Designklassediagram

Vi har anvendt trelagsarkitektur, som betyder at vi har delt systemet op i tre dele. De hedder hhv. Kontrollaget, modellaget og databaselaget.

Trelagsarkitektur gør det nemt at vedligeholde, skalerer og udvikle på senere hen. Hvert lag kan udvikles og testes uafhængigt af hinanden. Det gør det markant mere fleksibelt samt at man kan genbruge det. Ydermere gør det også systemet mere sikkert og reducerer de løbende omkostninger.

**Kontrollaget:**

Det er dette lag som sørger for at input fra brugeren igennem UI og modellaget. Den fortæller hvor det given input skal sendes hen, og hvilken operation som skal udføres.

A green screen with black text

Description automatically generated

Ser man på ShipmentController, så koordinerer den oprettelsen og styringen af shipments.

Det gør den ved at brugeren kan oprette shipments igennem createShipment. Her associereres den med en medarbejder igennem employeeNo.

Den kan også tilføje varer til en eksisterende shipment igennem additemLine med en barcode.

Til sidst kan der tilføjes en part, det kan være en kunde eller et firma, til en shipment igennem den sidste metode addPartyToShipment.

**Modellaget:**

I modellaget finds alt logikken bag systemet. Det er her kontrollaget sender de operationer hen som brugeren vælger i kontrollaget. Laget sørger for indkapsling således at brugeren ikke kan tilgå data som den ikke skal.

A screenshot of a computer

Description automatically generated

Shipment Klassen i modellaget har nogle forskellige attributer. Men det mest interessant er dens metoder, som sørger for, at dens data bliver indkapslet. Med data menes der det relevante data for shipment så som modtagere, datoer og varer.

**Databaselaget**

Databaselaget gør, som navnet antyder, at det relevante information for databasen bliver indkapslet her. Det er at logik som vedrører querying, retrieving, saving osv.