**Opgave 1**

1. **Hvad bruges ækvivalensklasser til i forbindelse med test?**

Ækvivalensklasser er en måde at opdele tests på. Alle tests i en given ækvivalensklasse accepterer/fejler sammen, da hvert element i ækvivalensklassen tester den samme del af et system. Man kan med fordel bruge ækvivalensklasser til stærkt at reducere antallet af overflødige tests.

1. **Hvad betyder det at et system har høj kobling?**

Hvis et system har høj kobling, så har de interne komponenter mange forbindelser til hinanden (høj afhængighed mellem moduler), hvilket ofte er overflødigt og bare gør systemet mere kompliceret end nødvendigt. Høj kobling resulterer også i, at det er svært at debugge, maintaine og udvikle nye features til et system.

|  |  |
| --- | --- |
| **Navn:** | Tilføj Vare |
| **Mål:** | Bruger ønsker at tilføje vare til SmartFridge |
| **Aktører:** | Primær: Bruger  Sekundær: BCDB |
| **Antal samtidige forekomster:** | 1 |
| **Prækondition:** | SmartFridge tændt og forbindelse til BCDB |
| **Postkondition:** | Brugeren har tilføjet en vare til SmartFridge |
| **Hovedscenarie:** | 1. Bruger sætter varens stregkode op mod stregkodeskanneren. 2. SmartFridge skanner stregkoden. 3. SmartFridge sender stregkoden til BCDB. 4. BCDB tjekker om stregkoden er i databasen. 5. BCDB returnerer varens navn til SmartFridge    1. [Extension 1] Vare findes ikke i database. 6. Bruger vælger antallet af varen der skal tilføjes.    1. [Extension 2] Bruger afbryder tilføjelsen. 7. SmartFridge tilføjer vare til indkøbslisten. 8. SmartFridge viser den nye indkøbsliste. |
| **Udvidelser/undtagelser:** | [Extension 1]   1. BCDB returnerer fejlkode til SmartFridge. 2. SmartFridge anmoder om indtastning af varens navn. 3. Brugeren indtaster varens navn. 4. SmartFridge sender navnet og stregkoden til BCDB. 5. UC forsættes ved trin 6 i hovedscenariet.   [Extension 2]   1. SmartFridge stopper tilføjelse af vare og returner til idle-stadiet. UC afsluttes. |