# Planering av arbetsflöde

Eftersom jag har bristande erfarenhet av MVVM och WPF så tänkte jag börja mitt projekt med att skapa en fungerande navigation med MVVM. Då får jag en lite bättre idé om vilka klasser och egenskaper/metoder som jag behöver planera för.

Efter det är klart kan jag börja på att skissa på mina vyer och rita grova utkast av hur vart och ett av dom ska se ut. Då kan jag rita ut var alla knappar och alla element ska finnas. Det blir då lättare att veta vilka metoder, vilka egenskaper och vilka kommandon som behövs i dom olika vymodellerna.

I sista delen av planeringsstadiet tänkte jag skapa ett klassdiagram. Nu borde jag ha tillräcklig kunskap för att få en grov bild av hur mitt program ska vara strukturerat. Jag lägger in det jag fått fram av dom två ovanstående momenten, och jag börjar planera för logiken som ska finnas i programmet.

Det sista momentet är att börja implementera alla klasser, vymodeller och vyer som jag planerat för i klassdiagramet och lösa eventuella problem som kommer att uppstå.

# Vyer och deras funktionsbeskrivning

## MainMenuView

Min MainMenuView ska bestå av en ganska enkel vy med två tabbar. Den ena tabben kommer innehålla en lista med alla olika Quiz, och den andra tabben kommer innehålla en lista med alla olika kategorier. I tabben med Quiz kommer det i nederkant att finnas knappar för att skapa/editera, importera, exportera eller ta bort Quiz, samt att spela. I tabben med kategorier ska du kunna välja en eller flera kategorier i listan. Du ska kunna skriva in max antal frågor som du vill ha och du ska kunna trycka på en knapp för att genera Quiz:et.

## PlayQuizView

I PlayQuizView ska man först få upp Quizens titel i några sekunder. Sen visas första frågan. Om en bild finns med i frågan visas frågan först i några sekunder, och sen visas bilden och dom 4 svarsalternativen. Om användaren klickar på det rätta svarsalternativet spelas ett ljudklipp upp och knappen användaren tryckte på blir grön. Om användaren klickar på fel svarsalternativ spelas ett annat ljudklipp upp, och knappen användaren tryckte på blir röd. Knappen med det rätta svarsalternativet blir också grön, så att användaren ser vilket som var det rätta svaret. När Quizet är färdigt står det ”Bra jobbat” och den totala poängen i några sekunder. Sen återvänder användaren till huvudmenyn.

Ovanför rutan som innehåller bilder/frågan ska en liten label vara placerad som visar användarens totala poäng.

## CreateQuizView & EditQuizView

I CreateQuizView och EditQuizView ska man kunna skapa eller editera Quiz. Dom kommer se i stort sett identiska ut. Det kommer finnas fält för titel, kategori, svarsalternativ. Det kommer också att finnas en lista för alla frågor, ett image element som visar frågans bild om en sådan finns, samt knappar för att lägga till och ta bort frågor och bild. Titeln kan man skriva in direkt, men för att man ska kunna få tillgång till dom andra fälten måste man först skapa en fråga genom att trycka på knappen.

# Problemanalys

## ReadOnly Properties i DataModels

Det mest uppenbara problemet är att alla properties i datamodellerna vi har fått är readonly. Detta innebär att vi inte kan binda våra element i CreateQuizView/EditQuizView direkt till våra properties, så det blir problem när vi ska ändra en Quiz eller en Question.

Om man hade valt att CreateQuizView/EditQuizView läste in en quiz direkt från JSON, och sen skrev ner direkt till JSON när man redigerat klart hade det här inte varit ett problem. Problemet den lösningen hade skapat är att absolut alla JSON filer måste läsas in på nytt när man till exempel vill spela en Quiz baserat på kategorier. Så programmet hade blivit väldigt snabbt när vi vill spela en färdig Quiz men när vi vill spela efter kategori hade det tagit väldigt långt tid att ladda, speciellt om det finns många JSON filer.   
Om vi har en Datastore behöver JSON filerna bara läsas in en gång, men vi kommer att behöva ta bort Quizet vi redigerar och ersätta det med det redigerade Quizet. Lite mer jobb, men det blir bäst i slutändan hoppas jag.

## Namngivning av bilder och JSON

Ett annat problem som kan uppstå är med bilderna som ska kunna kopplas till våra frågor. Dom måste ju flyttas till någon form av mapp i AppData\Local där all quiz data ligger lagrade. Dom behöver också döpas om till ett autogenererat namn så att det inte blir problem om användaren väljer att lägga in två bilder med samma namn. Det ska inte heller bli några problem om användaren väljer att skapa ett quiz med samma namn som ett redan befintligt quiz.

## Ge feedback tillbaka till Views

Ett annat problem som kommer uppstå är när vi ska ge feedback tillbaka till våra Views. Det kan till exempel vara när en knapp ska ändra färg beroende på om användaren svarar rätt eller fel. Detta är egentligen inte ett problem, men det kan vara svårt att förutse hur många properties som vi behöver planera för i förväg.

## Exportera och importera Quiz

En funktion jag tänkte implementera om tiden finns där är att importera och exportera quiz. Tanken är att programmet skapar en .zip fil med både JSON filen och alla bilder som är kopplade till Quizet. Filen döps sedan om till valfritt namn med filändelsen .quiz. Enligt Microsofts dokumentation verkar det vara ganska lätta att ska .zip filer, men eftersom jag inte gjort det förut är detta något jag behöver läsa på om först. Kanske ett problem, vem vet?

## Globala Styles

Ett problem som jag måste lösa är hur man implementerar globala styles i sina projekt. Eftersom vi har samma färgtema på alla views, så hade det varit bra om jag kunde implementera dom olika style:sen globalt. Detta är förmodligen inga problem, men jag vet hur mycket tid sånt här kan ta om det inte fungerar, och eftersom jag inte har någon erfarenhet av den sen innan så får det stå på min lista.