

## Case management System (ärendehanteringssystem)

### Projektidé (Scenario):

Projektet handlar om att utveckla ett **ärendehanteringssystem** (Case Management System, CMS) som hjälper organisationer att hantera och följa upp olika typer av ärenden på ett strukturerat och effektivt sätt. Systemet ska vara webbaserat (web-based) och erbjuda funktioner som ärendeskapande, tilldelning, statusuppföljning, notifikationer och rapportgenerering.

### Bakgrund och Syfte (på enkel svenska)

Många företag och organisationer hanterar ärenden på ett krångligt sätt – antingen manuellt eller i system som inte är välorganiserade. Detta gör att arbetet tar längre tid, blir rörigt och svårt att följa upp.

Ett digitalt system för ärendehantering kan lösa dessa problem genom att:

- Automatisera arbetsflöden så att ärenden hanteras snabbare
- Göra det lättare för användare att samarbeta.
- Se till att alla ärenden hanteras på rätt sätt enligt regler och riktlinjer

**Målet med projektet** är att skapa en modern, säker och enkel plattform som hjälper organisationer att arbeta effektivare och hålla hög kvalitet i sin ärendehantering.

### Vision & Mål

#### Vision/Problemformulering:

Lösningen ska vara ett **enkelt, smidigt och säkert system** för att hantera ärenden.

Systemet ska hjälpa organisationer att **skapa, spåra och lösa ärenden** på ett ordnat sätt. Det ska minska **manuellt arbete**, förbättra **kommunikationen mellan användare** och se till att alla ärenden hanteras **på rätt sätt och i tid**.

Systemet är till för **företag, myndigheter och andra organisationer** som arbetar med många ärenden, t.ex. kundservice, HR-avdelningar och supportteam. Genom att samla alla ärenden på ett ställe och automatisera delar av arbetet ska systemet göra det **snabbare och enklare att hantera ärenden, spara tid och ge bättre service**.

## Tydliga mål

### *Vad ska levereras? (Funktioner)*

Projektet ska resultera i ett **webbaserat system för ärendehantering** med följande funktioner:

#### ✓ Användarkonton och inloggning:

- Användare ska kunna logga in med en **säker metod** (t.ex. tvåfaktorsautentisering).
- Det ska finnas olika användarroller, t.ex. **administratörer, handläggare och kunder**.

#### ✓ Hantering av ärenden:

- Användare ska kunna **skapa, redigera och avsluta ärenden**.
- Ärenden ska kunna få en **prioritet och kategori**.
- Det ska finnas en **historik** för varje ärende.

#### ✓ Notifieringar och kommunikation:

- Användare ska få **e-post eller SMS** vid viktiga ändringar.
- Man ska kunna skriva kommentarer och **bifoga filer** i ärenden.

#### ✓ Rapporter och statistik:

- Systemet ska kunna skapa **rapporter** om ärenden.
- Data ska kunna **exporteras som PDF eller Excel**.

### ✓ Säkerhet och regler:

- Systemet ska **skydda användarnas information** enligt GDPR.
- Automatiska **säkerhetskopior** ska göras regelbundet.

### *Hur vet vi om projektet är lyckat?*

#### ✦ Fungerar som planerat:

- Alla funktioner ska finnas och **fungera utan stora problem**.
- Användare ska **enkelt kunna använda systemet**.

#### ✦ Användarna ska vara nöjda:

- Minst **80% av testpersonerna** ska tycka att systemet är **enkelt och bra att använda**.
- Företag som använder systemet ska **spara tid** och arbeta mer effektivt.

#### ✦ Systemet ska vara snabbt och stabilt:

- Det ska klara **minst 10 000 användare samtidigt**.
- Att öppna ett ärende ska ta **mindre än 3 sekunder**.

#### ✦ Säkerhet och regler:

- Systemet ska följa **GDPR** och andra viktiga regler.
- Inga **känsliga uppgifter** får läcka vid säkerhetstester.

## Intressenter och deras rolle

Intressent	Roll i projektet	Intresse i systemet
Kund/Organisation	Beställare	Vill ha ett system som effektiviserar ärendehantering, minskar manuell administration och följer regler.
Administratörer	Slutanvändare	Ansvarar för att skapa och hantera användarkonton, behörigheter och övervaka systemet.
Handläggare	Slutanvändare	Behöver ett enkelt och snabbt sätt att hantera, spåra och lösa ärenden.
Kunder/Medborgare	Slutanvändare	Skapar och följer upp sina ärenden. Vill ha tydlig kommunikation och statusuppdateringar.
Utvecklingsteam	Utvecklare, testare	Ansvarar för att bygga och testa systemet så att det fungerar enligt kravspecifikationen.
Dataskyddsansvarig	Rådgivare	Säkerställer att systemet följer GDPR och andra säkerhetskrav.
Ledning och chefer	Beslutsfattare	Vill ha rapporter och analyser för att kunna fatta datadrivna beslut.
Supportteam	Slutanvändare, support	Behöver kunna hjälpa användare och hantera tekniska problem i systemet.

### 3. Kravspecifikation

#### *Funktionella krav*

Här är en lista över funktionella krav som systemet ska uppfylla:

#### **1. Användarautentisering och behörigheter**

- a. Systemet ska tillåta användare att logga in med en säker metod (t.ex. tvåfaktorsautentisering).
- b. Användare ska ha olika roller (administratör, handläggare, kund) med olika behörigheter.

#### **2. Ärendehantering**

- a. Användare ska kunna skapa, redigera och radera ärenden.
- b. Ärenden ska kunna tilldelas specifika handläggare.
- c. Varje ärende ska ha en status (öppet, pågående, avslutat).

#### **3. Notifieringar och kommunikation**

- a. Systemet ska skicka automatiska notifieringar via e-post och/eller SMS vid statusändringar.
- b. Användare ska kunna kommentera och bifoga filer till ett ärende.

#### **4. Sök- och filtreringsfunktion**

- a. Användare ska kunna söka efter ärenden baserat på ID, status, datum och handläggare.
- b. Systemet ska erbjuda filtrering av ärenden efter kategori och prioritet.

#### **5. Rapportering och analys**

- a. Systemet ska kunna generera rapporter över antalet ärenden, handläggningstider och slutförda ärenden.
- b. Rapporter ska kunna exporteras i PDF- och Excel-format.

#### **6. Åtkomstkontroll och säkerhet**

- a. Systemet ska säkerställa att användare endast har åtkomst till information som är relevant för deras roll.
- b. Alla användaraktiviteter ska loggas för spårbarhet.

#### **7. Mobilanpassning**

- a. Systemet ska vara responsivt och fungera på både datorer, surfplattor och mobiltelefoner.

## Icke-funktionella krav (enkelt förklarat)

### 1. Systemet ska vara snabbt och klara många användare

- Systemet ska kunna hantera **minst 10 000 användare samtidigt** utan att bli långsamt.
  - Att öppna en sida eller ett ärende ska ta **max 3 sekunder**.
1. **Varför är detta viktigt?**  
Om systemet är långsamt blir det jobbigt att använda, och det kan ta för lång tid att lösa ärenden.
  2. **Hur löser vi detta?**  
Genom att använda **snabba servrar, bra databaser och smarta tekniska lösningar**.

### 2. Systemet måste vara säkert

- Användardata ska vara **krypterad** så att ingen obehörig kan läsa den.
  - Endast rätt personer ska kunna se eller ändra information genom **behörigheter**
  - Systemet måste följa **GDPR** (lagar om dataskydd).
1. **Varför är detta viktigt?**  
Det skyddar privat information och förhindrar att någon får tillgång till känsliga uppgifter.
  2. **Hur löser vi detta?**  
Genom att använda **starka lösenord, säker inloggning (t.ex. tvåfaktorsautentisering) och dataskyddsteknik**.

### 3. Systemet ska alltid fungera

- Systemet ska vara tillgängligt **99,9% av tiden**, så att det inte plötsligt slutar fungera.
- Om något går fel ska det finnas **säkerhetskopior** av all information.

1. **Varför är detta viktigt?**

Om systemet kraschar kan viktiga ärenden försvinna, vilket skapar stora problem.

2. **Hur löser vi detta?**

Genom att använda **säkra servrar och göra regelbundna säkerhetskopior**.

#### 4. Systemet ska vara enkelt att använda

- Gränssnittet ska vara **enkelt och lätt att förstå**.
- Det ska fungera både på **dator, mobil och surfplatta**.

1. **Varför är detta viktigt?**

Om systemet är krångligt att använda, tar det längre tid att arbeta och kan skapa misstag.

2. **Hur löser vi detta?**

Genom att skapa en **tydlig design och testa systemet med riktiga användare**.

#### 5. Systemet ska logga alla viktiga händelser

- Alla inloggningar, ändringar och viktiga händelser ska sparas i en **logg**.
- Loggarna ska sparas i **minst 12 månader**.

1. **Varför är detta viktigt?**

Om något går fel, eller om någon gör något olämpligt, måste det finnas **spårbarhet**.

2. **Hur löser vi detta?**

Genom att **lagra loggar automatiskt och ge administratörer tillgång till dem**.

## Must Have (M) – Måste finnas med

Dessa funktioner måste vara på plats för att systemet ska fungera bra:

### 1. Inloggning och behörigheter

- a. Säker inloggning (t.ex. tvåfaktorsautentisering)
- b. Användare ska ha olika roller (t.ex. admin, handläggare, kund) med olika rättigheter.

### 2. Ärendehantering

- a. Möjlighet att skapa, ändra och ta bort ärenden
- b. Ärenden ska kunna tilldelas en specifik handläggare
- c. Ärenden ska ha en status (öppet, pågående, avslutat)

### 3. Sök och filtrering

- a. Kunna söka efter ärenden baserat på t.ex. ID, status och datum
- b. Filtrera ärenden efter kategori och prioritet

### 4. Notifieringar och kommunikation

- a. Skicka automatiska meddelanden (e-post/SMS) när status ändras
- b. Användare ska kunna kommentera och lägga till filer i ärenden

### 5. Åtkomstkontroll och säkerhet

- a. Användare ska bara kunna se information som är relevant för deras roll

## Should Have (S) – Bör vara med

Dessa funktioner är viktiga, men systemet kan ändå fungera utan dem:

### 1. Rapportering och analys

- a. Kunna skapa rapporter om ärenden och handläggningstider
- b. Möjlighet att exportera rapporter till PDF eller Excel

### 2. Mobilanpassning

- a. Systemet ska fungera bra på både datorer, surfplattor och mobiltelefoner

## Could Have (C) – Skulle vara bra att ha

Dessa funktioner är inte nödvändiga, men kan vara bra att ha om tiden räcker till:

### 1. Loggning och spårbarhet

- a. Logga viktiga händelser, som inloggningar och ändringar av ärenden
- b. Loggar ska vara tillgängliga för administratörer i minst 12 månader

### 2. Anpassningsbara gränssnitt



- a. Möjlighet att ändra gränssnittet för en bättre användarupplevelse

## Won't Have (W) – Kommer inte vara med

Dessa funktioner ingår inte i den här versionen, men kan övervägas senare:

- Det finns inga funktioner här för nu, men i framtiden kan det finnas om behov uppstår.

## User Story

### 1. Inloggning och behörigheter

**Som** användare

**vill jag** kunna logga in med en säker metod (t.ex. tvåfaktorsautentisering)

**så att** jag kan skydda min information och få tillgång till mina ärenden.

#### Acceptanskriterier:

- ✓ Användare måste ange användarnamn och lösenord.
- ✓ Om tvåfaktorsautentisering är aktiverad, ska en kod skickas via e-post/SMS.
- ✓ Systemet ska visa felmeddelande vid felaktiga inloggningsuppgifter.

### 2. Skapa och hantera ärenden

**Som** handläggare

**vill jag** kunna skapa, redigera och radera ärenden

**så att** jag kan hantera arbetsuppgifter effektivt.

**Acceptanskriterier:**

- ✓ Handläggare kan fylla i ett formulär för att skapa ett nytt ärende.
- ✓ Endast handläggaren eller en administratör kan redigera eller ta bort ett ärende.
- ✓ Systemet visar en bekräftelse när ett ärende skapas eller ändras.

### 3. Tilldela ärenden

**Som** administratör

**vill jag** kunna tilldela ärenden till en specifik handläggare  
**så att** rätt person kan arbeta med ärendet.

**Acceptanskriterier:**

- ✓ Administratörer kan välja en handläggare från en lista.
- ✓ Tilldelad handläggare får en notifiering om att ett nytt ärende har tilldelats.
- ✓ Ärendets status ska uppdateras när det tilldelas.

### 4. Notifieringar och kommunikation

**Som** kund

**vill jag** få en notifiering via e-post eller SMS när mitt ärende uppdateras  
**så att** jag håller mig informerad om dess status.

**Acceptanskriterier:**

- ✓ Kunden får en notifiering vid statusändring (öppet → pågående → avslutat).
- ✓ Notifieringen ska innehålla information om vad som har ändrats.
- ✓ Kunden kan välja att stänga av notifieringar.

## 5. Sök och filtrering

**Som** handläggare

**vill jag** kunna söka och filtrera ärenden baserat på olika kriterier

**så att** jag snabbt hittar relevanta ärenden.

### Acceptanskriterier:

- ✓ Användare kan söka efter ärenden med ID, status, datum och handläggare.
- ✓ Systemet erbjuder filtrering efter kategori och prioritet.
- ✓ Resultaten visas snabbt och korrekt.

## Use Cases

### 1. Logga in på systemet

**Beskrivning:** Användaren loggar in på systemet för att få åtkomst till sina funktioner och ärenden.

**Aktörer:** Alla användare (kund, handläggare, administratör)

#### *Förutsättningar (Preconditions):*

- ✓ Användaren har ett registrerat konto i systemet.

#### *Huvudflöde:*

1. Användaren öppnar inloggningssidan.
2. Användaren skriver in sitt användarnamn och lösenord.
3. Systemet kontrollerar inloggningsuppgifterna.
4. Om tvåfaktorsautentisering är aktiverad, skickas en kod via SMS/e-post.
5. Användaren anger koden.
6. Systemet verifierar koden och loggar in användaren.

#### *Alternativflöden:*

- **(4A) Felaktigt lösenord:**
  - Systemet visar ett felmeddelande och låter användaren försöka igen.

- Om användaren anger fel lösenord flera gånger kan kontot låsas tillfälligt.
- **(6A) Tvåfaktorskod felaktig:**
  - Systemet visar ett felmeddelande och ber användaren försöka igen.
  - Efter flera misslyckade försök måste användaren begära en ny kod.

#### **Eftervillkor (Postconditions):**

- ✓ Användaren är inloggad och kan använda systemet.

## **2. Skapa ett nytt ärende**

**Beskrivning:** En handläggare skapar ett nytt ärende i systemet.

**Aktörer:** Handläggare, Administratör

#### **Förutsättningar (Preconditions):**

- ✓ Handläggaren är inloggad i systemet.

#### **Huvudflöde:**

1. Handläggaren går till sidan för att skapa ett nytt ärende.
2. Handläggaren fyller i ärendets information (titel, beskrivning, kategori, prioritet).
3. Handläggaren väljer en handläggare (om det inte är sig själv).
4. Handläggaren klickar på "Skapa ärende".
5. Systemet sparar ärendet och tilldelar en unik ID-kod.
6. Handläggaren och tilldelad handläggare (om någon valts) får en notifiering.

#### **Alternativflöden:**

- **(2A) Obligatorisk information saknas:**
  - Systemet visar ett felmeddelande och ber användaren fylla i alla obligatoriska fält.
- **(3A) Ingen handläggare väljs:**
  - Ärendet sparas, men förblir "otilldelat" tills en administratör väljer en handläggare.

#### **Eftervillkor (Postconditions):**

- ✓ Ärendet är skapat och kan hanteras vidare.

### **3. Uppdatera ärendestatus**

**Beskrivning:** Handläggaren ändrar status på ett ärende när det behandlas.

**Aktörer:** Handläggare, Administratör

#### **Förutsättningar (Preconditions):**

- ✓ Ärendet existerar i systemet.
- ✓ Handläggaren är inloggad och har behörighet att uppdatera status.

#### **Huvudflöde:**

1. Handläggaren går till ett ärende som hen hanterar.
2. Handläggaren ändrar status (t.ex. från "Öppet" till "Pågående").
3. Systemet sparar ändringen.
4. Kunden får en notifiering via e-post/SMS om statusändringen.

#### **Alternativflöden:**

- **(2A) Handläggaren försöker ändra status på ett ärende hen inte har behörighet till:**
  - Systemet visar ett felmeddelande och stoppar ändringen.

#### **Eftervillkor (Postconditions):**

- ✓ Ärendets status är uppdaterad och kunden har informerats.

## Projektplan och Struktur

### Tidsplan:

Jag ska dela upp mitt projekt i **4 sprintar** (2 veckor för varje sprint). Varje sprint kommer att fokusera på ett specifikt mål.

### Sprintplan

- **Sprint 1 (Vecka 1-2):**
  - Samla in resterande krav
  - Planering och databasschema
- **Sprint 2 (Vecka 3-4):**
  - Backend:
    - Databasanslutning
    - Autentisering
    - Implementera metoder
    - Testning
- **Sprint 3 (Vecka 5-6):**
  - Frontend:
    - Välja en lämplig UI-design
    - Be om hjälp från vår lärare
    - Testning
- **Sprint 4 (Vecka 7-8):**
  - Dokumentation
  - Buggfixar

## Arbetsprocess

Varje sprint består av:

- **Planering**
  - Identifiera uppgifter
  - Prioritera
- **Utveckling** (tar 2–10 dagar)
  - Varje dag arbetar jag **4 timmar i sträck**, sedan tar jag en **1 timmes promenad**, följt av **lunch**, och därefter ytterligare **2 timmars arbete**.
  - Om jag arbetar i ett team kommer vi att ha en **daglig stand-up** för att diskutera framsteg. Tiden för detta anpassas efter teamets behov.
- **Testning (Dag 11-12)**
  - Säkerställa att allt fungerar som förväntat
- **Feedback (Dag 13-14)**
  - Samla in och analysera feedback

## 3. Förändringshantering

Om nya krav uppstår hanteras de enligt följande process:

1. **Ny funktion föreslås** (t.ex. från kund eller teamet).
2. **Läggs i backlog** (Github project) för prioritering.
3. **Granskning & godkännande** – Beslut om ändringen görs direkt eller skjuts upp till en framtida sprint.
4. **Implementering** i kommande sprint om det godkänns.

**Exempel:** Om vi får en begäran att lägga till en ny rapporttyp så analyseras det innan sprintplaneringen.

