



MyStepCounter

Examensarbete rapport

Morgan Jalil | MA16_EXARB | 2017-09-14

Inledning:

Jag ville göra något helt nytt, samtidigt som jag skulle få användning för det i arbetslivet. Först bollade jag idén om att göra ett spel med en spelmotor, men kom fram till att även om det skulle vara roligt så hade nog inte haft någon större nytta med att lära mig Unity.

Tankarna gick sedan att använda hårdvaran i telefonen, och kollade lite på Bluetooth, men såg att den inom en snar framtid skulle uppdateras. Jag har intresse för framtiden med NFC men det jag kom fram till var att det var väldigt begränsat vad man fick göra, om man då inte "rootade" Android. Fanns inte på kartan.

Slutligen fastnade jag för att använda sensorn för att räkna steg, och varför inte göra det mer avancerat med att försöka logga flera dagar bakåt samt ha olika grafer som visar framstegen?

Jag bestämde mig för att göra en stegräknar-applikation som skulle vara så enkel som möjligt för folk som vill ha koll på att dom rör på sig tillräckligt men inte bli stressad. Min applikation skall vara en positiv sak och inte bli ett stressmoment för användaren, samt vara väldigt enkel att använda.

Metod:

Jag började med att kolla hur andra hade gjort en egen stegräknare, och hittade en algoritm för att räkna ut hur man gör för att telefonen skall reagera och få ut stegen. Jag fick en fungerande beta applikation efter att ha slagit samman kod från olika personer med egna ändringar i koden, samt att jag finjusterade algoritmen då den ursprungligen reagerade väldigt trögt. Kom fram till en bra känslighet samt att den autoupdaterade stegen. Men jag kände att jag inte riktigt lärde mig det jag ville samt att den som hade algoritmen jag tog grunden ifrån inte hade någon susning om hur den fungerade.

Jag leta vidare efter mer info om stegräknare och hittade Google fit API, jag började om från början.

Jag följde <https://developers.google.com/fit/overview> väldigt mycket och använde mig av SENSOR API för att få fram steg i realtid, och kom fram till att det fungerade bra men var inte användbar i långa loppet. Fortsatte med att använda mig utav RECORDING API och HISTORY API som egentligen är dom viktigaste delarna för att få grunden att fungera samt hämta steg, även om det inte skulle bli i realtid.

Hittade även bra information om hur man kunde använda HISTORY API för att få fram exakt datum inom ett intervall, och jag fastnade för att få fram data sju dagar bakåt i tiden

Och dagens totala steg. Jag kände att det gick så bra att jag valde att använda mig utav fragment med en Page Adapter, och sedan lägga in grafer i slutet som tar data från dom olika resultaten och fyller graferna.

Jag lekte även med kalorier och fick grundkoden att fungera men slopade det då jag kände att det kunde bli ett stressmoment för användaren. Och att jag började göra det för avancerat för mig då fragment med Page Adapter kändes krångligt.

Jag lyckades till slut få min kod för daglig stegräkning att skicka ett värde till mitt första fragment genom `onPostExecute`, göra om det till bundle och skicka till fragmentet. Samt använda mig utav `runOnUiThread` för att uppdatera `TextView`.

Resultat:

Jag har lyckats få fram en fungerande Applikation som automatiskt kräver inloggning med konto och förfrågningar om permissioner för diverse olika API:er och använder mig utav OAuth 2.0.

`MyStepCounter` uppdaterar stegen själv, inte i realtid men så fort du går in i applikationen, även om telefonen är enbart i viloläge eller att applikationen är avstängd. Den räknar steg i bakgrunden oavsett vad och börjar om från noll vid midnatt. Den drar otroligt lite batteri och är enkel att använda. Man kan pausa Applikationen från att räkna steg, och det sätts på automatiskt igen när man går in i applikationen för att göra det så enkelt som möjligt.

Den håller även reda på data för sju dagar bakåt men just nu enbart i loggen.

Problem:

Mitt störta problem var just fragment med Page Adapter, eftersom det var helt nytt för mig så fick jag leta land o rike för att lösa hur man kunde skicka data till fragmenten.

Men det andra problemet var att när jag skulle skicka nästa bundle till en graf så fick jag problemet att den skickade två bundles, den första som var null som crashade applikationen och den andra som faktiskt skickade kod. Jag fick ingen lösning på det förutom att jag eventuellt hade byggt upp mina fragment fel. Och då skulle det inte gå att skicka till några grafer överhuvudtaget. Jag lade ner den biten och putsade på UI/UX istället.

Det mest irriterande var att jag trodde att min kod för sju dagar bakåt inte fungerade tills idag. Men av en slump så fick jag in värden så koden för den biten fungerar men bara i loggen. Simon från min grupp har bugg testat applikationen, samt Mats, Mikael och Mikael och ingen har fått någon bugg.

Det enda som förbryllade mig var att den bara frågade efter permissioner en gång, oavsett om man rensade appdata och raderade applikationen samt rensade i Google play service. Kom fram till att det inte är min applikation som kräver permissionerna utan självaste Google API i telefonen (ibland är man bra dum).

Sammanfattning:

Detta examensarbete har varit väldigt kul att göra och även om jag inte fick min ursprungliga plan att fungera så blev resultatet ändå en bra applikation som är stilren snygg och räknar stegen korrekt!

Jag förstår min kod och varför jag var tvungen att använda vissa metoder och dylikt, lärt mig från misstagen om varför saker i slutändan inte fungerade, och jag har lärt mig otroligt mycket om hur jag borde planera bättre och inte lära mig många nya saker på en gång. Jag tror att om man kapade tex fragment helt så hade jag nog haft en applikation mer flera fungerande funktioner.

Jag har en tendens att ha för storslagna visioner och göra saker svårare för mig för att utmana mig, och mina enda två hinder för att lösa problemen har varit tiden, samt att jag sitter och samtidigt lär mig React Native för min praktik på Clas Ohlson.

Men det bästa är ändå att även om jag inte har lyckats lösa mina flertal felkoder, så har jag fått stor erfarenhet och lärt mig massor utav misstagen.

Morgan Jalil