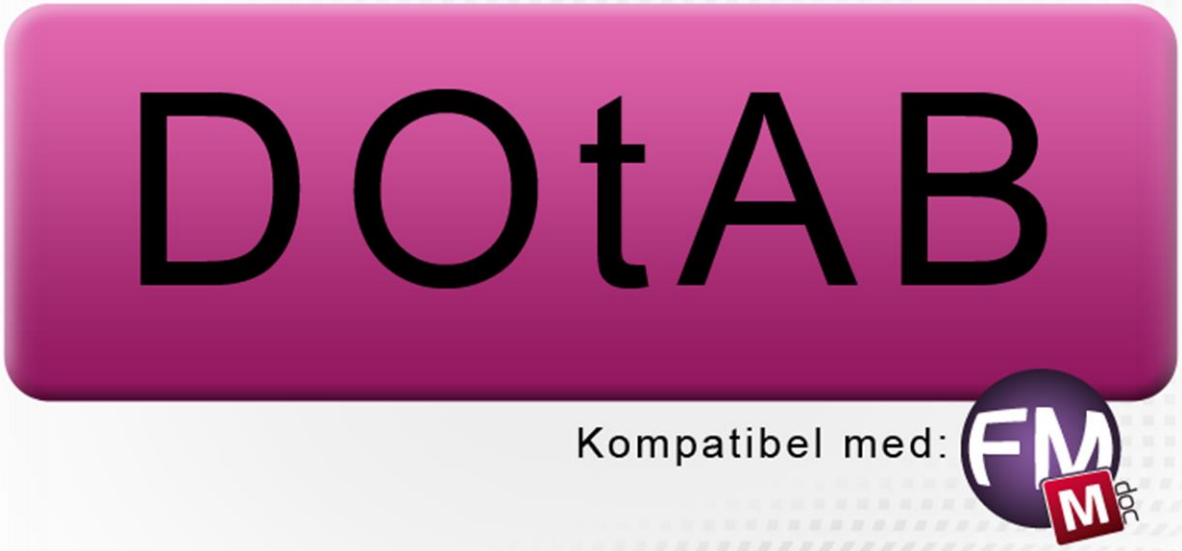
DOtAB

Teknisk rapport



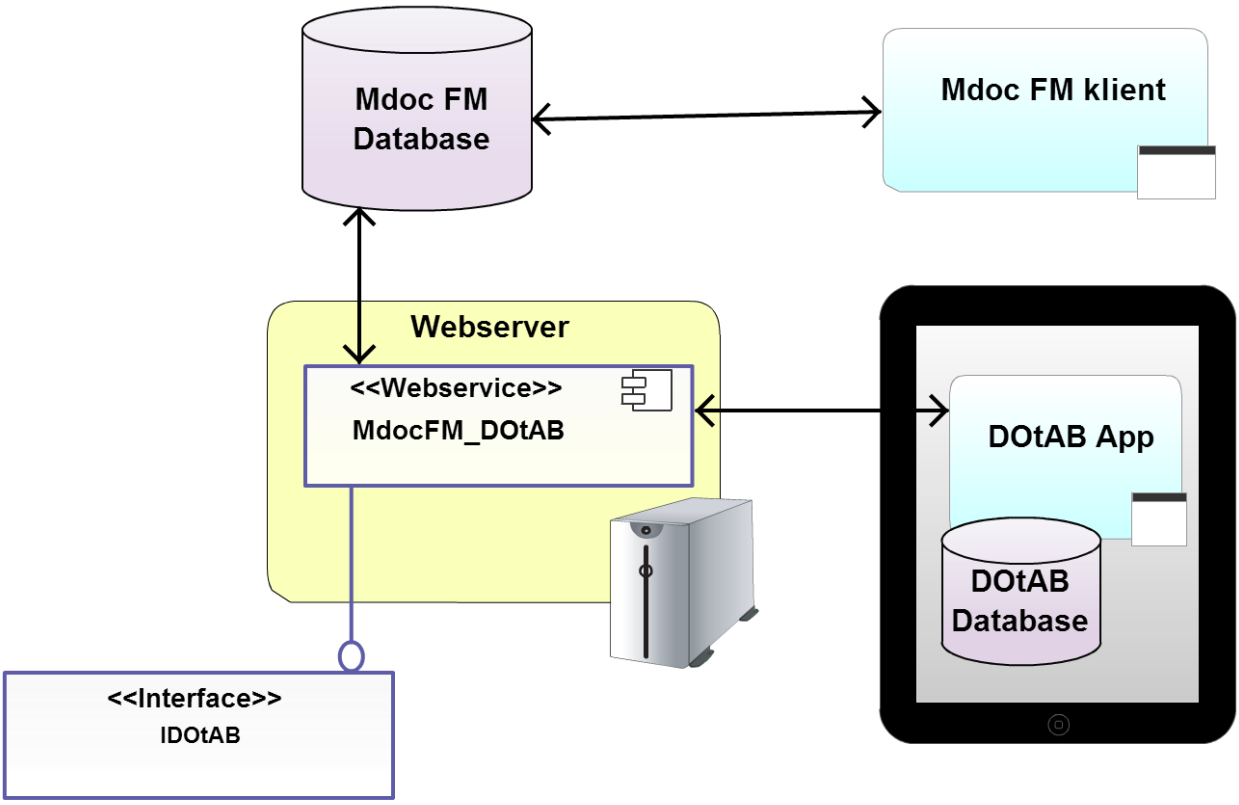
Indholdsfortegnelse

Introduktion

Denne rapport dokumenterer tekniske aspekter for projektet DOtAB (DriftsOptimering til Almene Boliger med Smartphone/Tablet). Rapporten er udfærdiget af NTI CADcenter A/S.

Systemarkitektur

Herunder ses den overordnet systemarkitektur i løsningsmodellen for DOtAB projektet.



Mdoc FM database og Mdoc FM klient var eksisterende elementer som løsningsmodellen skulle anvende som udgangspunkt. Webservicen der hostes på Webserveren er en konkret implementering af interfacet IDOtAB , som passer til Mdoc FM. Denne implementering er brugt under test og evaluering af DOtAB, men er ikke en del af DOtAB projektet.

Teknologier

Platforme for mobile enheder

Der er mange aspekter med hensyn til valg udviklingsmiljø, programmeringssprog og teknologier indenfor udvikling af mobile apps.

Det første valg i forbindelse med DOtAB projektet var at vælge en af to overordnet strategier for udvikling af kode til apps på mobile enheder:

1. Dedikeret kode til en specifik platform
2. Generisk kode som kan afvikles på/kompileres til flere platforme

Det blev hurtigt afklaret at projektet skulle levere apps til både iOS og Android platformene, potentielt flere.

Det ville være et stort arbejde at skulle udvikle de samme ting flere gange til forskellige platforme.

En dedikeret kode til hver platform giver bedre performance men som DOtAB er opbygget er hastigheden ikke det afgørende og programmet afvikles hurtigt og gnidningsfrit på den generiske platform.

Der findes en række udviklingsplatforme hvorpå man kan udvikle generiske apps men fælles for de fleste er at de ikke understøtter hele vejen. F.eks. er JQueryMobile et JavacScript bibliotek der indeholder css mm. til udvikling af mobile apps, men ingen compiler. Phonegap er en compiler der kan fortolke HTML, JavaScript og css filer. Phonegap indeholder dog ingen udviklingsværktøjer.

Det var også vigtigt at udviklingsplatformen understøttede de forskellige enheder på den mobile enhed vi skal bruge i forbindelse med DOtAB projektet. F.eks. skal vi bruge kamera, mikrofon og en database.

Valget faldt på ApplicationCraft da det passede bedst med de krav vi i øvrigt havde og samtidig gav os et drag and drop udviklingsmiljø til at udvikle brugerfladen i. ApplicationCraft samler mange komponenter og værktøjer i et samlet udviklingsmiljø. Bl.a. indgår Phonegap og JQueryMobile som integrerede moduler i miljøet.

På http://en.wikipedia.org/wiki/Multiple_phone_web-based_application_framework kan man i øvrigt se en oversigt over forskellige udviklingsplatform og sammenligne muligheder.

Kommunikations interfacet

Det er vigtigt at snitfladerne specificeres i anerkendte og almene standarder for at imødekomme fremtidig integrationsbehov

Der findes en række standarder for web service interfaces. To af de mest kendte er SOAP/XML og REST/JSON.

SOAP er XML baseret og har typisk få endepunkter og benytter POST metoden. Det er velegnet til computer – computer kommunikation hvor man har gode XML parsere og behov for at kommunikere sikkert over andet end webben.

REST/JSON er velegnet hvor klienten benytter JavaScript. Klienten kan typisk parse JSON direkte til JavaScript objekter.

Interfacet til kommunikation mellem den mobile enhed og FM systemet skal understøtte JavaScript bedst muligt og her er det valgt at bruge JSON som protokol og lave kaldene som REST. Derved undgår vi parsing og url er let at opbygge.

Udviklingsmiljø

Som udviklingsmiljø er ApplicationCraft valgt.

Til definitionen af interfacet samt den konkrete implementering af interfacet er Visual Studio valgt. Mdoc FM er i forvejen programmeret i Visual Studio hvorfor der kunne genbruges komponenter herfra. Desuden er Visual Studio og .NET en udbredt og kendt teknologi. Den understøtter også godt vores behov for webservice der kører REST/JSON.

IDOtAB (service interface)

DOtAB projektet skal definere en neutral web service snitflade således at alle kan tilbyde en bagende som app'en kan køre imod.

Der er samtidig lavet en konkret implementering af dette interface mod Mdoc FM. Denne implementering er brugt til at test og evaluere DOtAB app'en.

Dette er en beskrivelse af denne snitflade samt implementeringen.

Protokol og implementering

Implementeringen af webserveren er opbygget så den både understøtter SOAP/XML klienter og REST/JSON klienter. DOtAB klienten der er bygget til mobile enheder er programmeret i JavaScript og HTML. Af denne grund bruges REST servicedelen da REST/JSON er lettest at bruge i forbindelse med JavaScript og HTML. JavaScript objekter dannes direkte ud fra JSON og HTML kan direkte renderes uden konvertering. Det ville ikke have været tilfældet med SOAP, hvor alle data sendes og modtages som XML og derfor skulle konverteres til JavaScript objekter.

File up/download

Foruden de almindelige servicekald er fil upload og download pakket til et servicekald i stedet for at bruge standard filupload. Det er fordi ApplicationCraft og Phonegap ikke understøttede upload med bruger authentication på udviklingstidspunktet.

Sikkerhed

DOtAB app'en kobler op mod servicen med en servicebruger der henter og synkronisere data. Det er kun service brugeren der kommunikerer med webservicen. De almindelige brugere logger kun ind på DOtAB app'en og anvender offline data. Servicebrugeren logger ind og godkendes med basic authentication. Det anbefales at sikre servicen med et SSL certifikat. DOtAB servicen bruger således almindeligt kendt sikkerhed der kan implementeres på forskellige platforme.

Offline brug

DOtAB er beregnet til at kunne fungere uden netværksforbindelse, således den kan bruges f.eks. i kældre hvor dækningen kan være dårlig.

Derfor er snitfladen bygget til at synkronisere data når den moblie enhed er dokket på kontoret hvor man er sikker på netværksforbindelse. Det er derfor ikke muligt at hente en enkel lokation ud til enheden.

Under synkronisering hentes oplysninger om brugere (employees) ned i DOtAB app'en. Samtidig synkroniseres data for de afdelinger servicebrugeren tilhører således at DOtAB kan bruges offline.

Det er hensigten at flere brugere i samme team kan deles om en enhed. Når en bruger anvender DOtAB, logger vedkommende ind med et personligt brugernavn og password.

Synkroniseringsflow

Når enheden dokkes startes en synkronisering af enheden. Synkroniseringen kan også startes manuelt efter behov.

Som første skridt i synkroniseringen sendes information omkring de oprettede sager med tilhørende billeder til enheden. Det sker ved at sende et objekt med billeder, lyd og tekst ad gangen. Først når hvert objekt er synkroniseret fjernes det fra DOtAB app'en.

Efter data oprettet i app'en er synkroniseret tilbage til FM systemet sendes opdaterede bygningsdata til enheden.

Service beskrivelse

Herunder er listet de enkelte kald webservicen understøtter. Eksemplerne er vist som REST kald og data er pakket som JSON.

Systeminfo

Servicen returnerer informationer omkring bl.a. versioner af webservicen.

Type: Get

Parametre: Ingen.

Retur: Et systemInfo objekt.

Kald: http://localhost/DOtAB/systeminfo

Locations

Servicen returnere en samling af locations. En location er det sted qr koden er relateret til og vil typisk være en lejlighed. Men det kan også være f.eks. en vaskemaskine eller en legeplads.

Samlingen indeholder alle de locations brugeren der tilgår servicen har adgang til.

Type: Get

Parametre: Ingen.

Retur: En samling af locations objekter.

Kald: http://localhost/DOtAB/locations

Frases

Servicen returnerer fraser og opslagsord til forskellige opslagslister. F.eks. vil de typiske fejlbeskrivelser inden for en kategori som bad eller køkken være opført som fraser brugeren kan vælge så der ikke skal skrives mere end højest nødvendig. Også leverandører og standard pris overslag kan hentes som fraser.

I løsningen er der en webside hvor man selv kan redigere og tilføje fraser. De kan derved tilpasses de enkles behov.

Frase kan være afhængig af hvilke bruger der er logget på.

Type: Get

Parametre: Ingen.

Retur: En samling objekter af fraser

Kald: http://localhost/DOtAB/frases

Fraser kan også oprettes og rettes med denne service. Hvis det er en ny frase sættes id til 0. Hvis den skal slettes sættes id til -1.

Type: Post

Parametre: Et frase objekt.

Retur: Et frase objekt med idet på den oprettede frase.

Kald: http://localhost/DOtAB/frases

Case

Servicen opretter en sag i FM systemet..Sagerne der bliver oprettet på enheden sendes tilbage til enheden.

Type: Get

Parameter: Et case objekt.

Retur: Det oprettede case object.

Kald http://localhost/DOtAB/case

Order

Denne funktion sender en ordre til leverandøren der er sat på sagen. Dette sker kun hvis den der har oprettet sagen har tilladelse til at afgive en ordre. Ellers bliver den gemt i FM systemet og kan derfra afsendes. Funktionen kaldes typisk efter at hele sagen med billeder og lydklip samt beskrivelse er uploadet.

Parametre: Et case id.

Retur: En tekst der beskriver hvad der er sket. Kan antage følgende værdier:

- No email address
- No supplier
- Failed
- Order send

Kald: http://localhost/DOtAB/order

Items

Items er de enkelte dele der tilhører en location. Items kan være inddelt i grupper. F.eks. items i køkkenet og items i gangen.

Items kan hentes fra databasen men kan også rettes på enheden og lægges tilbage. Hvis en del f.eks. udskiftes eller der mangler oplysninger kan de indtastes direkte på enheden og senere synkroniseres tilbage i systemet.

Type: Get

Parametre: Ingen.

Retur: En samling af Item

Type: Post

Parametre: En samling af items der skal opdateres.

Retur: En status tekst.

Kald: http://localhost/DOtAB/items

Suppliers

Funktionen henter en liste over alle de leverandører der er tilknyttet til en afdeling.

Type: Get

Parametre: Ingen.

Retur: En samling af supplier

Kald: http://localhost/DOtAB/items

Image

Funktionen henter et billede med et id. Det kan returneres som en stream.

Type: Get

Parametre: Ingen.

Retur: En billede stream.

Kald: http://localhost/DOtAB/items

Imageinfo

Denne funktion henter informationer omkring et billede.

Type: Get

Parametre: Id'et på det billede man ønsker at hente informationer om.

Retur: Et imageinfo objekt.

Employees

Funktionen returnerer alle medarbejdere der tilhører den afdeling service brugeren angiver. Oplysningerne bruges bl.a. til at sikre login på den mobile enhed således at flere brugere kan deles om den. Bruger id bliver tildelt hver oprettet sag således at man kan se hvem der har oprettes hvad.

Type: Get

Parametre: Ingen.

Retur: En samling af employee.

FileInfo

Funktionen opretter en fil location og div. data omkring en fil. Denne funktion skal kaldes før man kan uploade en fil for at få oprettet en plads til filen og få et id der skal bruges når filen skal uploades.

Type: Post

Parametre: FileStore objekt med oplysninger om filen.

Retur: Et fileinfo objekt.

File

Funktionen bruges til at uploade en lydfil. Filen skal først være oprettet med FileInfo funktionen.

Type: Post.

Parametre: id filen skal gemmes under i databasen.

Retur: En tekst der kan antage følgende værdier:

- Succeded
- Failed

Modeler

Herunder er en beskrivelse af de forskellige objekter der indgår I interfacet.

Typerne er .NET typer.

ImageInfo

Indeholder data omkring et billede. Inden et billede uploades oprettes en pladsholder til billedet og der returneres et id der bruges når billedet uploades. Når et billede hentes kan den tilhørende information hentes i et imageinfo objekt.

```
[DataContract]
public class ImageInfo
{
    [DataMember]
    public int Id { get; set; }
    [DataMember]
    public string FileName { get; set; }
    [DataMember]
    public string Title { get; set; }
    [DataMember]
    public string Description { get; set; }
    [DataMember]
    public DateTime? ImageDate { get; set; }
}
```

FileStore

Indeholder informationer om en fil. Inden en fil uploades oprettes en pladsholder til filen og der returneres et id der bruges når filen uploades. Når en fil hentes kan den tilhørende information hentes i et imageinfo objekt.

```
[DataContract]
public class FileStore
{
    [DataMember]
    public int Id { get; set; }
    [DataMember]
    public string RelatedIndexType { get; set; }
    [DataMember]
    public string RelatedIndex { get; set; }
    [DataMember]
    public bool FileEmbedded { get; set; }
    [DataMember]
    public string RelativeFilePath { get; set; }
    [DataMember]
    public string FileName { get; set; }
    [DataMember]
    public bool Rasterex { get; set; }
    [DataMember]
    public DateTime? FileLastModifiedDateTime { get; set; }
    [DataMember]
    public string AddedByIndex { get; set; }
    [DataMember]
    public DateTime? AddedDateTime { get; set; }
```

```
        [DataMember]
        public string NickName { get; set; }
        [DataMember]
        public string Note { get; set; }
        [DataMember]
        public string FileType { get; set; }
        [DataMember]
        public string FileData { get; set; }
    }
```

Location

En location er den enhed hvortil en QR kode er bundet. Typisk en lejlighed men kan også være f.eks. en legeplads eller et vaskerum. Lokationen indeholder typisk en eller flere items. QR koden er guiden i den enkelte lokation.

```
    [DataContract]
    public class Location
    {
        [DataMember]
        public string Guid { get; set; }
        [DataMember]
        public int Id { get; set; }
        [DataMember]
        public string Name { get; set; }
        [DataMember]
        public string Number { get; set; }
    }
```

Items

Et Item er den enhed en sag oprettes på. Der kan tilknyttes forskellige oplysninger til et item. F.eks. garanti oplysninger. Til et item kan der også være filer og billeder der viser og beskriver den.

```
    [DataContract]
    public class Item
    {
        [DataMember]
        public int Id { get; set; }
        [DataMember]
        public int LocationId { get; set; }
        [DataMember]
        public string Group { get; set; }
        [DataMember]
        public string Description { get; set; }
        [DataMember]
        public string Image { get; set; }
        [DataMember]
        public Installer Installer { get; set; }
        [DataMember]
        public Warranty Warranty { get; set; }
        [DataMember]
        public ItemTypeInformation ItemTypeInformation { get; set; }
    }
```

Installer

Information om hvem der har installeret den ting itemet beskriver. Installer er tilknyttet et item.

```
[DataContract]
public class Installer
{
    [DataMember]
    public string CompanyName { get; set; }
    [DataMember]
    public string Address { get; set; }
    [DataMember]
    public string City { get; set; }
    [DataMember]
    public string Phone { get; set; }
}
```

Warranty

Information om garanti på den ting itemet beskriver. Warranty er tilknyttet et item.

```
[DataContract]
public class Warranty
{
    [DataMember]
    public DateTime? WarrantyStart { get; set; }
    [DataMember]
    public int? WarrantyLength { get; set; }
    [DataMember]
    public int? InstalledYear { get; set; }
    [DataMember]
    public int? LifeTime { get; set; }
}
```

ItemTypeInformation

Type information på det item itemTypeInformation er tilknyttet.

```
[DataContract]
public class ItemTypeInformation
{
    [DataMember]
    public string MadeBy { get; set; }
    [DataMember]
    public string Name { get; set; }
    [DataMember]
    public string ItemType { get; set; }
    [DataMember]
    public string SerialNumber { get; set; }
    [DataMember]
    public string Color { get; set; }
}
```

CaseItem

Indeholder informationer om en sag der oprettes på en ting.

```
[DataContract]
public class CaseItem
{
    [DataMember]
    public int Id { get; set; }
    [DataMember]
```

```
        public int EmployeeId { get; set; }
        [DataMember]
        public int LocationId { get; set; }
        [DataMember]
        public int BuildingPartId { get; set; }
        [DataMember]
        public string TextDescription { get; set; }
        [DataMember]
        public int? SupplierId { get; set; }
        [DataMember]
        public bool ContactedByPhone { get; set; }
        [DataMember]
        public string SupplierNote { get; set; }
        [DataMember]
        public string EstimatedCost { get; set; }
        [DataMember]
        public string Access { get; set; }
        [DataMember]
        public DateTime? StartDate { get; set; }
        [DataMember]
        public int? Duration { get; set; }
        [DataMember]
        public string Name { get; set; }
        [DataMember]
        public string Phone { get; set; }
        [DataMember]
        public string Email { get; set; }
    }
```

Frases

Bruges i forbindelse med bruger definerede lister. Forskellige steder i DOtAB optræder lister med forud definerede fraser.

```
    [DataContract]
    public class Frase
    {
        [DataMember]
        public string ItemTypeName { get; set; }
        [DataMember]
        public string Description { get; set; }
    }
```

Suppliers

Indeholder leverandør informationer.

```
    [DataContract]
    public class Supplier
    {
        [DataMember]
        public int Id { get; set; }
        [DataMember]
        public string Name { get; set; }
        [DataMember]
        public string Address { get; set; }
        [DataMember]
        public string City { get; set; }
```

```
        [DataMember]
        public string Phone { get; set; }
        [DataMember]
        public string ContactName { get; set; }
        [DataMember]
        public string Country { get; set; }
        [DataMember]
        public string Zip { get; set; }
        [DataMember]
        public string Email { get; set; }
    }
```

Employees

Indeholder information om de medarbejdere der kan logge på DOtAB.

```
    [DataContract]
    public class Employee
    {
        [DataMember]
        public int EmployeeId { get; set; }
        [DataMember]
        public string EmployeeNo { get; set; }
        [DataMember]
        public string FirstName { get; set; }
        [DataMember]
        public string LastName { get; set; }
        [DataMember]
        public string Password { get; set; }
        [DataMember]
        public string Initials { get; set; }
        [DataMember]
        public bool Active { get; set; }
        [DataMember]
        public string Department { get; set; }
        [DataMember]
        public string Gruppe { get; set; }
        [DataMember]
        public bool? AuthOrders { get; set; }
    }
```

Sysinfo

Indeholder information om versioner på database og service.

```
    [DataContract]
    public class SystemInfo
    {
        [DataMember]
        public int DbVersion = 0;
        [DataMember]
        public string ServiceVersion;
        [DataMember]
        public string Login;
    }
```