

INGENIØRHØJSKOLEN AARHUS UNIVERSITET

BACHELORPROJEKT

RAMBØLL TILSYN



Analyse

PROJEKT NR. 17103

Navn	Au ID	Studienummer
Ao Li	AU512161	201407737
Morten Sand Knudsen	AU463338	201270955

VEJLEDER: LARS CHRISTIAN JENSEN

DATO: 19/12-2017

Indholdsfortegnelse

Kapitel 1	Foranalyse	4
1.1	Back-end	4
1.1.1	Database	4
1.1.2	Back-end udviklingsværktøjer	5
1.1.3	Valg af back-end database	5
1.2	Applikation	7
1.2.1	Styresystem	7
1.2.2	Cross-platform udviklingsværktøjer	7
1.2.3	Valg af cross-platform udviklingsværktøj	8
1.2.4	Xamarin	8
1.3	Eksportering	10
1.3.1	Data eksportering	10
1.3.2	Valg af data eksportering	10
	Ordforklaring	10
	Litteratur	12

Læsevejledning

Denne rapport redegør for den analyse, som er lavet til projektet Rambøll Tilsyn.

Analysen er inddelt i tre sektioner:

I første sektion gives der en analyse af valg af back-end til projektet. Derefter en mere dybdegående analyse af, hvad Firebase kan, og hvorfor denne er valgt.

I anden sektion analyseres udviklingsværktøjer til applikationen i projektet og en nærmere analyse af Xarmarin. Ligeledes en begrundelse for hvorfor Xarmarin er valgt til dette projekt.

I tredje sektion gives der en kort analyse af, hvilke eksporteringsprogrammer der er en del af projektet og hvorfor disse valgt.

I slutningen af rapporten findes ordforklaring og litteraturliste.

1

Foranalyse

I dette dokument vil der være en dybdegående beskrivelse af de overvejelser, der er blevet gjort i forhold til valg af software.

1.1 Back-end

I følgende sektion gøres der rede for beslutningerne i forhold til udvikling af back-end, heriblandt database, authentication og storage.

1.1.1 Database

I følgende afsnit redegøres for de beslutninger, der tages angående valg af database, SQL[1] eller NoSQL[2]. Der er fordele og ulemper ved begge database typer, herunder nævnes nogle af forskellene mellem de to typer.

1.1.1.1 Database modellen

Ved en SQL database kræves der Schema[3], som definerer datamodellen og danner et billede af, hvad og hvordan data skal struktureres også kaldet et table. Ved NoSQL, som er "schema-less", er det op til brugeren at definere en datamodel. NoSQL er mere fleksibelt, når det omhandler ændringer i datamodellen, men dette kræver ansvar hos applikations udviklere til, at modellere data optimalt.

1.1.1.2 Queries[4]

Når data skal hentes, tilbyder SQL database JOIN[5], hvor data fra flere tabler flettes sammen og returneres. Dette har en NoSQL database ikke.

For at få tilsvarende effekt skal NoSQL databaser, query flere gange og selv sammensætte resultatet.

1.1.1.3 Transactions[6]

SQL tilbyder transaction, hvilket er nyttigt, når der arbejdes med flere databaser, der skal opdateres samtidigt. Transaction tillader, at opdatere flere databaser under et funktionskald til serveren. For at opnå tilsvarende effekt på NoSQL databaser, skal dette implementeres via applikations kode.

1.1.1.4 Skalering

Efterhånden som applikationen bliver brugt, vokser mængden af lagret data. Her tages der stilling til om serveren skal skales op eller om den skal skales ud ved at tilføje en ekstra server. Problemmet ved at udvide serveren er, at der kommer et tidspunkt, hvor det vil blive for dyrt at udvide serveren og det vil være bedre at skalere serveren ud.

Fordelen ved NoSQL i forhold til SQL, er at det er lettere at skalere ud. Projektet vil have en

begrænset mængde brugere, så derfor vil mængden af data stige i begrænset omfang. Derfor vil dette ikke have reelt betydning for dette projekt.

1.1.2 Back-end udviklingsværktøjer

I denne sektion vil der beskrives kort om nøglepunkterne om nogle af databaserne, der overvejes.

1.1.2.1 MySQL[7]

En populær database, der benytter sig af SQL.

- ACID[8] operationer understøttes.
- Tilbyder mange query muligheder.

1.1.2.2 Firebase[9]

Firebase indeholder flere databaser til hvert deres formål, som kan vælges efter behov.

- Server løst miljø.
- Databaser med SDK til både android og iOS.

1.1.2.3 MongoDB[10]

En dokument baseret database.

- Hurtigt og simpelt søgning.
- Bygget til at kunne skalere og understøtte fleksibilitet i data modeller.

1.1.3 Valg af back-end database

I denne sektion vil der komme en mere dybdegående beskrivelse af, hvordan Firebase er bygget op, og hvorfor Firebase er valgt til dette projekt.

Firebase[9] er en development platform til mobile applikationer, som Google står bag. Firebase tilbyder forskellige slags funktionaliteter til både Android og iOS applikationer. Herunder beskrives disse produkter fra Firebase, som Rambøll Tilsyn benytter sig af.

For at starte Firebase i Android og iOS, skal man have en google account og oprette et projekt i google. Derefter skal man blot tilføje firebase til ens app samt tilføje de biblioteker, man har behov for. Det giver simpel adgang til data, brugere og filer på grund af de biblioteker, der udstilles.

Firebase databasen blev valgt, fordi der ikke er behov for at sætte en server op, og da Rambøll ikke lægger en server til rådighed, er dette oplagt. Ved valg af en anden database, betyder det at der skal sættes en server op, som kan indeholde databasen i, alternativt vælge at Cloud hoste[11] sin database.

Firebase kræver ikke meget setup at bruge, hvilket var et af fordelene for, hvilken teknologi der skulle bruges til Rambøll Tilsyn. Firebase tilbyder tre forskellige databaser med hvert deres formål.

En af ulemperne med Firebase er prisen, som kan blive dyr hvis applikationen bliver brugt meget. Men da det kun er beregnet som en inhouse applikation med ca. 10 brugere, er dette acceptabelt[12].

Authentication[13]

Firebase Authentication giver en række funktioner indenfor User Management, der gør det let at logge på, oprette og logge af. Firebase Authentication tilbyder en række forskellige muligheder for login fra mail eller et andet sociale medie, som Facebook, Twitter og lign. Brugerdata som e-mail, navn og krypteret kodeord vil blive gemt i Firebase, og derfor vil der ikke være behov for, at Rambøll Danmark A/S skal sørge for at opsætte server med en brugerdatabase, forbindelse til serveren og derefter vedligeholdelse.

Realtime Database[14]

Firebase Realtime Database er et NoSQL database, der gemmer data som JSON[15], og med mulighed for at synkronisering af data med alle som benytter Ramboell Tilsyn, og som har forbindelse til internetet. Databasen er cloud hosted, hvilket betyder, at som udvikler vil give et "serverless"miljø at arbejde med. Dette betyder, at arbejdet med opsætning og vedligeholdelse af server ikke vil være nødvendig.

Hvor databasens indhold kan tilgås via API kald. En af de store fordele herved er, at det er en realtime database, der synchroniser data automatisk. Når der modtages ændringer i data vil alle enheder, der er koblet til internetet, modtage notifikation om ændringen og kan begynde at downloade det.

Firebase Storage[16]

Tilsvarende Firebase Database er Cloud Storage tilegnet for filtyper som video, billeder eller andre tilsvarende data. I dette tilfælde bliver Firebase Storage benyttet til at gemme PDF objekter, som brugeren kan hente og uploade. En vigtigt egenskab for Storage er, at den er bygget til at genoptage download af filer i tilfælde af ustabil internet. Dette kan være tilfældet hos Rambøll, hvis de skal ud til konstruktionsarbejde på en motorvej, hvor forbindelsen er dårlig.

1.2 Applikation

I følgende afsnit gøres der rede for beslutningerne i forhold til udviklingen af applikationen.

1.2.1 Styresystem

Der findes forskellige styresystemer til mobiltelefoner i dag. De to mest brugte[17] er dog iOS [18] og Android. [19]

IOS er Apples eget styresystem, som er udviklet i forhold til deres mobile og tablet enheder.

Android er et stort open source project, som bliver brugt på ca. 80% af alle mobile og tablet enheder i dag. Der findes forskellige versioner af Android styresystemet alt efter, hvilken producent der har leveret enheder[17].

Nogle af de kendte Android enheder leverendører er firmaer som Samsung, Huawei og HTC. De har alle en Android Core i deres styresystemer og har også alle videreudviklet styresystemet, så det passer specifikt til deres enheder.

Rambøll ønskede en applikation udviklet til iOS. Til en workshop med Rambøll fandt vi ud af, at der også var ansatte, som bruger Android. Derfor blev der aftalt at udvikle cross-platform.

1.2.2 Cross-platform udviklingsværktøjer

I denne sektion vil der være en kort beskrivelse af nogle udviklingsværktøjer til cross-platform.

1.2.2.1 Xamarin

Xamarin[20] tilbyder en cross-platform, hvor der udvikles i C#[21] og XAML[22]. Features Xamarin tilbyder:

- Integreret i Visual Studio.
- Fuld API Access til både iOS og Android.
- Deling af kode.

1.2.2.2 PhoneGap

PhoneGap[23] er en platform udviklet af Adobe[24]. Her bruges teknologier som HTML5[25], JavaScripts[26] og CSS[27].

Deres platform tilbyder nogle bonus features som:

- Build server.
- Easy share af applikationen.
- Multiple plug-ins.

1.2.2.3 Corona

Corona[28] tilbyder en cross-platform, hvor der udvikles i Lua scripting [29].

- Real time simulering.
- Live testing.
- Mange plug-in muligheder.

1.2.3 Valg af cross-platform udviklingsværktøj

I følgende afsnit redegøres for de beslutninger, der er taget angående valg af Xamarin.

Der var et lille kendskab til Xamarin fra et tidligere projekt lavet på Ingeniørhøjskolen Aarhus Universitet.

Xamarin giver mulighed for at skrive i C#, som er et sprog, der er blevet undervist i på Ingeniørhøjskolen Aarhus Universitet, som er et højniveau sprog.

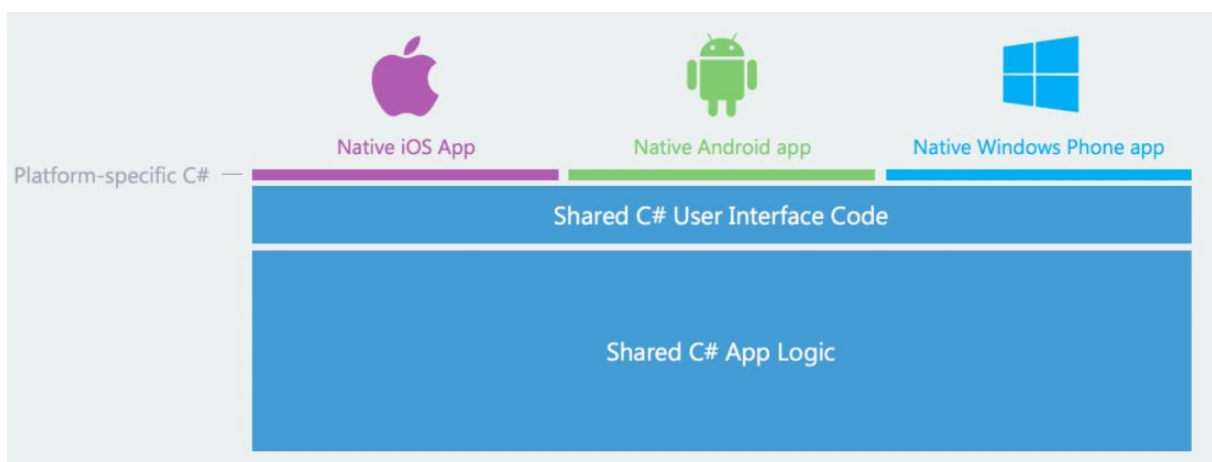
Xamarin blev i februar 2016 opkøbt af Microsoft, hvilket har betydet, at det siden da har været en integreret del af Microsoft Visual Studio, som er det fortrukne udviklingsværktøj.

Udvikler dokumentationen for Xamarin er også et plus. Hvis man sidder med et problem, har Xamarin lavet en meget grundig dokumentation, hvor man kan finde eksempler og guides, til hvordan forskellige dele i en applikation skal programmeres.

1.2.4 Xamarin

Denne sektion vil indeholde en mere dybdegående beskrivelse af, hvordan Xamarin platformen er bygget op.

Xamarin tilbyder en version der hedder Xamarin Forms[30]. Fordelene ved at skrive i Xamarin Forms er, at man kan dele C# kode for de forskellige applikationer. På billedet herunder kan man se, hvordan User Interface koden og App Logic koden er delt på tværs af alle tre enheder. Der skal altså kun skrive sin App Logic og User Interface én gang, og så virker det på både iOS, Android og Windows.

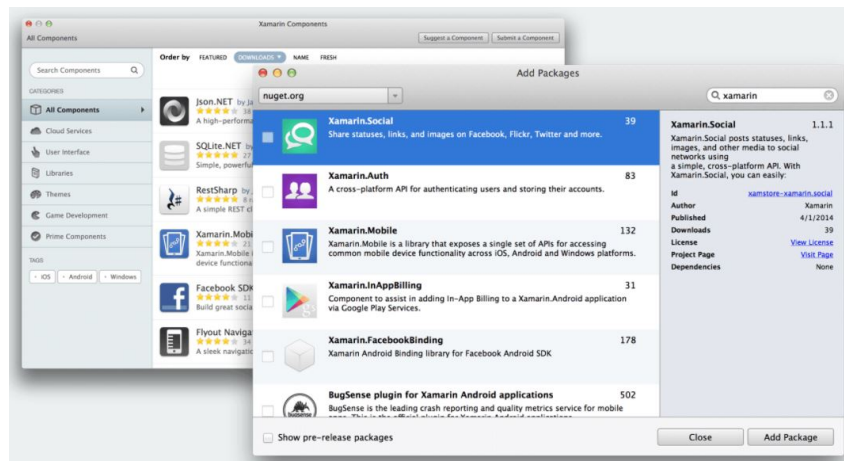


Figur 1.1. Kode deling i Xamarin Forms.[20]

Xamarin kommer både med Xamarin.iOS og Xamarin.Android. Hvad dette gør er, at når koden kompiles, kompiler den ned til native kode. IOS delen vil blive kompileret til ARM assembly, så ens app er native binære platform. Android kompileres, så den kører native Android APK. Så man vil få kompileret koden så den passer til den specifikke enhed, man vil bruge.

Xamarin er baseret på C# og ved hjælp af Xamarin.iOS og Xamarin.Android er det muligt, at kalde og binde det sammen med den eksisterende kode i både iOS og Android, via Xamarins automatiske binding generator.

Der bliver i både Xamarin.iOS og Xamarin.Android givet muligheder for at forbinde til alle API'er, der er nødvendige.



Figur 1.2. NuGet og Xamarin Componet Store.[20]

Xamarin tilbyder også en masse ekstra komponenter, som kan integreres i applikationen. Det kan være alt fra web API'er til sikkerheds elementer.

1.3 Eksportering

I følgende afsnit gøres der rede for beslutningerne i forhold til eksportering af information fra applikationen.

1.3.1 Data eksportering

I følgende afsnit vil der være en beskrivelse af forskellige eksporterings muligheder fra Rambøll Tilsyn.

1.3.1.1 Adobe Acrobat Reader

Portable Document Format[31] (PDF) er et fil-format, der gør det muligt at udveksle dokumenter uafhængigt af software, hardware eller operativ system. Via Adobe Acrobat Reader[32] kan man gratis arbejde med PDF-filer. Programmet giver mulighed for bl.a. at lave kommentarer og markere områder af filen. Det er dog ikke muligt at ændre i teksten, når først filen er i PDF formattet.

1.3.1.2 Microsoft Word

Word[33] er en word processor[34] til redigering af tekst dokumenter. Dette kræver en Microsoft Office pakke. Er der mulighed for at kommentere, opsætte tekst og direkte redigering af tekst, inden man f.eks. udskriver.

1.3.1.3 Microsoft Excel

Excel[33] er et regnearksværktøj til beregninger, tabeller og dataanalyse. Dette kræver en Microsoft Office pakke. Her er der mulighed for at sortere data efter alfabetisk orden, redigere direkte i teksten som i Word.

1.3.2 Valg af data eksportering

Efter nogle overvejelser blev der enighed om at der i Rambøll Tilsyn vil blive eksporteret data til Excel. Dette er valgt, da det giver en masse muligheder i forhold til analyse og sortering i forhold til den information, der bliver trukket ud af Rambøll Tilsyn.

Med valget af at eksportere til Excel, udnyttes de egenskaber som Excel kan og skal ikke selv til at lave sorterings algoritmer. Derved kan der lægges vores fokus andre steder i udviklingen af Rambøll Tilsyn.

Ordforklaring

Forkortelse	Forklaring
Android	Android Operating Sytem
Android Core	Den del af Android som et open source
API	Application Programming Interface
ARM assembly	Kode sprog
Back-end	En betegnelse for database, server og business logik på vores applikation
Firebase	Fælles betegnelse for Firebase database og SDK til firebase
Firebase database	En fælles betegnelse for Firebase Realtime Database, Firebase Authentication og Firebase Storage
Front-end	Det brugeren vil kan se på applikationen
iOS	Apple Operating Sytem
Kompiler	Oversætter kode fra et sprog til et andet
Krypteret	Omdanne data til hemmelig kode
NoSQL	Not Only SQL
Plug-in	Software tilføjelse
PDF	Portable Document Format
Rambøll Tilsyn	Applikationen der implementeres i dette projekt
SDK	Software Development Kit
SQL	Structured Query Language

Litteratur

- [1] Vangie Beal. Sql, . URL <https://www.webopedia.com/TERM/S/SQL.html>. Last Visited d. 07.11.2017.
- [2] Forrest Stroud. Nosql. URL https://www.webopedia.com/TERM/N/nosql_database.html. Last Visited d. 07.11.2017.
- [3] Vangie Beal. Schema, . URL <https://www.webopedia.com/TERM/S/schema.html>. Last Visited d. 07.11.2017.
- [4] Vangie Beal. Query, . URL <https://www.webopedia.com/TERM/Q/query.html>. Last Visited d. 20.10.2017.
- [5] sql join. Join. URL <http://www.sql-join.com/sql-join-types/>. Last Visited d. 20.10.2017.
- [6] Microsoft. Transactions definition, . URL [https://technet.microsoft.com/en-us/library/ms190612\(v=sql.105\).aspx](https://technet.microsoft.com/en-us/library/ms190612(v=sql.105).aspx). Last Visited d. 20.10.2017.
- [7] Mysql. URL <https://www.mysql.com/>.
- [8] Webopedia, .
- [9] Google Firebase. Firebase, . URL <https://firebase.google.com/>. Last Visited d. 20.09.2017.
- [10] mongodb. URL <https://www.mongodb.com/what-is-mongodb>. Last Visited d. 07.11.2017.
- [11] Vangie Beal. Cloud hosting, . URL https://www.webopedia.com/TERM/C/cloud_hosting.html. Last Visited d. 20.10.2017.
- [12] Google Inc. Firebase priser. URL <https://firebase.google.com/pricing/>. Last Visited d. 11.10.2017.
- [13] Firebase. Firebase authentication, . URL <https://firebase.google.com/products/auth/>. Last Visited d. 18.10.2017.
- [14] Firebase. Firebase realtime database, . URL <https://firebase.google.com/products/realtime-database/>. Last Visited d. 18.10.2017.
- [15] JSON. Json. URL <http://www.json.org/>. Last Visited d. 18.10.2017.
- [16] Firebase. Firebase storage, . URL <https://firebase.google.com/products/storage/>. Last Visited d. 19.10.2017.
- [17] IDC. Os. URL <https://www.idc.com/promo/smartphone-market-share/os>. Last Visited d. 27.11.2017.

- [18] Apple. ios. URL <https://www.apple.com/dk/ios/ios-11/>. Last Visited d. 17.10.2017.
- [19] Android. Android. URL <https://www.android.com/>. Last Visited d. 17.10.2017.
- [20] Xamarin. Xamarin platform. URL <https://www.xamarin.com/platform>. Last Visited d. 17.10.2017.
- [21] Webopedia. C sharp, . URL https://www.webopedia.com/TERM/C/C_sharp.html. Last Visited d. 17.10.2017.
- [22] Microsoft. Xaml, . URL <https://msdn.microsoft.com/en-us/library/cc295302.aspx>. Last Visited d. 17.10.2017.
- [23] Adobe. Phonegap, . URL <https://phonegap.com/>. Last Visited d. 08.11.2017.
- [24] Adobe. Adobe, . URL <http://www.adobe.com/dk/>. Last Visited d. 17.10.2017.
- [25] Webopedia. Html, . URL <https://www.webopedia.com/TERM/H/HTML5.html>. Last Visited d. 17.10.2017.
- [26] Webopedia. Javascripts, . URL <https://www.webopedia.com/TERM/J/JavaScript.html>. Last Visited d. 17.10.2017.
- [27] Webopedia. Css, . URL <https://www.webopedia.com/TERM/C/CSS.html>. Last Visited d. 17.10.2017.
- [28] Corona. Corona. URL <https://coronalabs.com/>. Last Visited d. 08.11.2017.
- [29] Lua.org. Lua scripting. URL <https://www.lua.org/about.html>. Last Visited d. 17.10.2017.
- [30] Xamarin. Forms. URL <https://www.xamarin.com/forms>. Last Visited d. 15.11.2017.
- [31] Adobe. Pdf, . URL <https://acrobat.adobe.com/dk/da/acrobat/pdf-reader.html>. Last Visited d. 19.10.2017.
- [32] Adobe. Pdf, . URL <https://acrobat.adobe.com/dk/da/why-adobe/about-adobe-pdf.html>. Last Visited d. 19.10.2017.
- [33] Microsoft Office. Office. URL <https://products.office.com/da-DK/>. Last Visited d. 19.10.2017.
- [34] Computer Hope. Word processor. URL <https://www.computerhope.com/jargon/w/wordssor.htm>. Last Visited d. 19.10.2017.