

## Kravspecifikation

Förberett för kursen TDDI02

Bearbetat av: Tobias Lundgren, Elias Resman,  
Jonatan Sävenhed, Martin Weman

12 september 2014

Projektnamn: SettleApp

Handledare: Rebecka Geijer Michaeli

Version 3

---

# Innehåll

1. Introduktion	3
1.1. Syfte	3
1.2. Projektets Syfte	3
1.3. Målgrupp	3
1.4. Beskrivning	3
1.4.1. Inloggning	3
1.4.2. Användargränssnitt	4
1.4.2.1. Användning/Funktioner	4
2. Avgränsningar	4
2.1. Lösning	4
2.2. Implementation	4
3. Funktionella Krav	5
3.1. Skallkrav	5
3.2. Börkrav	5
4. Icke-funktionella Krav	6
4.1. Användbarhet	6
4.2. Felhantering	6
4.3. Prestanda	6
5. Tidsuppskattning	7

# 1. INTRODUCTION

Här beskrivs projektet samt dess syfte och målgrupp, även en beskrivning över användning och funktioner.

## 1.1. SYFTE

Syftet med denna kravspecifikation är att beskriva vad projektet skall innehålla, hur användargränssnittet skall utformas samt vilka funktioner som skall finnas. Kravspecifikationen skall vara så pass fullständig att man utifrån den har all nödvändig information för att kunna börja utveckla projektet till att slutföra det.

När beslut om att gå vidare till utveckling är gjort så skall kravspecifikationen användas som beslutsunderlag.

## 1.2. PROJEKTETS SYFTE

En mobilapplikation som håller koll på när du och dina vänner lånar pengar av varandra för att underlätta vardagen samt motverka oense och konflikter.

Enligt livserfarenheter är detta en vanligt problem idag att folk glömmer betala det som man lånar av varandra, allra helst om det handlar om mindre summor. Mindre summor blir stora i längden vilket påverkar individens ekonomi.

## 1.3. MÅLGRUPP

Vår målgrupp är alla personer med en iOS-enhet men främst för den yngre generationen som ofta lånar pengar av varandra och ibland glömmer bort sin skuld. Ingen förkunskap skall krävas för att kunna använda applikationen så även de användare som inte är vana med smartphone skall kunna känna sig bekanta med appen.

## 1.4. BESKRIVNING

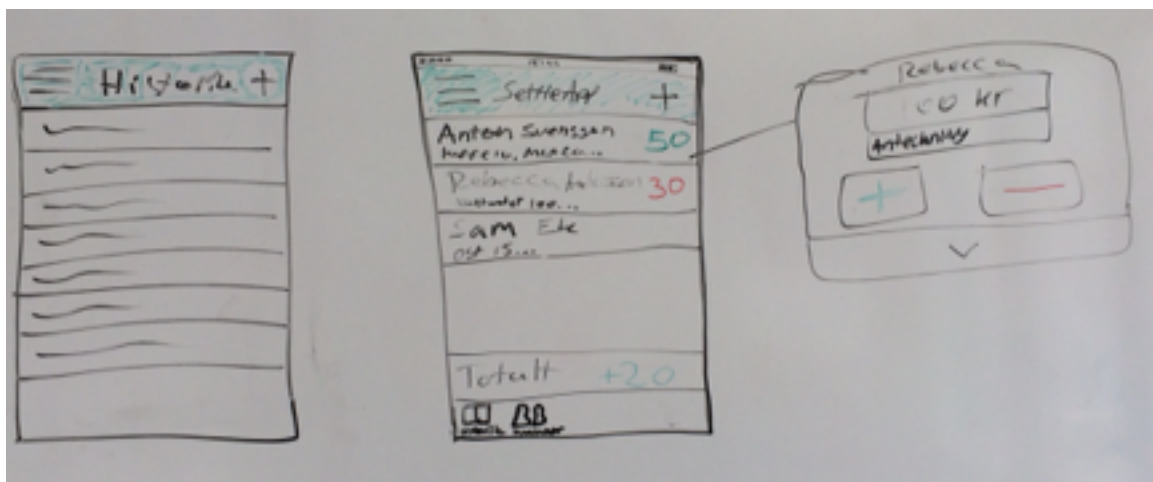
En snabb genomgång om hur applikationen skall användas och vilka funktioner som finns.

### 1.4.1. INLOGGNING

Applikationen startar första gången med en inloggningsruta där man kan välja att skapa en användare eller logga in med ett befintligt konto. Kontoupppgifterna som man anger är användarnamn, förnamn, efternamn, epostadress samt lösenord.

### 1.4.2. ANVÄNDARGRÄNSSNITT

När man befinner sig vid sig vid huvudskärmen så har man tillgång till alla funktioner samt en lista över dina kontakter med respektive skuld. Samt ett resultat över din totala balans mellan skulder (se figur 1).



Figur 1. En skiss som visar upp hur appen gränssnitt skall efterlikna.

#### 1.4.2.1. ANVÄNDNING/FUNKTIONER

För att lägga till en kontakt att debitera eller kreditera en skuld så skall man använda lägg till knappen (+) som finns uppe till höger i applikationen i figur 1.

För att komma till menyn som visar en översikt över alla funktioner så använder man meny-ikonen(☰) uppe till vänster i figur 1. I menyn finns det länkar för att visa användaruppgifter, lägg till användare, beskrivning om appen och visa kontakter.

## 2. AVGRÄNSNINGAR

### 2.1. LÖSNING

Lösningen är en applikation där användaren kan lägga in skulder till andra samt andras skulder till användaren för att hålla koll på den totala skulden till varje person.

### 2.2. IMPLEMENTATION

Systemet skall kunna användas på alla iOS-enheter med iOS-version 7.0 eller senare. Varje användare kan lägga till max 100 kontakter var och obegränsat antal skulder så länge det finns plats i databasen. Databasen kommer utökas om den blir full.

Applikationen kräver internetanslutning för att synka med användarens databas.

## 3. FUNKTIONALLA KRAV

Här nedan tas projektet skallkrav upp där det beskrivs vad som skall uppfyllas inom projektets ramar, samt projektets börkrav som skall uppfyllas vid mån av tid.

### 3.1. SKALLKRAV

Applikationens funktionalitet innefattar dessa krav:

- 3.1.1. Man skall kunna skapa en användare.
- 3.1.2. Användarens information skall lagras i en databas.
- 3.1.3. Informationen som skall lagras i databasen består av användarnamn, förnamn, efternamn, epostadress, lösenord och all info över skulderna.
- 3.1.4. Man skall kunna debitera och kreditera skulder mellan olika personer, den andra personen skall inte behöva ha applikationen.
- 3.1.5. Skulderna skall räknas ut för varje kontakt och läggas samman med den befintliga skulden om sådan finns.
- 3.1.6. En översikt för total balans av skulder skall finnas på huvudsidan.

### 3.2. BÖRKRAV

När övriga krav är uppfylla så kommer vi att jobba vidare med dessa funktioner:

- 3.2.1. En kommentar skall kunna läggas till tillsammans med skulden.
- 3.2.2. En sökmotor som kan användas för att söka efter andra användare.
- 3.2.3. Användaren skall kunna lägga till och spara andra användare som har ett konto i sin kontaktlista.
- 3.2.4. Skulder mellan användare skall synkroniseras så att de visar samma skuld hos båda parter.
- 3.2.5. En notis skall skickas till den andra användaren när man lägger till en ny skuld eller om en befintlig skuld förändras.
- 3.2.6. Applikationen skall fungera utan internet med begränsad användning, aktuella skulder skall kunna visas samt skall man kunna lägga till en skuld lokalt på en person.
- 3.2.7. All historik över händelser på skulderna skall sparas på användarens databas.
- 3.2.8. Räkna ut var och ens skuld om flera parter i ett nät av kontakter är skyldiga varandra pengar.
- 3.2.9. Automatiska påminnelser skall skickas ut om obetalda skulder.

## 4. ICKE-FUNKTIONELLA KRAV

Krav som inte har någon slags funktion för användaren beskrivs här.

### 4.1. ANVÄNDBARHET

Menyerna skall vara lätta att förstå sig på och endast enkla och nödvändiga funktioner skall finnas tillgängliga.

Det skall gå snabbt och vara enkelt att lägga till och ändra skulder.

Målet är att aldrig vara mer än tre steg ifrån att göra en önskvärd aktion när man står i startmenyn.

Språket i appen skall vara på svenska men kan möjligtvis också vara tillgängligt på engelska om tid finns.

### 4.2. FELHANTERING

- Om en ny användare skall skapas men användarnamnet redan finns i databasen ska användaren inte skapas. Ett meddelande skall då visas och användaren får mata in användarnamn på nytt.
- Skulle man försöka skapa en skuld till en användare som inte finns i databasen skall ett meddelande skrivas ut.
- När det inte finns tillgång till internet när så behövs så skall ett meddelande skrivas ut på skärmen.
- Stämmer inte inloggningsuppgifterna så skall också ett meddelande skrivas ut.

### 4.3. PRESTANDA

Inga krav på exakta laddningstider och dylikt finns, dock skall appen fungera flytande utan några störande fördröjningar.

## 5. TIDSUPPSKATTNING

Projektets sammanlagda tid skall ta cirka 440 timmar baserat på hur många poäng kursen är på.

Vi har valt att fördela dessa timmar på följande sätt:

- Planering och förberedelse: 30h
- Design och layout: 40h
- Skapa en databas: 20h
- Koppla ihop databas med applikation: 85h
- Programmera applikationens funktioner: 50h
- Programmera uträkningsmomentet: 50h
- Utsatt tid för felsökning: 100h
- Sammanställning: 25h
- Möte med handledare: 10h
- Diskussioner och ändringar under projektets gång: 30h