

# Peer Review Planka Gbg

- **Do the design and implementation follow design principles?**
  - **Does the project use a consistent coding style?**

Vissa metodkroppars "måsvingar" { } startar på ny rad medan andra startar på samma rad som metod-deklarationen. Kan tyckas petigt men försök använd samma stil.
  - **Is the code reusable?**

Ja koden är återanvändbar, det finns en tydlig modularitet som tillåter för expanderings av funktioner och informationsflöde. Det ser ut att vara lätt att lägga till busslinjer eller båtlinjer utan att behöva ändra i koden. Så länge man fyller i textfilen på rätt sätt så läser programmet in det.
  - **Is it easy to maintain?**

Cirkelberoendet hos node-network-route-station kan ställa till med problem om man vill ändra någonting där.
  - **Can we easily add/remove functionality?**

Ja det verkar vara lätt att både kunna lägga till och ta bort funktionalitet utan att man behöver göra stora ändringar i befintlig kod.
  - **Are design patterns used?**

MVC - Modellen är separerad från view och controller på ett bra och tydligt sätt.

Observer - Modellen är observable vilket gör att den inte beror på view eller controller vilket är korrekt och bra.

Command-query separation - Vad vi kan se följs denna princip bra genomgående i koden.
- **Is the code documented?**

De flesta stora klasserna är väldokumenterade, medan andra saknar kommentarer.
- **Are proper names used?**

Överlag väldigt bra metodnamn, men vissa variabelnamn inne i metoderna är väldigt otydliga, i till exempel makeStationReport i MODEL är variabelnamnen "n", "r", "i" och "s". Namnet på listan som skapas i

metoden `addEdge` i `Network`-klassen hade kunnat haft ett tydligare namn än `abc`, så att man får en bättre förståelse för vad det är för lista och vad den används till.

- **Is the design modular? Are there any unnecessary dependencies?**

Det finns en inre klass i `Network` som heter `Node`, denna klass används av väldigt många klasser vilket skapar ett cirkulärt beroende mellan vissa klasser. Ett par stora metoder skulle kunna brytas ned till mindre metoder.

- **Does the code use proper abstractions?**

Ja, t ex `abstractReport`, `Node` och `Observable` används som abstraktioner.

- **Is the code well tested?**

Vi kan just nu inte köra testerna och kan därför inte avgöra om de går igenom. Vi tycker däremot att testerna är krångligt skrivna, det är för många deltester i en och samma testmetod utan tydliga namn/kommentarer och det testas för många olika saker i samma metod.

- **Are there any security problems, are there any performance issues?**

Användare kan ladda upp olämpliga bilder.

- **Is the code easy to understand? Does it have an MVC structure, and is the model isolated from the other parts?**

Koden är i stort lätt att förstå både genom metodnamn och dokumentation med små otydligheter med val av variabelnamn inne i vissa metoder. Applikationen följer MVC och modellen är avgränsad från V och C.

- **Can the design or code be improved? Are there better solutions?**

Samlingen `if`-satser i `onClick`-metoden i `Mainactivity` kan skrivas med `“else if”` istället, så att inte varje sats körs igenom trots att rätt alternativ hittats.

Modellen borde vara en singleton så att man inte råkar skapa flera instanser av modellen så att inte alla användare får alla rapporter.