**Bilag 4**

**Kravspecifikation**

Finance Tracker

Semesterprojekt 4

Diplomingeniøruddannelsen i softwareteknologiA blue logo with dolphins and a anchor

AI-generated content may be incorrect.

|  |  |
| --- | --- |
| **Navn:** | **Studienummer:** |
| Ahmad Chiha | 202305734 |
| Ali Najafi | 202307397 |
| André Pelle Rashid | 202306343 |
| Christina Lavdal Braüner | 202306583 |
| Jahye Ali | 202309135 |
| John Nguyen | 202209849 |
| Khaled Rami Omar | 202307853 |
| Khizer Khan | 201710674 |
| Victoria Franca Edwards | 202308509 |
|  |  |
| **Vejleder:** |  |
| Jung Min Kim |  |
|  |  |
| **Bilagsansvarlig:** |  |
| Alle |  |
|  |  |
| **Beskrivelse:** |  |

I dette dokument præsenteres de krav, der stilles til løsningen, og hvordan de prioriteres. Fokus er på både funktionelle krav (user stories) og ikke-funktionelle krav. Accepttestspecifikation og testresultater findes i Bilag 5.

**Versionshistorik**:

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Dato** | **Initialer** | **Beskrivelse** |
| 0.1 | 12/2/25 | CLB | Dokument oprettet. |
| 1.0 | 14/2/25 | CLB | Indsat fra drev, med rettelser |
| 1.1 | 18/2/25 | CLB, VE | Rettelser af krav, ud fra inputs fra vejleder. |
| 1.2 | 25/2/25 | CLB, VE | Rettelser efter opdatering og ændringer. |
| 1.3 | 26/2/25 | CLB | Tilføjelse af IFK samt prioritering af krav vha. MoSCoW og FURPS+. |
| 1.4 | 22/5/25 | CLB | Rettelser og fjernet accepttest. |

Indholdsfortegnelse

[User Stories 3](#_Toc198821622)

[US1. Løn-estimering 3](#_Toc198821623)

[US2. Lønsammenligning 3](#_Toc198821624)

[US3. Tillægstider 3](#_Toc198821625)

[US4. Platformskompatibilitet 3](#_Toc198821626)

[US5. Oversigt over årsindkomst og SU-fribeløb 4](#_Toc198821627)

[US6. Advarsel ved overskridelse af SU-fribeløb 4](#_Toc198821628)

[US7. Brugerprofil 4](#_Toc198821629)

[US8. Registrering af arbejdsvagter 4](#_Toc198821630)

[US9. Eksport af lønestimering til PDF 4](#_Toc198821631)

[US10. Feriepengeberegning 5](#_Toc198821632)

[Ikke-funktionelle krav 5](#_Toc198821633)

[Priotering af funktionalitet 5](#_Toc198821634)

[MoSCoW for funktionelle krav baseret på user stories 6](#_Toc198821635)

[FURPS+ tabel med MoSCoW-prioritering for ikke-funktionelle krav 6](#_Toc198821636)

[Afgrænsning 7](#_Toc198821637)

[Konklusion 7](#_Toc198821638)

[Litteratur 8](#_Toc198821639)

# **User Stories**

User stories er valgt og beskriver brugerens behov samt ønsker til systemet. Hver historie indeholder en kortfattet beskrivelse af, hvem der udfører en handling, hvad de vil opnå, og hvorfor det er vigtigt. Formålet er at gøre kravene mere forståelige og sikre, at løsningen leverer værdi til brugeren. User stories er formuleret baseret på Gherkin.[1]

## **US1. Løn-estimering**

**Som** bruger

**ønsker** jeg at se en løn- estimering for en lønperiode,

**så** jeg kan planlægge min økonomi.

## **US2. Lønsammenligning**

**Som** bruger

**ønsker** jeg at systemet sammenligner min lønseddel med lønestimeringen,

**så** jeg bliver opmærksom på afvigelser.

## **US3. Tillægstider**

**Som** bruger

**ønsker** jeg at registrere tillæg for forskellige tidspunkter og ugedage,

**så** applikationen kan medtage dem i løn-estimeringen.

## **US4. Platformskompatibilitet**

**Som** bruger

**ønsker** jeg at tilgå applikationen via en webapp og en native app,

**så** jeg kan bruge appen i webbrowser eller som en lokal app.

## **US5. Oversigt over årsindkomst og SU-fribeløb**

**Som** bruger

**ønsker** jeg at se en oversigt over min indkomst for året og resterende SU-fribeløb,

**så** jeg kan holde overblik over min økonomi.

## **US6. Advarsel ved overskridelse af SU-fribeløb**

**Som** bruger

**ønsker** jeg at modtage en advarsel, hvis jeg nærmer mig mit SU-fribeløb,

**så** jeg kan reducere min arbejdstid og undgå tilbagebetaling af SU.

## **US7. Brugerprofil**

**Som** bruger

**ønsker** jeg at oprette en brugerprofil

**så** jeg kan gemme mine personlige oplysninger.

## **US8. Registrering af arbejdsvagter**

**Som** bruger

**ønsker** jeg at registrere mine arbejdsvagter,

**så** applikationen kan beregne en lønestimering.

## **US9. Eksport af lønestimering til PDF**

**Som** bruger

**ønsker** jeg at kunne eksportere min løn-estimation til en PDF-fil,

**så** jeg kan gemme den offline.

## **US10. Feriepengeberegning**

**Som** bruger

**ønsker** jeg at kunne se mine optjente feriepenge for et år,

**så** jeg kan få overblik over mine feriepenge.

# **Ikke-funktionelle krav**

De ikke-funktionelle krav fastsætter rammerne for systemets kvalitet, stabilitet og brugeroplevelse. De beskriver ikke, hvad systemet gør, men hvordan det gør det – eksempelvis med hensyn til ydeevne, sikkerhed, tilgængelighed og vedligeholdelse. Mange af kravene er inspireret af anerkendte principper inden for brugeroplevelse (UX) og interface-design (UI).

Nogle af kravene er teknisk målbare og kan verificeres via test eller værktøjer, som f.eks. responstid, sikkerhedsscore eller API-load. Andre krav – såsom overholdelse af GDPR eller farvekontraster – er i højere grad baseret på retningslinjer, principper og holdningsbårne valg. De kan ikke nødvendigvis måles præcist, men anses stadig som væsentlige for at sikre et ansvarligt og tilgængeligt system.

De følgende ikke-funktionelle krav er samlet og nummereret med henvisninger til relevante kilder.

1. Responstiden for en formularsendelse må ikke overstige **2 sekunder**.[2]
2. Generering af et månedligt lønestimat må ikke tage mere end **5 sekunder**.[2]
3. REST API’et skal kunne håndtere op til **5000 samtidige requests per minut** uden at returnere HTTP 5xx-fejl.[3]
4. Alle moduler i softwaren skal have mindst **80% code coverage**.[4]
5. Webapplikationen skal opnå en sikkerhedsscore på **medium** eller under ifølge ZAP-skala.[5]
6. Applikationen skal understøtte både **dansk og engelsk**.[6]
7. Farverne i UI’en skal sikre kontrastforholdet på minimum **4.5:1** mellem tekst og baggrund.[7]
8. Automatisk backup af brugerdata skal ske hver **24. time**.[8]
9. Systemet skal overholde **GDPR og nationale databeskyttelseslove**.[9]
10. Systemet skal understøtte **automatiserede softwareopdateringer uden nedetid**.[10]

# **Priotering af funktionalitet**

For at sikre fokus på de vigtigste funktioner i udviklingsprocessen er alle krav blevet prioriteret ud fra deres betydning for brugeren og projektets mål. Prioriteringen har dannet grundlag for udvælgelsen af funktioner i MVP’en og har hjulpet teamet med at træffe bevidste valg under udvikling og test.

## **MoSCoW for funktionelle krav baseret på user stories**

For at prioritere de funktionelle krav er MoSCoW-metoden anvendt. Denne metode opdeler krav i fire kategorier – Must have, Should have, Could have og Won’t have – og giver et struktureret overblik over, hvilke funktioner der er nødvendige for Minimum Viable Product (MVP), og hvilke der kan udskydes til fremtidige versioner.

|  |  |
| --- | --- |
| **Prioritet** | **Krav** |
| **Must have** | • **US1.** Lønestimering  • **US2.** Lønsammenligning  • **US3.** Tillægstider  • **US4.**  Platformskompatibilitet  • **US7.** Brugerprofil  • **US8.** Registrering af arbejdsvagter |
| **Should have** | • **US9.** Eksport af løn-estimation til PDF  • **US10.** Feriepengeberegning |
| **Could have** | **• US5.** Oversigt over årsindkomst og SU-fribeløb |
| **Won’t have** | • **US6.**  Advarsel ved overskridelse af SU-fribeløb |

Tabel 1, Prioritering af user stories med MoSCoW-metoden

## **FURPS+ tabel med MoSCoW-prioritering for ikke-funktionelle krav**

De ikke-funktionelle krav er kategoriseret og prioriteret ud fra FURPS+-modellen, som dækker funktionalitet (Functionality), brugervenlighed (Usability), pålidelighed (Reliability), performance (Performance) og vedligeholdelsesvenlighed (Supportability) samt udvidede kvalitetskriterier. Ved at kombinere FURPS+ med MoSCoW-metoden sikres en balanceret prioritering af systemets kvalitetskrav.

|  |  |  |
| --- | --- | --- |
| **Kategori** | **Ikke-funktionelle krav** | **Prioritet** |
| **Functionality** | **IFK6** - Applikationen skal understøtte både dansk og engelsk. | **Should have** |
| **Usability** | **IFK7** - Farverne i UI’en skal sikre kontrastforholdet på minimum 4.5:1 mellem tekst og baggrund. | **Should have** |
| **Reliability** | **IFK4** - Alle moduler i softwaren skal have mindst 80% code coverage. | **Could have** |
| **IFK5** - Webapplikationen skal opnå en sikkerhedsscore på medium eller under ifølge ZAP-skala. | **Must have** |
| **IFK8** - Automatisk backup af brugerdata skal ske hver 24. time. | **Should have** |
| **Performance** | **IFK1** - Responstiden for en brugerhandling skal ikke  overstige 2 sekunder. | **Must have** |
| **IFK2** - Generering af et månedligt lønestimat må ikke tage mere end 5 sekunder. | **Must have** |
| **IFK3** - REST API’et skal kunne håndtere op til 5000 samtidige requests per minut uden at returnere HTTP 5xx-fejl. | **Must have** |
| **Supportability** | **IFK9** - Systemet skal overholde GDPR og nationale databeskyttelseslove. | **Could have** |
| **IFK10** - Systemet skal understøtte automatiserede softwareopdateringer uden nedetid. | **Won’t have** |

Tabel 2, FURPS+ for ikke-funktionelle krav med MoSCoW-prioritering

# **Afgrænsning**

Der er foretaget bevidste afgrænsninger baseret på prioritering. Kun funktioner klassificeret som “Must have” er fuldt implementeret i MVP. “Should have” og “Could have” er planlagt som potentielle udvidelser ved ressourcemæssigt overskud.

# **Konklusion**

Kravspecifikationen har dannet et struktureret grundlag for udviklingsarbejdet og sikret et klart fokus på brugerværdi og kvalitet. Kravene er formuleret med udgangspunkt i brugerbehov og afstemt mod tekniske rammer og projektets ressourcer.

# **Litteratur**

[1] Cucumber.io, Gherkin Reference, 2024. [Online]. Available: https://cucumber.io/docs/gherkin/reference/. [Accessed: 21-May-2025].

[2] J. Nielsen, Time Scales in UX, 2023. [Online]. Available: https://jakobnielsenphd.substack.com/p/time-scale-ux. [Accessed: 21-May-2025].

[3] Testfully, API Load Testing Guide, 2024. [Online]. Available: https://testfully.io/blog/api-load-testing/. [Accessed: 21-May-2025].

[4] Microsoft Docs, Using Code Coverage in Visual Studio, 2024. [Online]. Available: https://learn.microsoft.com/en-us/visualstudio/test/using-code-coverage. [Accessed: 21-May-2025].

[5] OWASP, Risk Rating Methodology, 2024. [Online]. Available: https://owasp.org/www-community/OWASP\_Risk\_Rating\_Methodology. [Accessed: 21-May-2025].

[6] Apple Inc., Supporting Multiple Languages in Your App, 2024. [Online]. Available: https://developer.apple.com/documentation/xcode/supporting-multiple-languages-in-your-app. [Accessed: 21-May-2025].

[7] Medium Design Bootcamp, The Ultimate Guide to Contrast Ratio in UI Design, 2023. [Online]. Available: https://medium.com/design-bootcamp/the-ultimate-guide-to-contrast-ratio-in-ui-design-2502d42442c4. [Accessed: 21-May-2025].

[8] Proactive Info, Data Backup Best Practices, 2023. [Online]. Available: https://www.proactive-info.com/blog/data-backup. [Accessed: 21-May-2025].

[9] CookieYes, GDPR Principles: A Complete Guide, 2024. [Online]. Available: https://www.cookieyes.com/blog/gdpr-principles/. [Accessed: 21-May-2025].

[10] Microsoft Tech Community, Automating Software Updates, 2024. [Online]. Available: https://techcommunity.microsoft.com/blog/coreinfrastructureandsecurityblog/automating-software-updates/323183. [Accessed: 21-May-2025].