**Krav1:**

**Arkitektur**

Startpunkten för appen är App.tsx.

Appen inkluderar en Tab.Navigation med två huvudsakliga delar: Delays och Log in.

Vyer byggs utav komponenter i /components. Components mappar har en huvudkomponent som därefter använder sig av andra subkomponenter. Tex DelaysStack innehåller en StackNavigatior, som visar upp StationsScreen och DelaysScreen.

Utöver det finns det ett modell-lager. Modeller skickar anrop till trafik API:et, autentiserings-API, storage för tokens. Jag delar upp apiData och delays i två moduler eftersom första hätar rådata från delays APIet och den andra finns mer för att bearbeta datat.

Interfaces: jag valde att behålla endast auth interface. Överlag är koden i appen inte typad, men som ett förbättringsförslag skulle jag utveckla det.

Stylingen genom appen är genomfört med hjälp av styling moduler i /styles. Importerade typsnitt ligger i /assets/fonts.

Teknologier:

* React-native maps
* Expo-location
* React Navigation Native Stack
* React Navigation material bottom tabs
* React Native Flash Message
* React Native Paper
* React Native
* Expo SplashScreen

**Planering**

Jag planerade arbetet som en kanban board och gjorde en utförlig plan av alla vyer i förväg. Med agilutveckling i tanken försökte jag under utvecklings gång prioritera grundkraven. Dvs åstadkomma minimikraven först, även om extra funktionalitet finns beskriven i tickets. Jag har alltid mer ideer än vad som står i kraven, den här gången hade jag betydligt mindre tid på mid. Att utgå ifån existerande app gör det också lockande att försöka implementera extra funktionalitet, men jag fick hålla mig.

Som resultat har jag löst 4 krav: App, Kartan, Inloggningen och Native design och bara det jag anser vara grundkraven.

Graphical user interface, application

Description automatically generated

**Förbättringspotential**

Det finns många punkter där jag anser det skulle kunnat bli bättre. Första punkten är appens raison d’être: just nu visar appen bara förseningar. Det känns mer logiskt och nyttigt att visa all tåginformation, inte bara förseningar, utan också plattform och tågnummer.

Design:

* Titeln på vyn ska ligga i headern (dynamiskt kodad) med favoritikon
* Gult backgrund bakom nya tiden i listan av förseningar (DelaysList) är kortare (jämnt avstånd runt texten)
* Log in är en drawer, inte en vy.

Navigation:

* Detaljvyn för försenade tåg från My page ska ha en bakåt pil som länkar till ’My page’, inte ’Trafic info’ som den gör nu

Kartan:

* Kartan skulle kunna centreras efter att alla markörer är på plats. Alt knappen för att centrera skulle vara inaktiv fram till dess.
* Markörerna skulle ha appens egna ikoner/bilder (de tillgängliga ser oprofessionellt ut)
* Markörerna (i titeln) skulle hantera multipla avgångar bättre, tex som en lista över en markör i stället för överlappande markörer
* Visa destination med markör av en annan färg

**Förseningar**

För första kravet har jag byggt två vyer: hem-vyn hämtar stationsdata och skriver ut stationslänkar. Listan är filtrerad på förseningar till den stationen. Iom att det är just en försenings-app väljer jag att inte visa stationer som har inga förseningar. OBS! Detta blir viktigt när man testar appens favourites-funktionaliteten (favoriter som har inga förseningar visas inte i förseningsvyn).

Trycker man på stationsnamnet får man en lista på alla tåg (från-stationerna) och planerade samt nya ankomsttider. SJ’s app, som jag använde som förebild hade också information om plattform och tågnummer. Jag lagt till fälten för dessa på sidan för krav 3, men APIet ger ingen information om dessa, så de förblir tomma.

**Krav2:**

**Karta och GPS**

Jag tolkar kravet som att kartan ska visa alla från-stationerna. På listan av förseningar lägger jag till en knapp för att visa förseningarna på en karta. Eftersom kartan renderas separat för alla till-stationer var det mest meningsfullt att lägga till den där (Istället för i Tab.Navigation tex).

Kartan är centrerad ungefär på hela Sverige, men det förekommer att markörerna faller utanför. Iom att användarens position tar längst tid att hämta, och jag har inte luskat ut hur jag kan rita om kartan automatiskt lägger jag till en knapp som centrerar kartan kring alla markörer. Det händer att det finns fler förseningar från samma station. Just nu ligger de på samma plats och överskuggar varandra, om man klickar på samma markör flera ggr visas dem i ordning.

**Krav3:**

**Native design**

Jag väljer SJ’s app som förebild. Det är en etablerad och välbesökt app med antagligen bra budget för design.

Färgschemat är enkel: vit text på svart bakgrund, med gult och gröt som accentfärger.

Jag utför en analys av nyckelelement i planeringsfasen. Viktigaste var att bestämma hur navigationen skulle fungera. Jag bifogar skärmdumpar av appen i mina tickets som referens.

För att återskapa designen använder jag mig av style-konstanter, som i föregående projekt. Det finns återkommande block för text och vyer som skapar konsekvent utseende med jämna marginaler och typsnittval. Jag gör konstanter små och lagrar dem på elementen som behövs.

Jag använder mig av AdobeXD för att få bättre insikt om tex bredden på navbaren och färgerna. Medans jag utvecklar har jag skärmdumpar tillhands och jämför ofta.

En viktig del av designen var typsnittet och ikonerna. Jag analyserar flera typsnitt och väljer Lato, en sans serif, där tex bokstäver a och g är mest lika dem i appen. För att använda egna typsnitt laddar jag ner ett gäng i App.tsx med useFonts(). För att säkerställa att alla fonts är laddade använder jag SplashScreen.

För att få operativsystemet att känna igen mörkt bakgrund fick jag justera inställningen i app.json till ”userInterfaceStyle”: automatic.

Och till slut organiserar jag stationsdatat i <DataTable>.

A screenshot of a computer screen

Description automatically generated with medium confidence A screenshot of a computer screen

Description automatically generated with medium confidence

Graphical user interface, application

Description automatically generated

**Krav4:**

**Autentisering**

Autentisering består av två vyer under tabben ’Log in’. Det finns möjlighet att logga in såsom registrera sig. Jag behåller mejl och lösenordsverifiering från Lager appen och lägger till info-meddelanden, ifall registrering inte går igenom tex pga att mejl är redan registrerad.

Inloggad användare kommer ha tillgång till Tab ’My Page’. Där visas det en lista av favoriter med information om hur många tåg är försenade till varje favoritstation.

Första utmaningen var att bestämma var favoriterna ska sparas. Jag bestämmer mig för att använda auth.emilfolino.se för det och lägga till det som artefact för varje användare. Sålänge man kommer ihåg att spara id för varje artefakt kan man sen lätt hantera borttagningen.

Vid varje stationsnamn på My page och Trafic Info finns det en stjärna som är ifylld med gult om stationen är sparad. Stjärnan är en knapp som låter stationen sparas eller bli borttagen från listan av favoriter (obs! ibland går det lite långsamt, så tryck inte för många ggr). Ett meddelande visas om stationen blivit tillagd/borttagen. Listan av favoriter och stjärnorna uppdateras när man klickar på stjärnan.

Den här delen var mest utmanande, nämligen att få hela listan och färgen på stjärnorna uppdateras. När jag lagt till setFavourites() i stjärnknappen har det fallit på plats. Förutom det var det problem med att uppdatera stjärnorna efter man tex loggar ut. Där hjälpte det att lägga isLoggedIn som andra argumentet i useEffect().

**Projektet**

Det här projektet var lite tufft för mig. Jag är mer van att kunna genomföra alla extra kraven, men jag den här gången har jag fått en deltids praktikplats som ledde till hårdare tidspress.   
Jag skulle nog säga att jag utnyttjade all tid som avsågs för projektet och ändå inte klarade av alla extra krav. Trots att jag anser projektets omfång och svårighetsgrad att i princip vara rimligt stor.

Jag kände mig mkt kompetent i början: att planera ut navigationen och att hantera data var både spännande och lätt. Men under projektets gång fastnade jag många gånger i saker relaterade till state och api anrop, som bromsade mitt arbete. Överlag känns det som både under kursens gång och på nätet finns det otillräckligt med material när det gäller state. Detta stämmer överens från mina tidigare erfarenheter med Android utveckling (att det blir många mystiska errors och sökande på nätet för att hitta lösningar).

**Kursen**

Kursen var, som vanligt på BTH, mycket bra. Jag rekommenderar dessa kurser varmt till alla jag pratar med. Expo är ett smidigt ramverk att jobba med, och förutom mkt mer utförligt introduktion till useState() och useEffect() hade jag nog inget att önska mig. Det hade också varit intressant att faktiskt slutföra appen till en riktig app som man kan ladda ner och köra. Känns som en viktig del av utvecklingsprocessen. Kursen får 8/10. För 10 skulle jag nog uppskattat mer kontakt med andra studenter. Det är mkt svårt att uppmana distansstudenter för grupparbete. Jag skulle kunna föreslå tex presentationer i mindre grupper i början av kursen (10 pers, man kan se ansikten lite, få höra om bakgrund och ambitionerna) och i slutet då man kan tex dela med sig av sina framgångar och bekymmer (gällande kursen naturligtvis).