

# Prototype Design Document

Group: TruckDriversUnitedz

Product: Svapp

Members: Lina Blomkvist, Josefin Ulfenborg, Alexander Nordgren, Tove Ekman, Emilia Vestlund, Razmus Strandell

---

## 1. Design Rationale

### 1.1 – Choice of API level

När vi startade vårt projekt valde vi att göra så vår applikation går att köra på enheter med API level 18 och högre. Detta då det för oss var den bästa balansen av tillgängliga funktioner samt hur många enheter som stöds, vilket med API level 18 beräknas vara närmre 85% av alla androidenheter. Att sänka till exempelvis API level 17 eller lägre hade förvisso ökat antal kompatibla androidenheter ytterligare, men det på bekostnad av en del funktioner i android vilket gjorde att vi valde att stanna på just API level 18. Inget på det planet har ändrats och därför ligger vi givetvis kvar på samma API.

### 1.2 External Dependencies

#### 1.2.1 Support frameworks

Vi har använt oss av två av Androids egna support frameworks, design och appcompat v7. Med appcompat v7 medföljer bland annat önskvärda features som actionbaren som vi använder för sökning och AppCompatActivity som används av activityklasser. Design använder vi för alla våra snygga designkomponenter mm.

#### 1.2.2 Testing frameworks

Under både första delen av kursen och nu i vintras hade vi sagt att vi skulle göra tester till vårt projekt. Detta blev inte av men JUnit lades till i projektet tidigt för att göra det möjligt att enkelt skriva tester.

#### 1.2.4 Google Play

Vid förfrågan av en grupp intresserad av vår app har vi bestämt oss för att lägga upp vår app på Google play. Då vi just nu inte har haft tillräckligt med tid för detta har vi bara lagt till första biblioteket `com.google.android.gms:play-services-appindexing:8.1.0`. Detta bibliotek skall användas för att få vår app att komma upp i Google när man söker på dess namn.

## 2. Overview

### 2.1 Behavioral and Structural

Vi har försökt följa MVC-mönstret så gott som det går, dock skiljer sig Androids standardmönster av appar från detta genom att kombinera View och Controller i samma fil, kallad Activity.

Model-, Activity-, Enum-, och databas-klasserna har vi fördelat i olika paket med självbeskrivande namn. Då Activity fungerar som både View och Controller så har vi valt att kalla paketet innehållande dessa för "controllers". Databas och andra manager-klasser har vi lagt i paketet "services".

En tydligare uppdelning i paket hade varit lättare att åstadkomma om vi helt och hållet hade kunnat gå via MVC-mönstret.

### 2.2 Protocol (client/server)

Eftersom appen inte använder sig av internetuppkoppling har vi inte använt oss av någon klient/server byggnad. Databasen är inte heller beroende av externa bibliotek eftersom den är tillräckligt liten för att kunna laddas i samband med uppstart av appen.

## 3. User Stories (nya)

Våra user stories ligger nu i en ordning som är sorterad efter prioritet. I den första tabellen finns de user stories som vi genomförde, och i den andra tabellen det vi pratade om men valde att inte göra, då vi inte ansåg att just de var lika viktiga. De user stories vi implementerat har vi implementerat för att fixa en bristande app, som i många avseenden hade stora förbättringspotentialer.

Vi skulle komma på user stories till "storlek fem" och implementera dem. På grund av att vissa stories var så små så slutade det med att vi fick sju stories, varav två av dem delades in till tre vardera. Därav ser det ut som att vi implementerat betydligt mer än anvisningarna, men det är helt enkelt på grund av att vi ansåg att vissa var så pass små och vi behövde implementera fler.

Velocity per person: 20 (120)

EP: Står för Energy Points, refereras till som "siffra" i reflektionsrapporten.

Prioritet: Indelade från mest viktigt (1), mindre viktigt (2) och om hinns med (3).

## IMPLEMENTERADE

Prioritet	Namn	Story	EP
1	Sökfunktion(1)	För att åtgärda brister i applikationen vill vi göra en förbättrad sökfunktion så att man förstår vad man kan skriva in så att sökfunktionen blir enklare att använda.	4
1	Sökfunktion(2)	För att åtgärda brister i applikationen vill vi göra en utökad sökfunktion så att man kan söka på mer än bara bokstäver.	14
1	Inspelning	För att åtgärda brister i applikationen vill vi införa en klarare inspelningsknapp så att det blir lättare att förstå när man spelar in och när man inte gör det.	14
1	Grid	För att åtgärda brister i applikationen vill vi införa en knapp-vy istället för en list-vy över bokstäverna så att GUI:t blir mer konstant och det blir enklare att använda menyn för folk med större fingrar.	8
1	Tutorial	För att åtgärda brister i applikationen vill vi införa fler hints hur GUI:t fungerar så att det är enklare att förstå hur vår app skall användas.	16
2	Sökfunktion(3)	För att åtgärda brister i applikationen vill vi införa en resultatmeny när man söker så att man kan välja vad man letar efter vid utökad sökfunktion.	14
2	Volym	För att åtgärda brister i applikationen vill vi höja volymen på de ljudfiler vi har så att det är enklare att lyssna på orden.	4
2	Otydliga bilder/ord	För att åtgärda brister i applikationen vill vi byta ut otydliga ord så att det är enklare att förstå vad ikoner betyder.	4
2	Achievement system(1)	För att åtgärda brister i applikationen vill vi implementera ett belöningssystem så att vår applikation blir mer interaktiv.	6
3	Achievement system(2)	För att åtgärda brister i applikationen vill vi implementera ett belöningssystem med notifikationer när man får en belöning så att vår applikation blir mer interaktiv.	14

3	Achievement system(3)	För att åtgärda brister i applikationen vill vi implementera ett belöningssystem som sparar achievements när man startar om appen så att vår applikation blir mer interaktiv.	12
---	-----------------------	---	----

#### **EJ IMPLEMENTERADE**

Prioritet	Namn	Story	EP
3	Spel	För att åtgärda brister i applikationen vill vi implementera ett spel så att vår applikation blir mer interaktiv.	x