

PRODUKTRAPPORT

‘Mange Bække Små’



Resumé:

Produktrapporten beskriver planlægningen af applikationen ‘Mange Bække Små’, en mobilapplikation som skal hjælpe brugeren til at opnå større mental balance og tilfredshed.

Udarbejdet af
Gruppe 6 - Elvin E. A. Jensen
Hold: 1205hf22026p

Titelblad

Projekttitel:	Mange Bække Små
Projektperiode:	21/02-2022 - 29/03-2022
Hovedvejleder:	John Storm Ellehammer
Skole:	TEC, Ballerup
Hold:	1205hf22026p
Projektgruppe:	6
Antal sider:	20 (11,2 normalsider)
Antal bilag:	2

Titelblad	2
1 Læsevejledning	5
1.1 Relevante begreber	5
1.2 Engelsk kode og danske begreber	5
2 Produktets kernefunktioner	6
3 Kravspecifikation	6
3.1 Indledning	6
3.1.1 Formål	7
3.1.2 Et soloprojekt	7
3.2 Generelle beskrivelser	7
3.2.1 Systembeskrivelse	7
3.2.2 Aktørbeskrivelser	8
3.2.3 Programmets individuelle funktioner	9
3.2.4 Programmets begrænsninger	10
3.2.4.1 Generelle begrænsninger	10
3.2.4.2 Begrænsninger ift MBS som et skoleprojekt	10
3.2.5 Produktets fremtid	10
3.2.5.1 Rettelser og optimering af kode	11
3.2.6 Brugerprofil	12
3.2.7 Krav til udviklingsforløbet	12
3.2.7.1 Krav fra udvikleren	12
3.2.7.2 Udviklingsmiljøer, redskaber og frameworks	12
3.2.8 Forudsætninger	13
3.3 Specifikke krav	13
3.3.1 Use-Cases	13
3.3.1.1 Use Case 3: Bruger registrerer en dag	14
3.3.1.2 Use Case 6: Bruger vil slette en tidligere registrering fra sin historik	15
3.4 Eksterne grænseflade-krav	16
3.4.1 Flow chart	16
3.4.2 Entity Relations (databaseopbygning)	17
3.4.3 Bruger-grænseflade	17
3.4.4 Hardware-grænseflade	18
3.4.5 Software-grænseflade	18
3.4.6 Krav til programmets ydelse	18
3.5 Kvalitetsfaktorer	18
3.5.1 Pålidelighed	18
3.5.2 Vedligeholdelsesvenlighed	19
3.5.3 Udvidelsesvenlighed	19

3.5.4 Brugervenlighed	19
3.5.5 Genbrugbarhed	19
3.5.6 Effektivitet	19
3.5.7 Integritet	19
3.6 Øvrige krav	20
3.7 Tidsestimering	20

1 Læsevejledning

Produktet ‘Mange Bække Små’ (fremover refereret til som MBS) beskrives gennem to rapporter. Produktrapporten og procesrapporten.

Produktrapporten beskriver selve produktet og dets funktionaliteter- *hvad* produktet er og gør. Procesrapporten beskriver forløbet omkring udviklingen af produktet- *hvordan* produktet blev til og hvilke teknologier der er benyttet, samt forklaring for valget af disse. Begge rapporter har tilhørende bilag.

1.1 Relevante begreber

MBS: Applikationen ‘Mange Bække Små’.

CRUD: Create, Read, Update, Delete. Bruges typisk ift at manipulere data i en database.

Gyldig email: En gyldig email vil sige at adressen overholder formatet for emails, og derved indeholder både et snabel-a (@) og et punktum (.).

Gyldig kodeord: Et gyldigt kodeord betegnes som en kode på minimum 8 tegn, med mindst ét stort bogstav og ét tal. Brugerens email adresse vil heller ikke betegnes som gyldig.

SQL: Structured Query Language som typisk benyttes ifm databaser.

CI-CD: Continuous Integration-Continuous Deployment. Beskriver den kontinuerlige udvikling og indsættelse af et software produkt gennem “pipelines”, evt. med tilhørende software/unit tests.

REST API: Representational State Transfer Application Programming Interface. Benyttes som “mellemand”, i dette tilfælde, mellem en applikation (MBS) og en server.

IDE: Integrated Development Environment. Det stykke software hvori softwareudviklingen af projektet foregår.

Smiley: En smiley er i kontekst af MBS de tre overordnede kategorier for inddeling af dage. Hhv. god (😊), ok (😐) og skidt (😞).

Påvirkning: En “påvirkning” er i kontekst af MBS underkategorier til inddeling af registrerede dage, med andre ord de ting som har gjort dagen hhv. god, ok eller skidt. Eksempler på disse er karriere, uddannelse, forhold, venner, helbred, søvn m.m.

1.2 Engelsk kode og danske begreber

Da jeg som udvikleren anser det for at være “best practice” at holde variabler m.m. i koden og databasen (inkl. ER-diagrammet) på engelsk for at undgå eventuelle fejl i brugen af unicode standarder ift danske tegn (æ, ø, å), vil jeg for god ordens skyld her specificere hvilke engelske ord/begreber der svarer til hvilke danske ord/begreber.

User: Bruger.

Mood: Det der ovenover er hentydet til som en “smiley” (‘good’, ‘ok’, ‘bad’).

Influence: Det der ovenover er hentydet til som en “påvirkning”.

Main Page: Hovedskærmen hvor brugeren registrerer sin dag.

History: Historik skærmen hvor brugeren kan se/ændre/slette sine tidligere registreringer.

My Stream: Skærmen “Min Å” hvor brugeren kan se sin tendens og største påvirkninger.
The Inner Stream: “Den Indre Å” som sender notifikationer til brugeren.

2 Produktets kernefunktioner

Ved opstart af applikationen får brugeren mulighed for at oprette en konto, eller logge ind med en allerede oprettet konto.

Ved log ind ser brugeren i bunden af skærmen de altid tilstedeværende tre ikoner for henholdsvis “hovetsiden” (‘Registrér’), ‘Min Å’ og ‘Historik’.

Det er ‘hovetsiden’ som brugeren vil se ved log ind/åbning af applikationen, hvor der er mulighed for at registrere sin dag ved at trykke på én af de tre “smileys” (😊😐😞), som repræsenterer at brugeren har oplevet dagen som enten ‘god’, ‘ok’ eller ‘skidt’ (“moods”), samt trykke på/vælge én primær ‘påvirkning’ fra en liste (f.eks. ‘karriere’, ‘uddannelse’, ‘helbred’ osv.) , som har gjort dagen henholdsvis god, ok eller skidt.

Når begge er valgt får brugeren mulighed for at tilføje en note til registreringen, eller blot gemme registreringen.

Ved gemt registrering føres brugeren til siden ‘Min Å’.

Her ses brugerens tendens for registreringer i grafform, samt hvilke ‘påvirkninger’ der er registreret oftest med hhv. positiv og negativ betydning for brugeren (f.eks. kan påvirkningen ‘karriere’ være den oftest registrerede påvirkning på en ‘god’ dag, og ‘helbred’ den påvirkning som er oftest registreret på en ‘skidt’ dag).

I brugerens historik ses en liste af tidligere registrerede dage med dato samt den tilhørende smiley for den pågældende dag. Har den registrerede dag en note vises et ikon ud for denne. Trykker brugeren på en tidligere registrering vises et pop-up vindue med de registrerede informationer (smiley, påvirkning og evt. note), og har mulighed for at ændre eller slette den pågældende registrering.

3 Kravspecifikation

3.1 Indledning

Idéen til MBS er kommer fra og udvikles udelukkende af mig selv. Derfor agerer jeg både som kunde, udvikler og leverandør på projektet.

Programmet og dens tilhørende rapporter og dokumentation er mit svendeprøveprojekt.

3.1.1 Formål

Formålet med Mange Bække Små (MBS) er at hjælpe brugeren til at kunne fokusere på deres succesoplevelser, samt belyse hvilke områder der påvirker dem negativt så disse kan handles på.

For at benytte programmet til fulde skal der foretages daglige registreringer, således at programmet har noget data at kunne basere feedback og overblik ud fra.

Ved internet søgninger finder man andre lignende løsninger, men disse indgår typisk i kontekst af en anden kernefunktion (f.eks. meditationsapplikationer), gemt væk som en underdel af den større applikation.

MBS sigter efter at være simple og hurtigere at benytte, samt undgå kendte “new age/wellness” associerede begreber (se afsnit 3.2.6 [brugerprofil](#) for begrundelse herom).

3.1.2 Et soloprojekt

Ved opstart af svendeprøveforløbet blev det gjort klart at soloprojekter er en absolut undtagelse. Jeg er en af disse.

Grundet et forkortet uddannelsesforløb har jeg ikke fulgt et fast hold gennem min tid på TEC, og har derfor skabt meget få personlige forbindelser. Jeg havde en stående aftale om en tomands-gruppe på benene, men grundet uforudsete omstændigheder på falderebet blev dette ikke en realitet.

3.2 Generelle beskrivelser

3.2.1 Systembeskrivelse

MBS er et stykke software (en mobilapplikation).

Applikationen kommunikerer via et REST API med serveren der opbevarer brugerens data, herunder loginoplysninger og brugerens egne, registrerede data (dage).

Den Indre Å er et notifikationssystem der skal guide brugeren til at benytte sig af MBS, og derved opretholde den gode udvikling samt undgå eventuelle negative tendenser.

Hardwaren der benyttes er brugerens egen enhed (mobiltelefon). Denne skal kunne understøtte nyere mobilapplikationer (Android, iOS) samt forbinde til internettet.

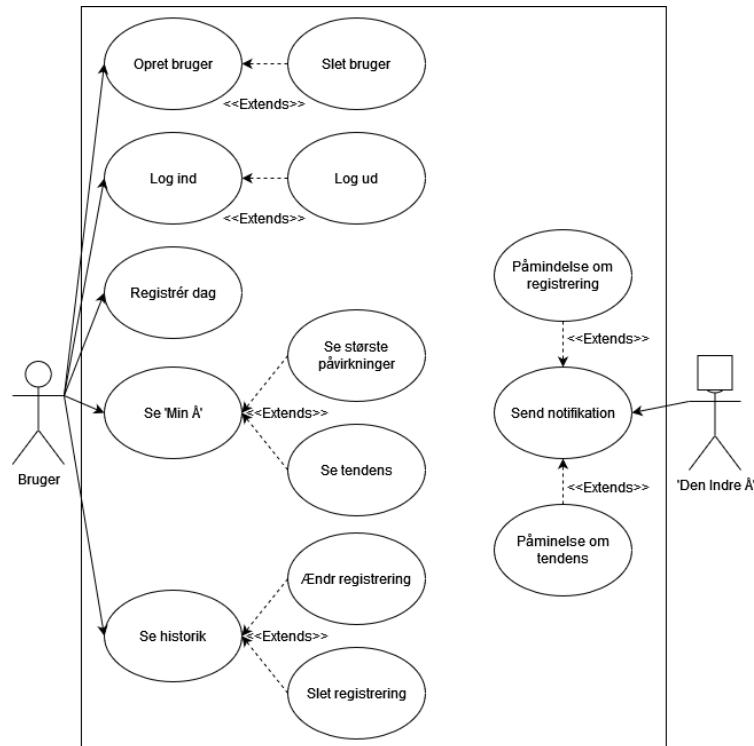
3.2.2 Aktørbeskrivelser

Aktør navn	Bruger
Type (Person/Maskine)	En person der er aktør i MBS
Beskrivelse	En bruger benytter sig af MBS som led i sin søgen efter større mentalt velvære. Brugeren kan: <ul style="list-style-type: none">- Logge ind og ud- Registrere en dag- Se sine tidligere registreringer- Se en overordnet tendens ud fra tidligere registreringer- Ændre eller slette en tidligere registrering- Slette sin konto
Antal samtlige aktører	1-500

Aktør navn	Den Indre Å
Type (Person/Maskine)	Et notifikationssystem der er aktør i MBS
Beskrivelse	Den Indre Å bruger MBS til at motivere brugerne til at registrere sin sindstilstand på daglig basis. Den Indre Å kan: <ul style="list-style-type: none">- Sende påmindelsesnotifikationer- Sende motiverende notifikationer
Antal samtlige aktører	1

3.2.3 Programmets individuelle funktioner

Programmets individuelle, overordnede funktioner er beskrevet gennem Use Cases (se afsnit [3.3.3.2 Use Cases](#)), med udgangspunkt i det følgende use-case diagram (figur 1).

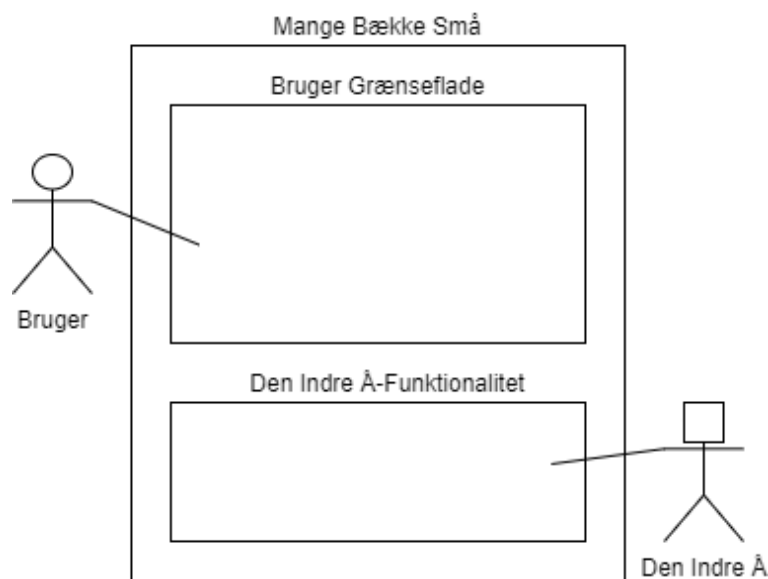


Figur 1.

Da MBS ønsker at have brugerens individuelle rejse som fokus, er diagrammet centreret omkring funktionaliteter tilgængelig for denne.

“Den Indre Å” støtter brugeren i sin rejse gennem notifikationer.

Følgende aktør kontekst-diagram (figur 2) giver et overordnet billede af MBS i sin planlagte iteration.



Figur 2.

3.2.4 Programmets begrænsninger

3.2.4.1 Generelle begrænsninger

Applikationen skal ikke kunne erstatte en personlig, udførlig skrevet dagbog, eller almen sund fornuft omkring livets op- og nedture.

3.2.4.2 Begrænsninger ift MBS som et skoleprojekt

Grundet en kort udviklingsperiode vil visse dele af applikationen blive udeladt af den første iteration, som denne rapport beskriver. Alle dele er naturligvis vigtige for applikationens endelige version, men anses ikke som værende kritiske for kernefunktionaliteterne.

- Brugerprofil og administration af disse

Jeg ønsker at siden 'Min Å' i en senere iteration udvides med yderligere informationer som er relevant for den enkelte bruger, samt mulighed for at kunne slette sin konto og ændre/nulstille kodeord.

- Administratorfunktioner

Skønt systemets opbygning og datadesign laves med øje for fremtidige udvidelser og justeringer, er der ikke planlagt specifikationer for hvordan disse skal se ud og kunne tilgås.

- Den Indre Å

Notifikationssystemet 'Den Indre Å' udelades af første iteration.

- Best practices og GDPR

Mens MBS er i udviklingsfasen uden indblanding af udefrakommende personer, ses der stort på best practices omkring bl.a. CI-CD, implementering af SSL kommunikation til serveren og GDPR. Dette er for at minimere udviklingstiden og fremme hastige iterationer.

3.2.5 Produktets fremtid

Det forventes at applikationen og tilhørende server kan køre på ubestemt tid.

For den individuelle bruger forventes det at applikationen kun er relevant over en periode.

- En animeret å som repræsenterer brugerens tendens, og derved bliver større desto flere positive dage der registreres (inden for en given tidsramme).
- Notifikationssystemet "Den Indre Å", og længere nede ad vejen en smartere version, som er i stand til at genkende mere præcise tendenser gennem noterne skrevet for en registreret dag, og skabe mere præcise, relevante notifikationer ud fra disse.
- "Audio queues" til de ovenstående punkter, taget fra naturen (f.eks. vand og vind).

- Overholdelse af GDPR bliver mere avanceret og relevant desto flere oplysninger om brugeren MBS besidder. Noter bør i den forbindelse hashes på samme måde som brugerens kodeord, da disse potentielt kan indeholde identificerende oplysninger.
- SSL er naturligvis en integral del af ethvert REST API og generel online kommunikation, men under udviklingen hvor alt holdes “in house” skønnes et sådan certifikat at tage længere tid at implementere end hvad der kan forsvares.
- OAuth 2.0 tokens til brugergodkendelse ved login og mellem sessioner.
- På sigt synes jeg som udvikler og teknør, at det kunne være interessant at implementere understøttelse af biometrisk data, fra f.eks. et smarturs evne til at måle stressniveau. En medianværdi ville blive koblet sammen med registrerede dage for at kunne sammenligne urets, forhåbentlige, objektive data med hvordan brugeren har oplevet sin dag.
- Som supplement til det forrige punkt kunne det være ligeledes sjovt at sammenkoble data fra en smartvægt, som kan sammenligne tendenser mellem registreringer af dage og vægt-tab eller -forøgning over tid.

3.2.5.1 Rettelser og optimering af kode

- Klasserne ApiHelper og DataFiller (samt deres implementering adskillige steder i koden) skal “renskrives” som statiske klasser, hvilket bør give bedre performance pga. mindre brug af RAM og CPU-kraft grundet de mange instanser af disse klasser.
- View’ene, og deres tilhørende viewmodels, DayView, HistoryPage og HistoryPagePopup skal “renskrives” efter den mere korrekte implementering af MVVM brugt i MyStream view’et.
- PreLaunch klassen skal “renskrives” med mere “solide” metoder og tasks, som også henter en logget ind brugers historik, samt gemmer dem på en mere hensigtsmæssig måde end den nuværende brug af Get/Set metoder i Xamarin.Essentials.Preferences.
- Større ensartethed i navngivningen af variabler og properties, samt overholdelse af gængse best practices angående tegnsætning (variabler starter med små bogstaver (camelcase), properties og metoder med store (pascalcase) osv.).
- Der skal redigeres i databasens Days tabel. Jeg vil gerne oprette en “composite primary key” således at tabellens primære nøgle bliver skabt ud fra kolonnerne “UserID” og “Date”, hvilket ville gøre at hver bruger kun kan lave én registrering per dag, som efter produktdesignets hensigt (hvilket ikke er tilfældet i den nuværende iteration).

- Rettelse af øvrige bugs registreret på projektets kanban-board (Azure DevOps).

3.2.6 Brugerprofil

MBS er tiltænkt alle og enhver (danskere) som ønsker et bedre overblik over deres (selvopfattede) mentale velvære. Dog er brugere som i forvejen oplever deres hverdag som værende mentalt udmattende/hård en oplagt målgruppe.

For at sikre stor tilgængelighed og brugervenlighed designes applikationens brugergrænseflade ud fra simple, alment brugte designprincipper.

Pga. negative associationer forbundet med “new age/wellness” i visse samfundslag og mennesker, samt for at fremme udbredelsen af applikationen, vil MBS undgå at bruge begreber associeret med denne bevægelse på trods af den åbenlyse inspiration og association hertil.

3.2.7 Krav til udviklingsforløbet

Der arbejdes efter den agile udviklingsmetodik, således at der er plads til gode idéer og smutveje hvis disse skulle opstå. KISS (Keep It Simple, Stupid) og DRY (Don't Repeat Yourself) principperne følges så vidt muligt.

Afviigelser fra den oprindelige kravspecifikation fremgår i afsnit 7.1 af processrapporten.

3.2.7.1 Krav fra udvikleren

- Brugergrænsefladen skal være simpel således at der ikke kræves en brugervejledning.
- Applikationen udvikles i sproget C# med .NET Standard og Xamarin Framework.
- REST API'et udvikles i ASP.NET med .NET 5.0 og .NET Framework Core.
- Serveren kører Windows Server 2016 med SQL Server og API service.
- Kodeord opbevares sikkert vha hashing med tilhørende salt (BCrypt).
- Gyldighed af e-mail og kodeord tjekkes vha “regular expressions”.

3.2.7.2 Udviklingsmiljøer, redskaber og frameworks

Disse redskaber m.m. er blevet brugt ifm. udviklingen af MBS. Nærmere begrundelse vedr. valget af disse er beskrevet i procesrapportens afsnit 6.

- Visual Studio 2022

Denne IDE understøtter mange funktioner som jeg er dybt bekendt med og har størst erfaring med. Al kode vil blive skrevet i denne IDE.

- .NET 5.0

Dette framework er meget alsidigt med sin store understøttelse af mange forskellige class libraries, hvilket gør det velegnet til projektet.

- SQL Server (og SSMS)

SQL Server er en database software løsning som er kompatibel med Entity Framework Core. Jeg har stor erfaring med det tilhørende SQL Server Management Studio.

- Azure DevOps (opgavestyring)

Azure er en cloud baseret udviklingsplatform. Jeg vil bruge dele af dennes DevOps pakke til opgavestyring. Senere i projektet er her gode muligheder for CI-CD samt automatisering af unit tests.

- GitHub (versionsstyring)

GitHub er et versioneringsværktøj. Det er her al kode opbevares og kan sammenlignes fra version til version, samt genkaldes skulle der opstå komplikationer med en iteration.

- Entity Framework Core

Entity Framework Core er et Object Relational Mapping (ORM) værktøj, som tillader at "kortlægge" data hentet fra SQL databasen til de (.NET) modeller der bruges i selve applikationen automatisk. Ved korrekt implementering af dette undgås der også potentielle SQL injections (cyberangreb).

- Xamarin Framework

Xamarin Framework gør det muligt at udvikle mobilapplikationer til flere platforme (bl.a. iOS), hvilket gør det velegnet til dette projekt.

Xamarin.Forms forsimples designprocessen af brugergrænsefladen med et indbygget "shell" og visse delte funktioner de forskellige platforme imellem.

3.2.8 Forudsætninger

MBS forudsætter brugen af en server til SQL og API service.

Da der ikke forventes at gøre brug af hverken avancerede stored procedures eller meget store datamængder på serversiden, vurderes det at denne vil fungere efter hensigten med specifikationer på 2,0 GHz processor, 2GB RAM, 8GB lagringsplads og standard (100 Mbps) NIC og internetopkobling.

Der forventes at brugeren selv leverer den nødvendige hardware til afvikling af applikationen (en mobiltelefon med minimum Android OS v. 9,0 (API Level 28) og med adgang til internettet).

3.3 Specifikke krav

3.3.1 Use-Cases

Hele projektet er opbygget omkring og tager udgangspunkt i de udførte use-cases som beskrevet i use-case diagrammet (se afsnit [3.2.3 Programmets individuelle funktioner](#)).

Inkluderet her er to eksempler på use-cases. Samtlige er at finde i dokumentet "MBS - Bilag - Use-Cases"

3.3.1.1 Use Case 3: Bruger registrerer en dag

Mål	Brugeren ønsker at registrere en dag			
Initiering	Aktiveres af brugeren			
Aktører (Primær, sekundær)	Ekstern bruger			
Type	Selvstændig: X	Include:	Extends:	Specialisering:
Frekvens	2000-3000 pr. uge	Antal samtlige forekomster		X
Ikke-funktion krav	Brugeren skal kunne registrere en dag			
Referencer/Bemærkninger	-			
Startbetingelser	Brugeren er logget ind og står på hovedskærmen			
Beskrivelse	<div>1. Bruger trykker på en smiley (glad, ok, skidt)</div> <div>2. Bruger trykker på en påvirkning</div> <div>3. Bruger trykker på "GEM"</div>			
Variationer	<div>Variation 1:</div> <div>1. Bruger trykker på en påvirkning</div> <div>2. Bruger trykker på en smiley (glad, ok skidt)</div> <div>3. Bruger trykker på "GEM"</div> <div>Variation 2:</div> <div>1. Bruger trykker på en smiley (glad, ok skidt)</div> <div>2. Bruger trykker på en påvirkning</div> <div>3. Bruger indtaster en note</div> <div>4. Bruger trykker på "GEM"</div> <div>Variation 3:</div> <div>1. Bruger trykker på en påvirkning</div> <div>2. Bruger trykker på en smiley (glad, ok skidt)</div> <div>3. Bruger indtaster en note</div> <div>4. Bruger trykker på "GEM"</div>			
Undtagelser	<div>1. Brugeren lader de indtastede data bortfalde ved at trykke på ikonet for “Registrér”, “Min Å” eller “Historik”</div> <div>2. En meddelelse informerer brugeren om dette</div> <div>3. Brugeren bedes bekræfte sit valg ved at gentage trykket</div> <div>4. Brugeren sendes til den valgte side</div>			

Slutresultat	Succes	Brugeren har registreret sin dag og sendes til skærmen “Min Å”.
	Fejl	Brugeren har ikke registreret sin dag og sendes tilbage til hovedskærmen. En relevant fejlmeddelelse vises.

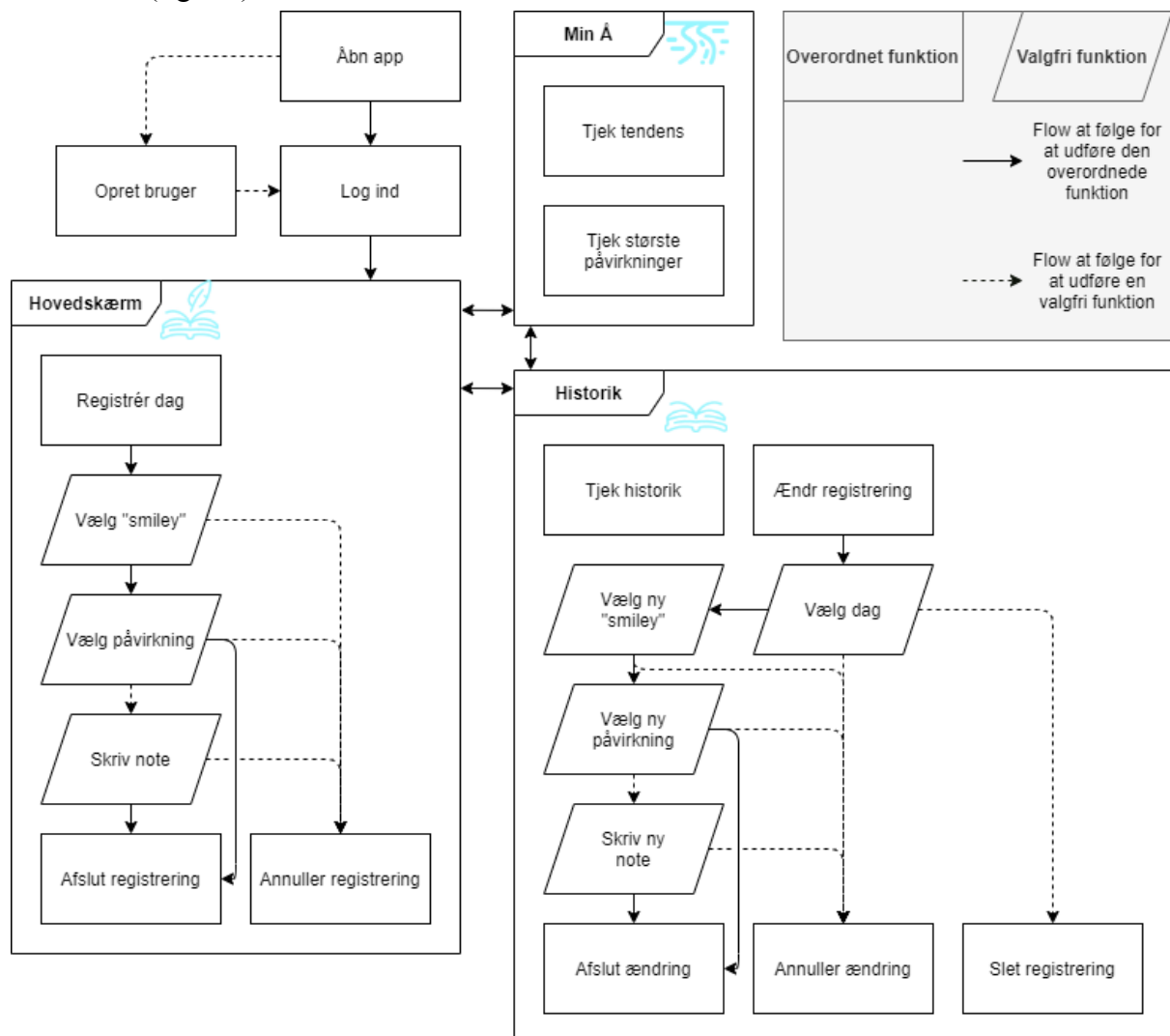
3.3.1.2 Use Case 6: Bruger vil slette en tidligere registrering fra sin historik

Mål	Brugeren ønsker at slette en tidligere registrering			
Initiering	Aktiveres af brugeren			
Aktører (Primær, sekundær)	Ekstern bruger			
Type	Selvstændig: X			
Frekvens	5-10 pr. uge			
Ikke-funktion krav	Brugeren skal kunne slette en tidligere registrering			
Referencer/Bemærkninger	-			
Startbetingelser	Brugeren er logget ind på sin konto og står på sin historik			
Beskrivelse	<ol style="list-style-type: none"> 1. Bruger trykker på en tidligere registrering 2. Bruger trykker på “Slet” 3. Bruger bedes bekræfte sletning 4. Bruger bekræfter ved at trykke på knappen igen 			
Variationer	-			

3.4 Eksterne grænseflade-krav

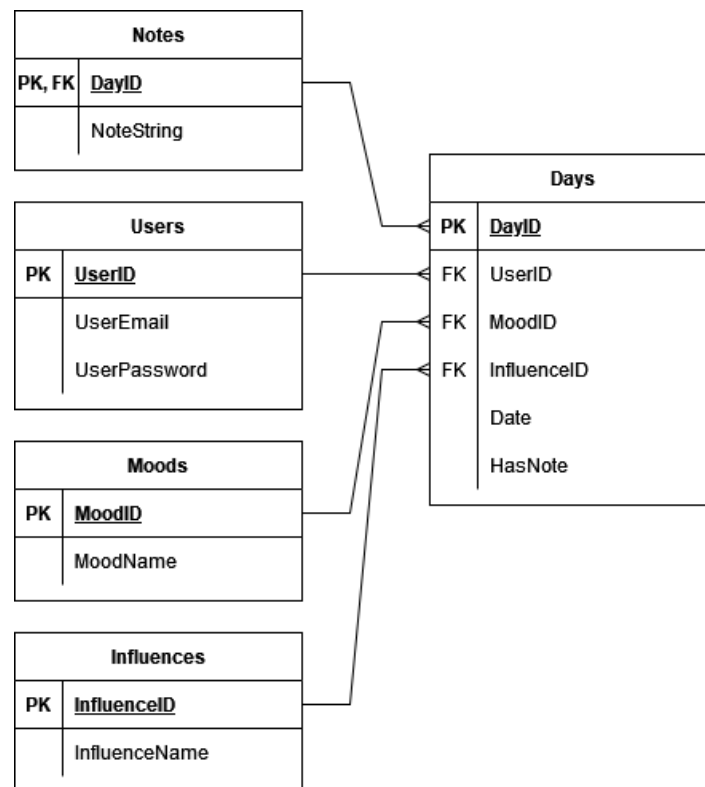
3.4.1 Flow chart

Flow chart (figur 3) beskriver visuelt flowet af de kernefunktioner der er beskrevet i [afsnit 2](#).



Figur 3.

3.4.2 Entity Relations (databaseopbygning)



Figur 4.

Tabellerne i databasen (figur 4) har en simpel og funktionel opbygning. Tabellerne Notes, Users, Moods og Influences har hver sin egen primary key, der kobles sammen med en foreign key reference i Days-tabellen, med undtagelse af Notes-tabellen hvor dens primary key også er foreign key til primary key'en i Days-tabellen.

3.4.3 Bruger-grænseflade

Da MBS afvikles på en mobiltelefon betjenes applikationen med fingrene. Bruger-grænsefladen designes med øje for dette og vil derfor have sine funktioner placeret i bunden af skærmen, så brugeren kan tilgå disse hurtigt med få tryk.

Brugeren vil have CRUD adgang til sine registrerede data gennem grænsefladen.

Da der kun findes én slags bruger, vil der kun blive lavet én bruger-grænseflade til applikationen.

Den Indre Å har rettigheder til at sende notifikationer til brugerens enhed.

Wireframes blev udviklet i starten af projektet. Følgende er et eksempel på "hovedsiden" (figur 5) og siden 'Min Å' (figur 6).

Samtlige wireframes for projektet er inkluderet under bilaget "MBS - Bilag - Wireframes".



Figur 5.



Figur 6.

3.4.4 Hardware-grænseflade

Serveren skal kunne give et svar på under 800 ms.

3.4.5 Software-grænseflade

På serveren køres der en Windows Server 2016 installation, som leverer SQL Server og REST API tjenester.

Applikationen modtager og konsumerer data fra REST API'et.

3.4.6 Krav til programmets ydelse

Et skærbillede (ekskl. loadingskærmen ved applikationens opstart) skal kunne vises på under 1 sekund, inkl. svartid fra serveren.

3.5 Kvalitetsfaktorer

Hver kvalitetsfaktor vurderes på en skala fra 1 til 5, med 1 værende lavest prioritet og 5 højest.

3.5.1 Pålidelighed

Vurdering: 4

Udvikleren står selv for hosting, deployment og vedligeholdelse. Oppe-tid af systemet skal være over 95%.

3.5.2 Vedligeholdelsesvenlighed

Vurdering: 4

Systemet er designet med øje for vedligeholdelsesvenlighed med et minimum af indblanding fra udviklerens side.

Vedligeholdelse må ikke overskride 3 timer om ugen.

Nede-tiden må ikke overskride de ovennævnte 5%.

3.5.3 Udvidelsesvenlighed

Vurdering: 2

Ved levering vil de grundlæggende features af applikationen være funktionelle og kunne opfylde de fastsatte krav.

Med kompleksiteten af de planlagte udvidelser taget i betragtning vil udviklingsprisen stige.

Disse udvidelser vil dog efter release ikke være et fastsat mål med en fastsat deadline.

3.5.4 Brugervenlighed

Vurdering: 5

Da der intet forudfattet kendskab forventes til brug af applikationen, er en høj brugervenlighed et krav.

Det er et krav at brugergrænsefladen designes efter alment brugte designprincipper, som straks kan genkendes og benyttes af brugeren.

3.5.5 Genbrugbarhed

Vurdering: 2

Udvikleren forventer en genbrugelighed af kode på 25-30%.

Dette kommer til udtryk når brugeren redigerer en tidligere registreret dag, som forventes at benytte sig af samme kode som selve registreringen af en dag, samt modeller som går igen.

Objekt orienterede-programmeringspraksisser benyttes hvor muligt.

3.5.6 Effektivitet

Vurdering: 3

Som beskrevet tidligere kræves en effektivitet af programmet således at det kan vise et skærbillede (inkl. svartid fra server) på under 1 sekund.

3.5.7 Integritet

Vurdering: 3

Det er et krav at lagerenheder på server back-enden er opsat med RAID 1, således at integriteten af brugerregistreret data bevares i tilfælde af hardware-relaterede fejl.

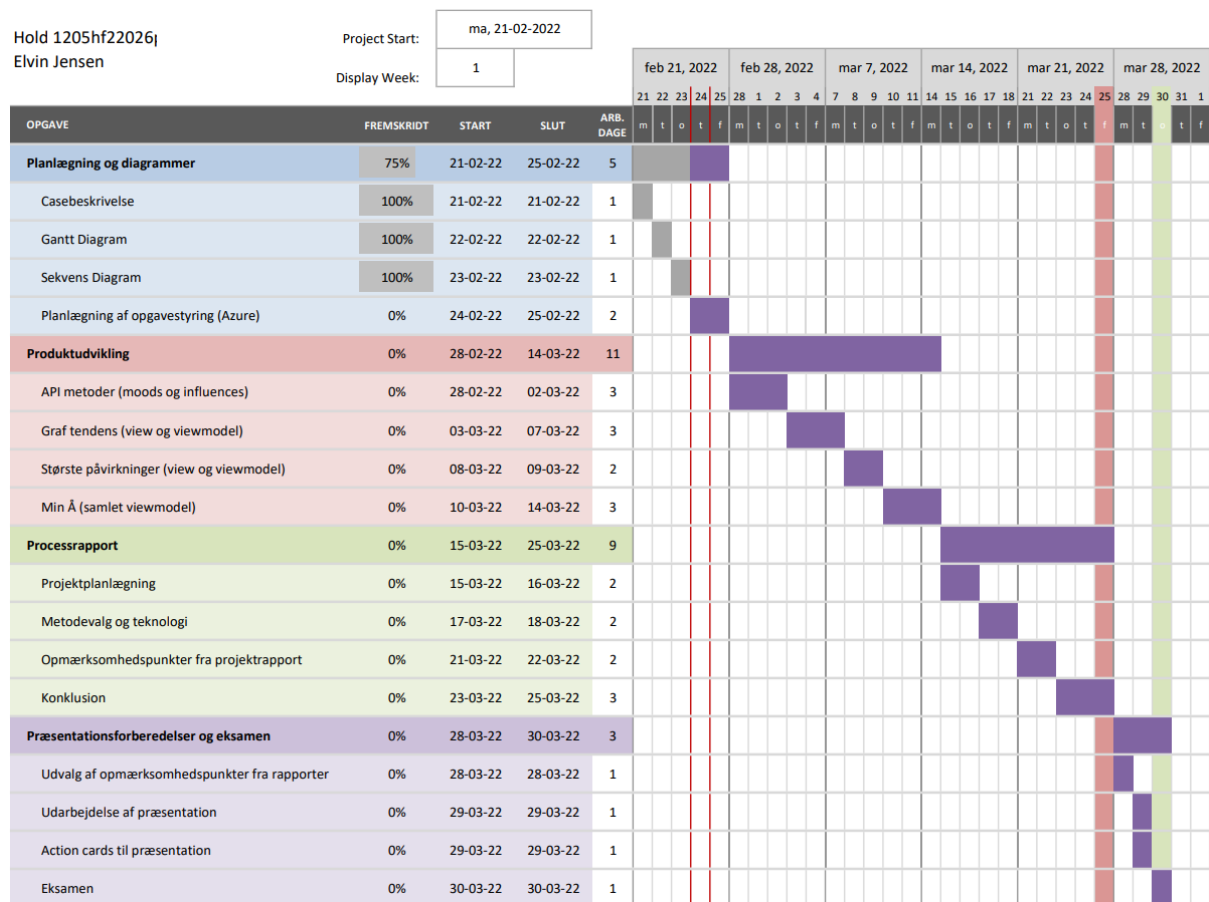
Brugerregistreret data opbevares i overensstemmelse med godkendte sikkerhedsstandarder.

3.6 Øvrige krav

- Beta-testning af applikationen vil foregå på arbejdspladsen med dummy-data, med mulighed for udvidelse til andre personer/brugere efter vurdering fra udvikleren.
- Al testning udført af udvikleren foregår udelukkende i arbejdstiden.

3.7 Tidsestimering

Min tidsestimering (figur 7) sætter en stram plan der ikke kan viges fra hvis målet skal nås til tiden.



Figur 7.