



JÖNKÖPING UNIVERSITY

School of Engineering

Performance comparison between React Native and Kivy

Undertitel på en eller
flera rader

HUVUDOMRÅDE: *Datavetenskap*

FÖRFATTARE: *Joacim Lilja*

HANDLEDARE: *Handledare*

JÖNKÖPING 2020 Månad

Detta examensarbete är utfört vid Tekniska Högskolan i Jönköping inom [se huvudområde på föregående sida]. Författarna svarar själva för framförda åsikter, slutsatser och resultat.

Examinator: XXX XXXXXXXX

Handledare: XXX XXXXXXXX

Omfattning: 15 hp (grundnivå)

Datum: 2020-XX-XX

Postadress:

Box 1026

551 11 Jönköping

Besöksadress:

Gjuterigatan 5

Telefon:

036-10 10 00 (vx)

Abstract

[illegible][illegible][illegible][illegible][illegible][illegible]

Sammanfattning

Syfte – XXX

Metod – XXX

Resultat – XXX

Implikationer – XXX

Begränsningar – XXX

Nyckelord – XXX

Sammanfattningen är en kort beskrivning (minirapport) av rapportens innehåll och bör omfatta högst en A4-sida. Syftet med sammanfattningen är att läsaren snabbt ska få en uppfattning om arbetets förutsättningar och resultat. Det är därför viktigt att sammanfattningen innehåller så konkreta uppgifter som möjligt. Det är viktigt att du lägger mycket tid på att formulera dig eftersom sammanfattningen ofta är det som avgör om rapporten bedöms som värd att läsa eller inte.

Innehållsförteckning

Abstract	1
Sammanfattning	2
Innehållsförteckning	3
Introduktion	5
Bakgrund	5
Problembeskrivning	6
Syfte och frågeställningar	6
Omfång och avgränsningar	7
Disposition	7
Metod och genomförande	8
Koppling mellan frågeställningar och metod	8
Arbetsprocessen	8
Ansats	8
Design	9
Datainsamling	9
Dataanalys	9
Trovärdighet	9
Teoretiskt ramverk	10
Koppling mellan frågeställningar och teori	10
Mobilapplikationer	10
React Native	10
Kivy	10
Cross-Platform verktyg	10
Skillnader mellan Kivy och React Native	10
Empiri	12
...beskrivning...	12
XXX	12
XXX	12
Analys	13
Frågeställning 1	13
Frågeställning 2	13
Frågeställning 3	13

Diskussion och slutsatser	14
Resultat	14
Implikationer	14
Begränsningar	14
Slutsatser och rekommendationer	14
Vidare forskning	14
Referenser	15
Bilagor	16

1 Introduktion

Mobilappar har blivit en stor del av vardagen för de allra flesta, allt mellan viktiga bankärenden och tidsfördriv när man väntar på bussen. Mobiltelefonen är ett av våra viktigaste redskap i det moderna samhället, och det ställer vissa krav på prestandan i apparna. En app som är dåligt optimerad irriterar användaren och appen slutar användas, samtidigt som en till synes snabb app istället kan använda alltför mycket batterikapacitet. Det gör att prestandan i appar blir ett viktigt område, balansen mellan prestanda och energieffektivitet är hårfin.

1.1 Bakgrund

Då React Native och Kivy, två open source cross-platform verktyg, är skapade för att fylla samma funktion, att enkelt kunna bygga mobilapplikationer med kunskap i språk som är skapade för andra huvuduppgifter, var tanken att jämföra dessa för att se vilka styrkor och svagheter de olika ramverken har, och även göra olika tester av prestanda med hjälp av de vanligaste algoritmerna där man kan mäta exekveringstid och jämföra de två ramverken. Anledningen att detta är ett viktigt område ligger i hur utvecklingen av applikationer ändras, och även om Native utveckling fortfarande stödjer fler funktioner, är det viktigt att undersöka vilka möjligheter och skillnader som finns hos alternativa lösningar. Då de ramverken som ska undersökas kan bygga appar till både Android och iOS krävs bara en kodbas för utveckling mot båda plattformar, vilket är mer kostnads- och tidseffektivt gentemot att ha olika kodbasen och utvecklare för de två plattformarna.

1.2 Problembeskrivning

Problembeskrivningen syftar till att ge en mer utförlig beskrivning av de delar som ingår i den problembild som målats upp i bakgrunden. Ett viktigt syfte i problembeskrivningen är att lyfta det specifika problemet t ex det specifika uppdraget från ett företag till ett mer generellt och allmängiltigt problem som kan vara relevant för fler intressenter än det specifika företaget, t ex andra branscher eller affärsområden. Detta görs genom ett resonemang och argumentationskedja där det specifika i uppdraget framställs som ett specialfall av något mer generellt. Förutom att visa på ett bredare behov av den specifika uppgiften är det också viktigt att vad som är "state of the art" forskningsmässigt för detta område och de olika relevanta delarna inom detta område. Detta görs lämpligen genom att beskriva forskningsfronten för området med dess ingående delar. Genom detta resonemang X görs successiva ställningstaganden vilket skall leda till en välargumenterad och spårbar specificering och "nertrattning" av problemområdet ur vilket sedan syfte och frågeställningar kan falla ut på ett naturligt sätt.

På grund av den snabba utvecklingen av ramverk och verktyg för apputveckling är det viktigt att kunna jämföra olika verktyg för att bättre kunna välja det som passar bäst för den uppgift som ska lösas. Det finns idag ingen vetenskaplig jämförelse på skillnaden i prestanda mellan Kivy, ett ramverk för apputveckling i Python, och React Native, ett ramverk för apputveckling i Javascript. Då dessa båda ramverk fyller samma funktion från olika programmeringsspråk som är två av de mest använda programmeringsspråken (PYPL, u.å.) är en jämförelse viktig för att se vilket av dessa ramverk som är effektivast(Asp, 2018).

1.3 Syfte och frågeställningar

I problembeskrivning framgår att en vetenskaplig jämförelse mellan dessa ramverk saknas. Vidare framgår att båda språken är väldigt populära och jämförbara i sin anpassningsbarhet. En studie som jämför dessa verktyg kan då bidra med ny kunskap inom området och ge stöd för framtida val av ramverk för utveckling av mobilapplikationer.

Därmed är syftet med denna studie:

Att jämföra prestandan i ramverken Kivy och React Native, två ramverk skapade för utveckling av mobilapplikationer.

För att kunna besvara syftet har det brutits ned i 2 frågeställningar.

Då prestanda är ett viktigt område i mobilapplikationer som påverkar exempelvis användarupplevelse och batterianvändning(Asp, 2018) kan skillnaden hos olika tekniker vara kritisk för att skapa en så bra produkt som möjligt. Appens prestanda skulle kunna vara den avgörande faktorn om appen används eller ersätts av en annan app med samma funktionalitet. Därmed är studiens första frågeställning:

- [1] Varför uppstår en prestandaskillnad i ramverken React Native och Kivy för utveckling av mobilapplikationer, med fokus på sorteringsalgoritmer?

Eftersom prestandan och exekveringstiderna beror på enhetens prestanda bör mer än en enhet testas. För att lättare kunna generalisera experimentet lämpar sig en fysisk enhet och en simulerad enhet, då dessa kan återskapas och jämföras för att potentiellt se olika resultat om vilket ramverk som presterar bäst. En simulerad enhet minskar även felmarginalen, då det går att kontrollera simuleringen i högre grad än en fysisk enhet, även om en fysisk enhet representerar verkligheten bättre(Pai & Krishnakumar, 2019). Därmed är studiens andra frågeställning:

- [2] Vilka skillnader uppstår mellan ramverken på en simulerad enhet och en fysisk enhet?

1.4 Omfång och avgränsningar

Här ska du komplettera syfte och frågeställningar med att beskriva vad som inbegrips i studien men framför allt vad som inte omfattas. Att tänka på är att, om arbetet i problemdiskussionen är väl genomfört kommer det att minska antalet avgränsningar som behöver göras.

Denna studie kommer omfatta apputveckling, javascript, python, ramverk för apputveckling och simulering. För att minimera missuppfattningar med Javascript och React, omfattar denna studie inte webbutveckling.

1.5 Disposition

Beskriv hur resten av rapporten är disponerad, alltså kort hur rapporten är uppbyggd för att hjälpa läsaren att få en bild av rapportens upplägg.

Kommande delar i rapporten beskriver vilka metoder som används och hur frågeställningarna är kopplade till metoderna, samt arbetsprocessen. Sedan beskrivs experimenten som kommer utföras och även datainsamlingen och dataanalysen. Det teoretiska ramverket beskriver hur de olika teorier som studien använder kopplas till frågeställningarna.

2 Metod och genomförande

Kapitlet ger en översiktlig beskrivning av studiens arbetsprocess. Vidare beskrivs studiens ansats och design. Därtill beskrivs studiens datainsamling och dataanalys. Kapitlet avslutas med en diskussion kring studiens trovärdighet.

Här börjar du med att motivera och beskriva vilken typ av studie du gjort, t ex en fallstudie eller en experimentell studie samt vilka specifika metoder du valt och vad de innebär, t ex intervju, enkät, experiment, design eller observation. Hänvisa till referenser bakom de olika metoderna på samma sätt som vid hantering av teori. Därefter redogör du för hur du utfört arbetet, dvs. hur du gått tillväga för att kunna besvara dina frågeställningar och uppnå syftet. Hit hör t ex. hur du genomfört intervjuer, utrustning du har använt, samt beskrivningar av försök som du gjort. Redogör för hur du har samlat in, bearbetat och analyserat data. Var noggrann i din beskrivning eftersom den påverkar bedömningen av validitet (giltighet) och reliabilitet (trovärdighet) i examensarbetet!

2.1 Koppling mellan frågeställningar och metod

I följande kapitel beskrivs metoder för datainsamling och dataanalys som används för att besvara studiens frågeställningar.

För att besvara studiens frågeställning kommer experiment genomföras för att kunna jämföra prestandan i de olika ramverken genom att analysera genererad empirisk data. De båda ramverken kommer användas för att utveckla likadana appar, där testningen av algoritmer och jämförelsen av ramverken kan utföras. De algoritmer som kommer testas kommer inte jämföras med varandra, utan endast ramverken varierar mellan experimenten.

Experiment är viktigt för att ge en objektiv jämförelse mellan de två ramverken(Zobel, 2014), vilket gör experiment till en passande metod för att generera empirisk data som kan analyseras i detta arbete och generera svar på frågeställningarna.

2.2 Arbetsprocessen

Arbetet kommer inledas med en förstudie bestående av en litteraturstudie för att kunna få en bättre inblick in ramverken och deras utvecklingsprocess.

Implementationen av applikationerna kommer även inledas så tidigt som möjligt för att ha tid till flera iterationer av applikationen, där data kan samlas in och jämföras, för att se olika skillnader i prestanda.

Den empiriska data som genereras kommer jämföras mellan ramverken, och analyseras för att se om det existerar skillnader, och om skillnader finns vilket ramverk som presterar bäst.

Rapporten kommer bearbetas och skrivas på under så lång tid som möjligt för att undvika fel och minska stressnivån under arbetets tidsintervall.

Bilaga 1 beskriver en tidsplanering för arbetet, där det framgår vilka moment som är planerade och hur mycket tid som är beräknat att behövas till varje moment.

2.3 Ansats

Denna studie är en deduktiv studie(Sternberg et.al., 1954), och baserat på att det existerar skillnader i andra ramverk för apputveckling, bör även en viss skillnad i prestanda mellan dessa ramverk existera.

2.4 Design

Experimenten kommer jämföra tiden det tar för algoritmer att bearbeta samma data i de olika ramverken, för att kunna mäta skillnaden. Resultatet av detta kommer bli en tid som kan jämföras för att ta reda på vilket av ramverken som är effektivast. Detta tillvägagångssättet kommer användas för att jämföra ramverkens prestanda med de algoritmer som är vanligast i mobilapplikationer för att jämföra vilket ramverk som presterar bäst. Algoritmerna kommer jämföras var för sig, så det bara är ramverket som testas mot varandra, och inte algoritmerna.

2.5 Datainsamling

Studiens datainsamling bestod dels av litteraturstudier och dels utav insamling av empirisk data genom experiment.

Experimenten kommer bestå av att jämföra tiden det tar för en algoritm att utföra operationer på en förbestämd datamängd och sedan kommer tiden att jämföras mellan ramverken för samma algoritm.

2.6 Dataanalys

Datan som ska analyseras kommer främst bestå av den tid ramverken kan bearbeta data med en algoritm, där de kommer jämföras mot varandra. Processoranvändning är också intressant, för att se hur mycket av enhetens kapacitet som används. Varje algoritm kommer jämföras var för sig, för att sedan kunna dra en slutsats om vilket ramverk som presterar bäst för olika ändamål. En jämförelse mellan en simulerad enhet och fysisk enhet kommer göras, för att se hur enheten som testerna utförs på påverkar slutresultatet.

2.7 Trovärdighet

Datan som kommer användas för att jämföra ramverken kommer vara identisk för att ge algoritmerna samma förutsättningar, och göra ramverket i sig till den största variabeln. Experimenten kommer även köras både i en simulerad enhet på en dator och minst en fysisk enhet, för att se om någon skillnad uppstår.

3 Teoretiskt ramverk

3.1 Koppling mellan frågeställningar och teori

För att ge en teoretisk grund till den första frågeställningen "Finns det någon skillnad i prestanda i ramverken React Native och Kivy för utveckling av mobilapplikationer, med inriktning på vanliga algoritmer?" beskrivs följande områden i det teoretiska ramverket: React Native, Kivy, och Cross-Platform verktyg. React Native och Kivy behandlas därför att det är de ramverken som ska jämföras. Cross-platform verktyg behandlas därför att studien jämför två av dessa verktyg för att undersöka prestandaskillnader.

För att ge en teoretisk grund till den andra frågeställningen "Vilka skillnader uppstår mellan ramverken på en simulerad enhet och en fysisk enhet?" beskrivs följande områden i det teoretiska ramverket: Mobilapplikationer. Mobilapplikationer behandlas därför att det är en jämförelse av en fysisk och simulerad mobiltelefon, och jämför ramverken.

3.2 Mobilapplikationer

Då mobila applikationer används till allt från anteckningar till bankärenden och legitimering är prestanda och effektivitet viktigt för att användaren ska få en bra upplevelse.

En applikation skrivs traditionellt sett i det språk som plattformen kräver, vilket gör att man får en kodbas per plattform som man vill utveckla applikationen mot, vilket är tidskrävande. Där kommer cross-platform verktyg och låter dig utveckla appar mot flera plattformar från samma kodbas. Detta gör att utvecklingsprocessen blir betydligt snabbare och mer kostnadseffektiv(Pinto et al., 2018).

Ofta används både fysiska och simulerade enheter under utvecklingsprocessen. De simulerade enheterna är praktiska tidigt i processen, då man kan testa ändringar direkt i datorn, medans en fysisk enhet get mer verklighetstroga resultat och används oftast mer mot slutet av utvecklingsprocessen för att säkerställa att applikationen fungerar på ett korrekt sätt.(Pai & Krishnakumar, 2019).

3.3 React Native

Det populäraste ramverket för apputveckling(Jetbrains, u.å.), stödjer utveckling till både Android och iOS från samma kodbas. React Native är ett ramverk för Javascript utvecklat av Facebook som ger en enkel övergång från webbutveckling till mobilappar(Facebook, u.å.). React Native appar skrivs i Javascript, vilket är ett av de populäraste språken enligt Popularity of Programming Languages (PYPL, u.å.) och har många användningsområden, främst inom webbutveckling.

3.4 Kivy

Kivy beskrivs som ett ramverk för utveckling av mobilapplikationer, som stödjer utveckling till både Android och iOS från samma kodbas enligt Kivy(u.å.). Kivy är optimerat för att ge så bra prestanda som möjligt. Kivy använder sig utav OpenGL ES2 för att kunna nyttja enhetens grafikprocessor när det är effektivare(Kivy, u.å.), vilket kan ge fördelar i vissa fall gällande prestanda. Python är ett mångsidigt programmeringsspråk som kan användas inom många olika områden(Welcome to Python, u.å.).

3.5 Cross-Platform verktyg

Man kan utveckla appar på olika sätt, de två vanligaste är native, där man utvecklar en app för varje plattform med ett programmeringsspråk per kodbas, och Cross-Platform, som skapar en app utifrån samma kodbas till olika plattformar. Fördelen med cross-platform är att det går snabbare att utveckla applikationer då man bara behöver en kodbas, vilket snabbar på utvecklingsprocessen(xxx).

Skillnader mellan Kivy och React Native

De största skillnaderna mellan Kivy och React Native är att de är ramverk för olika programmeringsspråk, vilket leder till olika skillnader. Javascript används främst inom webbutveckling och körs i en Just in Time kompilator, kallad JIT(Zlatkov, 2018), vilket gör att Javascript koden exekveras snabbare(Wong, 2019) än Python koden som körs genom en interpreter(Karani, 2020).

4 Empiri

Kapitlet ger en översiktlig beskrivning av den empiriska domän som ligger till grund för denna studie. Vidare beskrivs empirin som samlats in för att ge svar på studiens frågeställningar.

I detta kapitel ska du beskriva dina empiriska data som du samlat in i ditt arbete. Beroende på vad du gjort kan detta kapitel se ut på olika sätt. Det är viktigt att du empiriska data på ett stringent sätt och att du presenterar fakta utan att lägga in personliga synpunkter eller värderingar. Det gör du i stället senare i analyskapitlet. Det är viktigt att hålla isär och tydligt redovisa vad som är insamlad empiri och vad som senare är din analys av empiriska data, d v s dina resultat.

4.1 ...beskrivning...

XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX

4.2 XXX

XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX

4.3 XXX

XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX

5 **Analys**

Kapitlet ger svar på studiens frågeställningar genom att behandla insamlad empiri och teoretiskt ramverk.

I detta kapitel skall studenten genomföra analysen baserat på behandlad teori och insamlad empiri, d v s att komma fram till resultaten av arbetet baserat på tidigare formulerade forskningsfrågor och syfte. Analysen skall genomföras enligt de principer som tidigare beskrivits i metodkapitlet.

5.1 **Frågeställning 1**

XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX
XXX

5.2 **Frågeställning 2**

XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX
XXX

5.3 **Frågeställning 3**

XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX
XXX

6 Diskussion och slutsatser

Kapitlet ger en sammanfattande beskrivning av studiens resultat. Vidare beskrivs studiens implikationer och begränsningar. Dessutom beskrivs studiens slutsatser och rekommendationer. Kapitlet avslutas med förslag på vidare forskning.

6.1 Resultat

XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX

6.2 Implikationer

XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX

6.3 Begränsningar

XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX

6.4 Slutsatser och rekommendationer

XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX

6.5 Vidare forskning

XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX XXX

Referenser

- The State of Developer Ecosystem in 2020 Infographic. JetBrains. <https://www.jetbrains.com/lp/devecosystem-2020/>.
- React Native · A framework for building native apps using React. <https://reactnative.dev/>.
- PYPL PopularitY of Programming Language index. <http://pypl.github.io/PYPL.html>.
- Facebook. facebook/react-native. <https://github.com/facebook/react-native>.
- Welcome to Python. Python.org. <https://www.python.org/>.
- Facebook. facebook/react-native. GitHub. <https://github.com/facebook/react-native>.
- Asp, F. (2018). A comparison of Ionic 2 versus React Native and Android in terms of performance, by comparing the performance of applications (Dissertation). Hämtad från <http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-147556>.
- Sternberg, Robert J., Mio, Jeffery Scott. (1954). Cognitive psychology. Cengage Learning/Wadsworth.
- Zobel, J. (2014). Writing for Computer Science (3rd ed.). Springer London. <https://doi.org/10.1007/978-1-4471-6639-9>.
- Zlatkov, A. (2018, December 12). *How JavaScript works: inside the V8 engine + 5 tips on how to write optimized code*. <https://blog.sessionstack.com/how-javascript-works-inside-the-v8-engine-5-tips-on-how-to-write-optimized-code-ac089e62b12e>.
- Karani, D. (2020, September 2). *How does Python work?* Medium. <https://towardsdatascience.com/how-does-python-work-6f21fd197888>.
- Wong, W. (2019, April 25). *NodeJS vs. Python 3 Performance*. DEV Community. <https://dev.to/wrrnwng/nodejs-vs-python-3-performance-10k6>.
- Pai, A., & Krishnakumar, P. (2019, June 20). *Emulators vs Simulators vs Real Device for Testing*. BrowserStack. <https://www.browserstack.com/guide/testing-on-emulators-simulators-real-devices-comparison>.
- Pinto, C. M., & Coutinho, C. (2018). From Native to Cross-platform Hybrid Development. *International Conference on Intelligent Systems (IS)*. pp. 669-676. doi: 10.1109/IS.2018.8710545

Bilagor

Bilaga 1 Tidsplanering

	V38	V39	V40	V41	V42	V43	V44	V45	V46	V47	V48	V49-5	V6	V7	V8	V9
Förstudie																
Implementation																
Datainsamling																
Dataanalys																
Rapportskrivande																
Etappredovisning																
Slutinlämning																
Framläggning																

Bilaga 2 etc.

Varje bilaga numreras enligt denna förteckning högst upp i högra hörnet på respektive bilaga.
Varje bilaga anges på ny sida.