

**FØRSTEÅRSPRØVE**

**Datamatikeruddannelsen**

**DMVE231, 2.semester**

**Foråret 2024**

**Titel:** IT-system til LawHouse

**Opgavestillere:**

Henrik Steen Krogh  
Brian Soltau  
Christina Rehmeier  
Per Larsen

**Udlevering:** 19.04.2024 kl. 12.00 i WISEFLOW  
**Afleveres:** 31.05.2024 kl. 12.00 i WISEFLOW.

## 1. Indledning

### Prøvens tilrettelæggelse

Prøven er en ekstern mundtlig gruppeprøve på baggrund af et skriftligt gruppeprojekt og bedømmes efter 7-trinsskalaen.

Gruppen må være på 3-4 studerende. Afvigelser herfra kræver en dispensation.

Der gives én samlet individuel karakter ud fra en helhedsvurdering af den skriftlige og den mundtlige præstation. Projektet præsenteres af projektgruppen, max. 30 min. Herefter individuel eksamination af gruppens medlemmer. Der afsættes her 30 min. pr. eksaminand inkl. votering.

Projekt opgaven skal munde ud i en projektrapport der skal afleveres til bedømmelse af jeres undervisere og en ekstern censor.

Rapporten der afleveres, skal være i formatet **pdf**. Sourcekode, eksekverbare programmer, database og lignende skal zippes inden det uploades, der må ikke være zip-filer inde i zip-filen.

Det kan anbefales, at man i denne projektperiode arbejder på skolen fra kl. 9.00-15.30. Jeres undervisere/vejledere i fagene vil kunne konsulteres i de normalt skemasatte lektioner eller efter nærmere aftale.

**VIGTIGT:** Sørg for at disponere tiden således, at alle delopgaver besvares og sørg for at prioritere opgaverne i henhold til casen.

## 2. Vedrørende rapporten

### Formkrav til den skriftlige projektrapport

Følgende elementer skal indgå:

**Forside** med titel. Yderligere skal forsiden indeholde: eksamens-termin, skolens navn, klassebetegnelsen og gruppemedlemmernes navne.

#### Indholdsfortegnelse

**Indledning**, inkl. præsentation af problemstilling, problemformulering og tilgangsvinkler

**Opgaveløsningen** (proces og produkt)

**Konklusion** husk, der skal være en sammenhæng mellem indledningen og konklusionen. De to skal i princippet kunne forstås, uden at læse baggrunds-og analyseafsnittene

#### Perspektivering

**Litteraturliste** (medtag alle kilder, der er lavet henvisninger til i projektet - også fra nettet)

**Bilag** (inkluder kun bilag, som er centrale for rapporten)

Projektet skal som minimum fylde 10-normalsider og maksimum 30-normalsider. For hver studerende, som deltager i projektet, bliver det maksimale sideantal udvidet med 5 normalsider.

En normalside er 2400 tegn inkl. mellemrum og fodnoter. Forside, indholdsfortegnelse, litteraturliste samt bilag tæller ikke med i antal sider. Bilag er uden for bedømmelse.

Det skal fremgå af rapporten hvilket gruppemedlem der har haft ansvaret for det enkelte afsnit. Tilsvarende skal det være muligt at se de enkelte medlemmers bidrag til koden, f.eks. via git.

## 3. Fagenes ca. vægtning i opgaven

Fagene vægtes som de indgår i ects fordelingen på hele 1. år. Det vil sige:

Virksomhed: 10/60

Systemudvikling: 15/60

Programmering/teknologi: 35/60

## 4. Bedømmelsen af besvarelsen

Ved bedømmelsen af opgavebesvarelsen lægges bl.a. vægt på:

- Korrekt brug af de faglige begreber og termer der er indlært i fagene (fagsprog).
- Sammenhæng mellem rapportens enkelte dele (sporbarhed).
- Argumentation for programmernes opbygning, struktur mv.
- Argumentation for databasens opbygning og afkobling til resten.
- Argumentation for hvordan systemet giver værdi for virksomheden og brugerne.

## 5. Opgavebeskrivelse

Den vedlagte CASE (se bilag) beskriver dele af arbejdsgangene i LawHouse. Prøven tager udgangspunkt i en fiktiv virksomhed – men problemstillingerne er realistiske nok, f.eks. <https://www.bechbruun.com/>, <https://ideahlaw.com/>, <https://www.codexlaw.dk/da/> og <https://familieadvokaten.dk/>. Der findes endvidere softwarehuse der har specialiseret sig i software til advokatbranchen. F.eks. Advosys, der udbydes af UniK System Design Vejle.

Opgaverne er undertiden integrerede, hvorfor en opgavebeskrivelse under et fagområde ikke altid kan besvares uden at se den i sammenhæng med øvrige fagområders opgavebeskrivelser. Tilsvarende overlapper visse opgavebeskrivelser hinanden.

### Virksomheden.

LawHouse lægger vægt på at de udarbejdede systemforslag igennem en agil process (Ex. SCRUM) for at sikre de ideer bliver udviklet, skaber mest mulig værdi samt forbedre arbejdsgangen.

1. Udarbejd forslag til ekstra features til LawHouse som kan forbedre deres arbejdsgange (Skal ikke nødvendigvis implementeres, men der skal reflekteres over ideerne)
2. Forslagene skal dokumenteres igennem eventuelt Epic User Stories eller almindelige User stories
3. Hvis disse features implementeres skal disse fremhæves, hvis ikke skal disse features beskrives i opgaven under et separat afsnit om udvikling

### Projektstyring.

I denne del skal I demonstrere anvendelsen af agile værktøjer til projektet.

1. Med henblik på brug af en agil metode (ex. SCRUM) skal der udarbejdes en backlog med User Stories for alle features, for at en potentiel Product Owner kan se hvad der skal udarbejdes. Denne backlog skal prioriteres og dokumenteres via screenshots.
2. User Stories for hver features skal være formuleret korrekt efter User Stories principperne.
3. Når product backloggen er udarbejdet skal I estimere efter bedste evne hvor mange sprints I skal bruge til projektet for at løse opgaven. Dette dokumenteres i et afsnit omkring projektledelse.
4. Til dette systemforslag efterspørger ledelsen en risiko analyse, for at sikre at projektet har en ide om hvilke risici der involveret ved udviklingen af dette projekt. Denne risikoanalyse skal dokumenteres i afsnittet omkring projektledelse.

## Systemudvikling

Dan et overblik over forretningsområdet ved hjælp af en domænemodel. Der skal fokuseres på journalisering/indberetningsdelen af det samlede system.

Anvend use cases mv. til at modellere de funktionelle krav undervejs i iterationerne. Foreslå endvidere en række ikke-funktionelle krav til systemet, med udgangspunkt i gennemlæsning af case-beskrivelsen. Er der krav til afkobling af UI f.eks.? Det er vigtigt at I undervejs som udviklere får afklaret kerneområderne i applikationen, detaljerne omkring applikationen samt bidrager med input til teknologiske løsninger.

Designet af de mest væsentlige use cases skal dokumenteres. Dokumentationens skal som minimum indeholde et designklasse-diagram, eksempler på systemsekvensdiagrammer og sekvensdiagrammer samt et brugervenligt, funktionelt og fleksibelt brugergrænsefladedesign. Et endeligt designklassediagram over systemet skal medfølge.

GRASP-mønstre bør anvendes i designet og dokumenteres via eksempler, f.eks. via ansvarsfordeling på klasserne. Andre mønstre benyttes efter behov til etablering af fleksibilitet, afkobling, udskiftelighed mv.

Det er også vigtigt, at der etableres en fornuftig arkitektur af systemet. Denne arkitektur skal naturligvis også kunne genkendes og forefindes i implementeringen i programmerne.

Som en lille detalje skal der til systemet implementeres et simpelt hjælpesystem hvor man for den enkelte form skal kunne få en "hjælpefunktion" åbnet således at man får en vejledning til hvordan det enkelte skærbillede fungerer. Hjælpefilerne til de enkelte forms kan være pdf-filer.

Det er naturligvis tilladt at anvende yderligere metoder og teknikker ud over de nævnte.

Hvis der i CASE-beskrivelsen forekommer ting, der er unøjagtige eller tvetydigheder etc., kan projektgruppen gøre sig sine egne (relevante) forudsætninger, med eventuelle udvidelser eller indskrænkninger.

GitHub skal anvendes til versionsstyring og teamsamarbejde.

## Database

1. Systemets nødvendige data skal kunne implementeres i en SQL-database. Derfor udform et E/R diagram, omform det til tabeller og implementer databasen.
2. Væsentlige SQL-kommandoer i tilgangen til denne database skal dokumenteres. Dette kan ske via sammenhæng med C# programmeringen.

## Programmering

LawHouse prioritering af opgaverne besluttet af gruppen iht Scrum-processen.

Den præcise opsætning og vedligehold af stamdata aftales idet det dog forventes at der programmeres eksempler på CRUD-programmer.

I hele opgaven er der fokus på diverse anvendte collections og design patterns samt at den valgte arkitektur er at finde i implementeringen. I bedømmelsen af programmerne indgår derfor også en kvalitetsmæssig bedømmelse.

Programmeringssproget skal være C# under anvendelse af Windows Forms i MS Visual Studio, idet fremtidige versioner dog forventes at skulle være web-baserede. Dette senere skift skal indtænkes udviklingen, - i særlig grad for selvbetjeningsdelen.

Som en del af opgaveløsningen skal der vises eksempler på anvendelse af UNIT-test.

Dokumentation af programmerne skal bestå af udvalgte kommenterede programudsnit i selve projektrapporten, eventuelt med referencer over i programkoden i bilagsmaterialet. Den interessante source-kode er naturligvis den egenproducerede source-kode og ikke den autogenererede source-kode fra værktøjsmiljøet

Programmernes funktionalitet **skal altid** eksplicit dokumenteres via indsatte skærbilleder fra systemet og eksempler på udskrifter/filer dannet fra systemet i selve rapporten, altså ikke kun som bilag. Et screendump er ikke dokumentation men kan bruges til at understøtte jeres forklaring.

I koden kan anvendes Entity framework til access af den designede database, men tilgang til database via SqlClient skal kodes for mindst 2 tabeller i systemet.

Da den konkrete database ikke kendes skal der implementeres fornuftig afkobling til forretningslogikken fra det valgte persistenslag. Det gælder uanset om man anvender Entity Framework eller SqlClient.

Som del af lagdelingen skal der implementeres en beregningsalgoritme, nævnt under casen nedenfor.

## Teknologi

LawHouse har stor fokus på at deres App kører async igennem hele arkitekturen. 'MultiThreading' skal håndteres på en sådan måde at der ikke er mulighed for at igangsætte flere ens tråde samtidig. F. eks: Et buttonklik disables button indtil tråden er kørt færdig.

## **BILAG**

### **CASE-beskrivelse: Advokatfirmaet: LawHouse**

Firmaet LawHouse er et større advokatfirma der oprindeligt er startet i grundlæggerens tagetage. Firmaets primære arbejdsopgaver er salg af advokatydelser (skilsmisssager, bolighandel osv.) men efterhånden som firmaet nu er vokset til 8 advokater og 3 advokatsekretærer er der nu et STORT behov for at automatisere arbejdsgangene omkring specielt registrering af ydelser på de enkelte sager, så der kommer mere styr på pengeflowet i virksomheden. Endvidere er der i virksomheden et behov for at kunne styre efteruddannelse af advokaterne således, at man altid kan se hvem af advokaterne der har dybtgående viden f.eks. indenfor skatteret eller EU-ret.

Der foreligger følgende om virksomhedens arbejdsopgaver:  
I virksomheden sælges der flere forskellige ydelser. Eksempler på ydelser er det advokatarbejde, der er i forbindelse med skatteret, bolighandel eller skilsmisssager. Typisk er assistancen ved en bolighandel en fast pris pr. handel, hvor en skilsmisssag er baseret på timer, der faktureres til klienten. Specialopgaver som f.eks. rådgivning om generationsskifte i en virksomhed faktureres altid på timebasis.

Derudover er der et voksende behov for selvbetjening, - f.eks. at udføre beregninger og for at købe formularer som kunderne selv kan udfylde, - f.eks. omkring samlivspakker, testamente, ægtepagt m.v.

For at drage fordel af dette salg af ydelser på nettet påtænker i tilgift at etablere et bibliotek af beregningsalgoritmer, som abonnenter gratis kan anvende. Se evt <https://familieadvokaten.dk/beregninger.asp>, som har implementeret en del heraf.

Som eksempel på beregningsalgoritme implementeres en låneberegner, som her:

### Beregning af ydelse på lån

Lånets størrelse	<input type="text" value="51.000,00"/>
Rentefod p.a.	<input type="text" value="6,5"/>
Antal ydelser pr. år?	<input type="text" value="1"/>
Antal år	<input type="text" value="15"/>



Fast ydelse: 5.423,99 kr.  
Fast ydelse pr. år: 5.423,99 kr.

Formel:

$$y = G \frac{r}{1 - (1 + r)^{-n}}$$

Betegnelser:  $G$  er hovedstolen (det lånte beløb).  $r$  er rentefoden pr. termin, dvs. renten som decimalbrøk (f.eks.  $8\% = 0,08$ ),  $y$  er ydelsen og  $n$  antallet af terminer.

Den årlige ydelse der skal betales på et annuitetslån hvor der lånes 51.000 kr, renten er 6,5% p.a. (pr. år) og løbetiden er 15 år er, -

$$G = 51\ 000, \quad r = 0,065, \quad n = 15$$

$$y = G \frac{r}{1 - (1 + r)^{-n}} = 51000 \frac{0,065}{1 - (1 + 0,065)^{-15}} = 5424$$

Da den en årlige ydelse på det 15-årige lån er 5424 kr., betaler vi i alt  $15 * 5424 = 81360$  totalt.

For alle advokatydelser gælder, at ydelsen har et nummer og et navn. For nogle af tjenesteydelserne opereres der som sagt med et fast honorar for ydelsen, medens specialopgaverne er timebaserede. Hver gang der arbejdes på en sag, skal timeforbruget registreres. Specielt skal al kørsel registreres og viderefaktureres til klienten.

Virksomheden ønsker en nøje styring af deres klienter. Om klienterne ønsker man bl.a. at gemme klientnummer, navn, adresse, eventuelle telefonnumre. Det er vigtigt for virksomheden hele tiden at have en klar oversigt over kundernes sager og de ydelser, herunder abonnementer, man har gennemført for klienten.



Når en klient henvender sig til LawHouse, oprettes der en sag - til denne sag tilføjes der så efterhånden ydelser, der registreres med art, dato samt antal timer. Såfremt der til en arbejdsopgave som f.eks. bolighandel, aftales et andet honorar end listeprisen, skal denne aftalte pris naturligvis registreres.

Vi afgrænser os i første omgang fra selve faktureringsdelen af systemet, men alle nødvendige oplysninger, der senere skal anvendes hertil, skal gemmes på sagen samt sagens ydelser.

Når klienten henvender sig, oprettes der (som nævnt tidligere) en sag. Sagen får bl.a. tildelt et sagsnummer og en arbejdstitel ligesom startdato og aftalt færdiggørelsestidspunkt (foreløbigt) registreres. Til sagen kan også gemmes et estimeret antal timer, der kan medgå til alle tjenesteydelserne, som skal indgå i sagen. Desuden knyttes der altid en advokat til sagen, som ansvarlig for sagsforløbet. Selvbetjeningssager eller køb af dokumenter kan afvige datamæssigt fra ovennævnte efter behov.

I forbindelse med bemanding af de enkelte sager skal der gemmes visse oplysninger om medarbejderne. Ud over medarbejdernummer, navn osv. ønsker man også at registrere, hvilke tjenesteydelser medarbejderen er i stand til at levere.

For at være i stand til at følge op på opgaven skal der gemmes oplysninger om timeforbruget pr. medarbejder pr. tjenesteydelse på den enkelte sag.

Det er endvidere et ønske fra LawHouse at der for en kunde kan dannes en "sigende" rapport over registrerede timer på en given sag. Denne rapport skal kunne udskrives på en tekstfil.