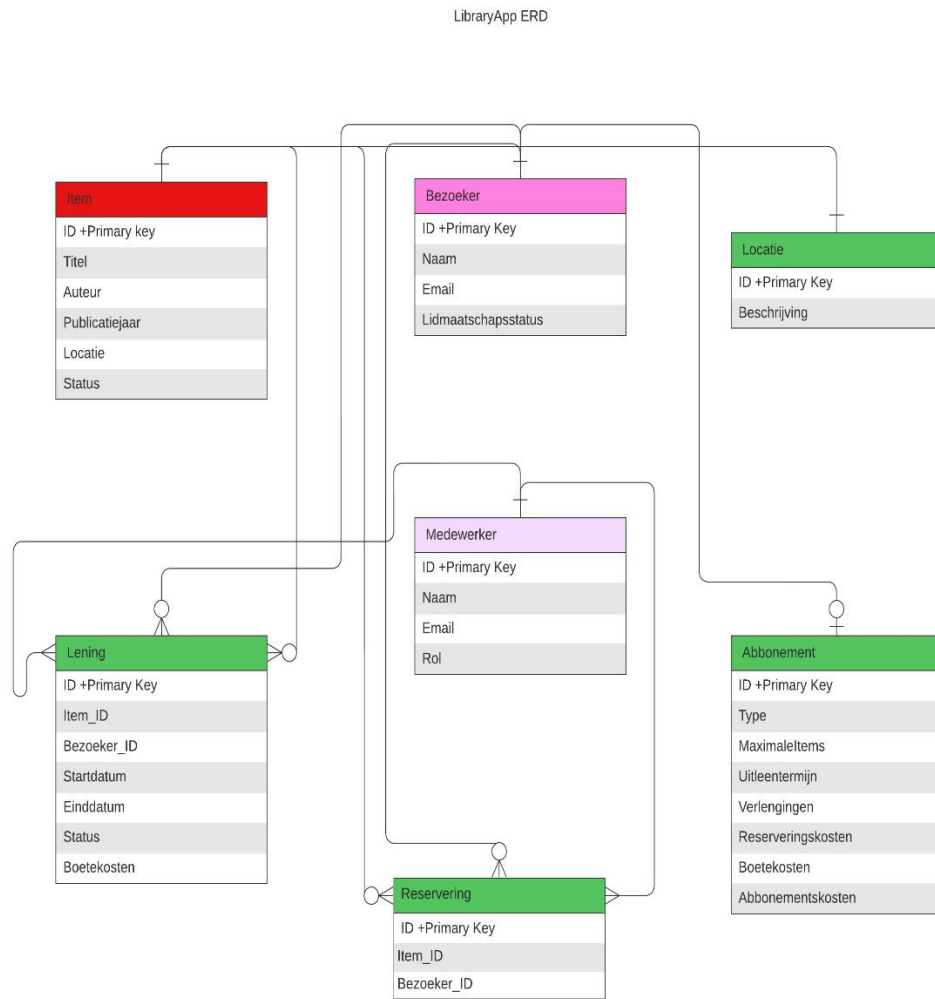
1. Begin Activiteit: Startpunt van het diagram
2. Beheerder Logt In: De Beheerder logt in op het systeem.
3. Beheerder Navigeert naar Locatiebeheer: De Beheerder navigeert naar het gedeelte van de applicatie dat zich bezighoudt met locatiebeheer.
4. Beheerder Selecteert Optie om Nieuwe Locatie Toe te Voegen: De Beheerder selecteert de optie om een nieuwe locatie toe te voegen.
5. Beheerder Voert Gegevens van de Nieuwe Locatie In: De Beheerder vult het formulier in met de details van de nieuwe locatie, zoals de naam, de verdieping, de rij en het nummer.
6. Beheerder Bevestigt Locatie Toevoeging: Na het controleren van de gegevens, bevestigt de Beheerder de toevoeging van de nieuwe locatie.
7. Systeem Voegt Locatie Toe: Het systeem verwerkt de aanvraag en voegt de nieuwe locatie toe aan de database.
8. Bevestiging aan de Beheerder: Het systeem stuurt een bevestiging naar de Beheerder dat de nieuwe locatie met succes is toegevoegd.
9. Einde Activiteit: Eindpunt van het diagram.

4.2 Sequentiediagrammen

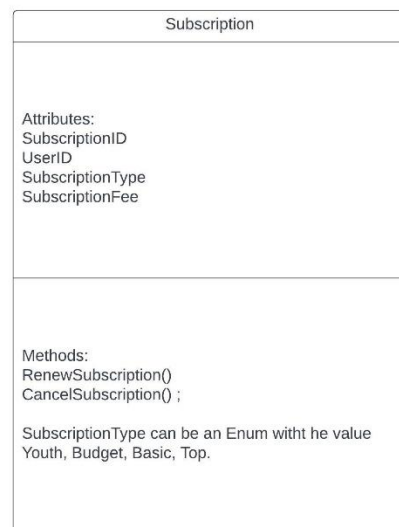
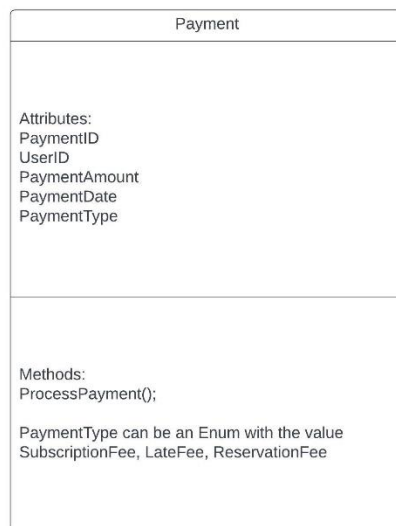
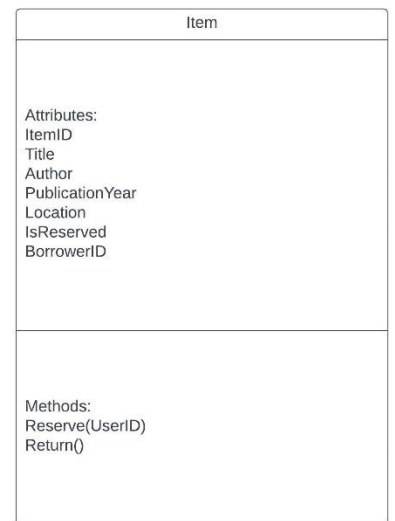
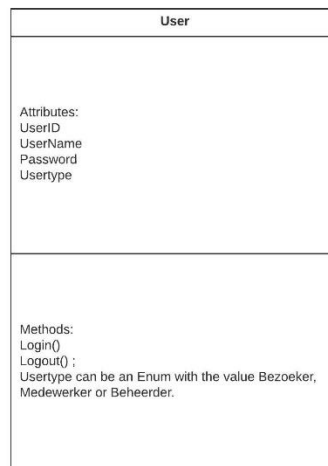
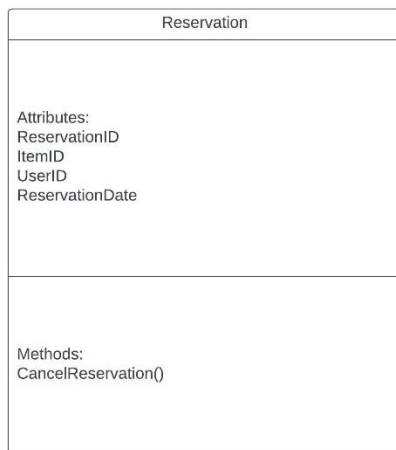


1. Bezoeker (Actor): Startpunt van het diagram. De Bezoeker kiest een boek om te reserveren.
2. Web App (Controller): De interactie begint wanneer de Bezoeker een boek selecteert om te reserveren. De Bezoeker stuurt een bericht naar de Web App met het verzoek om een reservering te maken.
3. Reservering Service (Model): De Web App stuurt een bericht naar de Reservering Service met het verzoek om een nieuwe reservering te maken.
4. Database (Model): De Reservering Service communiceert met de Database om de reservering op te slaan. De Database stuurt een bevestiging terug naar de Reservering Service dat de reservering is opgeslagen.
5. Reservering Service (Model): Zodra de Reservering Service de bevestiging van de Database ontvangt, stuurt deze een bericht terug naar de Web App met de status van de reservering.
6. Web App (Controller): De Web App ontvangt het bericht van de Reservering Service en stuurt een bevestiging naar de Bezoeker dat de reservering is gemaakt.
7. Bezoeker (Actor): De Bezoeker ontvangt de bevestiging van de reservering.

4.3 ERD diagram



4.4 Klassendiagram



4.5 Wireframe

4.5.1 Homepage

LibraryApp

LoginRegistreer

Zoek hier uw Boek of Auteur:

Over LibraryApp

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed id risus nec neque fringilla vestibulum vel nec massa. Vivamus justo lectus, lacinia non interdum ut, gravida in elit. Cras id placerat libero. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse potenti. Sed hendrerit lectus non lorem lobortis, malesuada efficitur nisi malesuada. Morbi nec vestibulum nulla. Fusce rhoncus, sem nec fringilla lobortis, mi dolor luctus est, quis porta nunc tortor non diam. Cras congue vel mi vitae pretium. Donec vel mi sit amet nunc viverra vehicula quis pellentesque ligula. Cras vulputate, enim vitae eleifend aliquet, sapien velit porttitor ligula, a malesuada sapien leo in sem. Vestibulum lacinia justo non quam molestie vestibulum. Phasellus a tortor volutpat, blandit dui ut, luctus erat. Fusce vehicula nec nunc at maximus. Suspendisse potenti. In non dui non ligula egestas posuere quis quis orci.

Registreer nu!

vestibulum. Phasellus a tortor volutpat, blandit dui ut, luctus erat. Fusce vehicula nec nunc at maximus. Suspendisse potenti. In non dui non ligula egestas posuere quis quis orci.

4.5.2 LoginPage

LibraryApp

Registreer

Gebruikersnaam:

Wachtwoord:

[Wachtwoord vergeten?](#)

4.5.3 Registratie Pagina

LibraryApp

Login

Registreer nu!

Naam:

Email:

Adres:

Telefoonnummer:

Registreer!

[al een account?](#)

4.5.4 Gebruikersdashboard

LibraryApp

Logout

Zoek naar uw boek!

Zoek!

Overzicht Boeken:

Nieuw Auteur*

Nieuw Item*

reserveer!

Leen uit*

Neem in*

Delete**

Uw Reserveringen & Leningen:

beheer

Einde Bezoekersrechten

Alle openstaande leningen & reserveringen:

Alle Bezoekers Accounts:

Blokkeer

Einde Medewerkersrechten

Alle Gebruikers Accounts:

Create

Update

Delete

Massablok

5. Resultaten

Het doel van dit project is het opleveren van een functionele bibliotheekapplicatie gebouwd met ASP.NET Core, volgens de MVC architectuur, die voldoet aan de functionele en technische vereisten. De applicatie zal zowel voor bezoekers, medewerkers als beheerders verschillende functionaliteiten bieden. De applicatie zal naadloos integreren met een database voor het beheren van gebruikers, items, reserveringen en betalingen. Tot slot zal de applicatie een gedetailleerde log van activiteiten en transacties bijhouden voor beoordeling en rapportage.

6. Kwaliteitscriteria

De kwaliteit van het project zal worden bepaald door de volgende criteria:

- **Functionaliteit:** De applicatie moet alle voorgeschreven functies bevatten, waaronder gebruikersbeheer, itembeheer, reserveringen en betalingsverwerking.
- **Betrouwbaarheid:** De applicatie moet stabiel zijn en zonder fouten functioneren. De applicatie moet ook goed presteren onder belasting.
- **Gebruiksgemak:** De applicatie moet intuïtief en gemakkelijk te gebruiken zijn voor alle gebruikerscategorieën (bezoekers, medewerkers, beheerders).
- **Veiligheid:** De applicatie moet veilige authenticatie en autorisatiemechanismen bevatten. Gebruikersgegevens en transacties moeten beveiligd zijn tegen ongeoorloofde toegang.
- **Codekwaliteit:** De code van de applicatie moet leesbaar, onderhoudbaar en goed gestructureerd zijn. De code moet voldoen aan de Microsoft coding conventions.

7. Acceptatiecriteria

De acceptatiecriteria voor het project zijn als volgt:

- De applicatie bevat alle benodigde functionaliteiten zoals omschreven in de functionele en technische vereisten.
- De applicatie functioneert zonder fouten en crashes.
- De applicatie presteert goed onder belasting en is in staat om een groot aantal gelijktijdige gebruikers te ondersteunen.
- De gebruikersinterface is gebruiksvriendelijk en intuïtief.
- De applicatie heeft robuuste veiligheidsmaatregelen geïmplementeerd, inclusief veilige authenticatie en autorisatie.
- De code van de applicatie is leesbaar, goed gestructureerd en volgt de Microsoft coding conventions.
- De applicatie is volledig gedocumenteerd, inclusief commentaar in de code en een uitgebreid design document.