**Ducomentaties**

**User Stories**

1. Als medewerker wil ik kunnen inloggen op het studentenhuisvestingssysteem, zodat ik toegang heb tot de beheerfunctionaliteiten.
2. Als medewerker wil ik kunnen uitloggen uit het studentenhuisvestingssysteem, zodat mijn account veilig blijft wanneer ik de website verlaat.
3. Als medewerker wil ik studenten kunnen toevoegen aan de huisvestingslijst, inclusief hun persoonlijke gegevens en voorkeuren.
4. Als medewerker wil ik de gegevens van studenten kunnen wijzigen in de huisvestingslijst, bijvoorbeeld wanneer er veranderingen zijn in hun situatie.
5. Als medewerker wil ik studenten kunnen verwijderen uit de huisvestingslijst, indien nodig.
6. Als medewerker wil ik beschikbare woningen kunnen toevoegen aan het systeem, inclusief details zoals locatie, grootte en huurprijs.
7. Als medewerker wil ik de gegevens van beschikbare woningen kunnen wijzigen, bijvoorbeeld wanneer er updates zijn over de beschikbaarheid.
8. Als medewerker wil ik woningen kunnen verwijderen uit de lijst van beschikbare woningen, bijvoorbeeld wanneer deze niet langer beschikbaar zijn.
9. Als medewerker wil ik studenten kunnen koppelen aan beschikbare woningen, zodat ik studenten kan helpen bij het vinden van huisvesting.
10. Als medewerker wil ik een overzicht kunnen oproepen van de huisvestingssituatie van een specifieke student, inclusief de woning waaraan ze zijn gekoppeld.
11. Als medewerker wil ik gegevens van huisvestingsaanbieders kunnen toevoegen aan het systeem, inclusief contactinformatie en beschikbare woningen.
12. Als medewerker wil ik de gegevens van huisvestingsaanbieders kunnen wijzigen, bijvoorbeeld wanneer er updates zijn over hun beschikbare woningen.
13. Als medewerker wil ik huisvestingsaanbieders kunnen verwijderen uit het systeem, indien nodig.
14. Als medewerker wil ik studenten kunnen koppelen aan huisvestingsaanbieders, zodat ik studenten kan plaatsen in beschikbare woningen.
15. Als medewerker wil ik een overzicht kunnen maken van geplaatste studenten, inclusief de woning waarin ze zijn geplaatst.
16. Als medewerker wil ik gegevens van een medewerker kunnen toevoegen aan het systeem, zoals naam, gebruikersnaam en wachtwoord.
17. Als medewerker wil ik gegevens van een medewerker kunnen wijzigen, bijvoorbeeld wanneer er veranderingen zijn in hun functie of contactgegevens.
18. Als medewerker wil ik gegevens van een medewerker kunnen verwijderen uit het systeem, bijvoorbeeld bij het vertrek van een medewerker.

**UML Class Diagram**

**Medewerker**

* **Methoden**:
  + inloggen(): void
  + uitloggen(): void
  + studentToevoegen(student: Student): void
  + studentWijzigen(student: Student): void
  + studentVerwijderen(studentId: int): void
  + woningToevoegen(woning: Woning): void
  + woningWijzigen(woning: Woning): void
  + woningVerwijderen(woningId: int): void
  + studentKoppelenAanWoning(studentId: int, woningId: int): void
  + studentHuisvestingOverzicht(studentId: int): void
  + aanbiederToevoegen(aanbieder: HuisvestingsAanbieder): void
  + aanbiederWijzigen(aanbieder: HuisvestingsAanbieder): void
  + aanbiederVerwijderen(aanbiederId: int): void
  + studentKoppelenAanAanbieder(studentId: int, aanbiederId: int): void
  + geplaatsteStudentenOverzicht(): void
  + medewerkerToevoegen(medewerker: Medewerker): void
  + medewerkerWijzigen(medewerker: Medewerker): void
  + medewerkerVerwijderen(medewerkerId: int): void

**HuisvestingsLijst**

* **Attributen**:
  + studenten: List<Student>
  + woningen: List<Woning>
* **Methoden**:
  + toevoegenStudent(student: Student): void
  + wijzigenStudent(student: Student): void
  + verwijderenStudent(studentId: int): void
  + toevoegenWoning(woning: Woning): void
  + wijzigenWoning(woning: Woning): void
  + verwijderenWoning(woningId: int): void
  + studentKoppelenAanWoning(studentId: int, woningId: int): void
  + huisvestingOverzicht(studentId: int): void

**Database**

**Medewerker Table**

CREATE TABLE Medewerker (

medewerkerId INT PRIMARY KEY,

gebruikersnaam VARCHAR(255),

wachtwoord VARCHAR(255)

);

**Student Table**

CREATE TABLE Student (

studentId INT PRIMARY KEY,

naam VARCHAR(255),

leeftijd INT,

email VARCHAR(255),

telefoonnummer VARCHAR(255),

voorkeuren VARCHAR(255)

);

**Woning Table**

CREATE TABLE Woning (

woningId INT PRIMARY KEY,

locatie VARCHAR(255),

grootte INT,

huurprijs DECIMAL(10, 2)

);

**HuisvestingsAanbieder Table**

CREATE TABLE HuisvestingsAanbieder (

aanbiederId INT PRIMARY KEY,

naam VARCHAR(255),

contactgegevens VARCHAR(255)

);

**Huisvesting Table**

CREATE TABLE Huisvesting (

huisvestingId INT PRIMARY KEY,

studentId INT,

woningId INT,

aanbiederId INT,

status VARCHAR(255),

huurcontract VARCHAR(255),

FOREIGN KEY (studentId) REFERENCES Student(studentId),

FOREIGN KEY (woningId) REFERENCES Woning(woningId),

FOREIGN KEY (aanbiederId) REFERENCES HuisvestingsAanbieder(aanbiederId)

);