

Ärendehanteringssystem

Del 1

Vision Statement:

Vi vill skapa ett starkt och lättanvänt ärendehanteringssystem som hjälper organisationer att enkelt hantera, spåra och lösa ärenden på ett strukturerat sätt. Genom en central, säker och skalbar plattform vill vi förbättra samarbetet, förenkla arbetsflöden och säkerställa att regler och standarder följs.

Funktionella krav:

- 1- Must:** Användare måste logga in med en säker autentiseringsmetod.
- 2- Must :**Användare kan skapa, redigera och radera ärenden.
- 3- Must:** Ärenden kan tilldelas olika användare.
- 4- Should:** Statusuppdateringar och kommentarer kan läggas till ärenden.
- 5- Must:** E-post och SMS-notifieringar ska skickas vid statusförändringar.

Icke-funktionella krav:

- 1- Must:** Systemet ska stödja upp till 10 000 samtidiga användare.
- 2- Must:** Responstiden för att hämta ett ärende ska vara mindre än 3 sekunder.
- 3-Must:** Säkerhetskrav: Kryptering av lagrade och överförda data.

Vilka krav känns mest kritiska för att systemet över huvud taget ska fungera?

- **Användarinloggning med säker autentisering** – Detta är en grundläggande funktion för att säkerställa att endast auktoriserade användare kan komma åt systemet och hantera ärenden. Utan en säker inloggning skulle systemet vara sårbart för intrång och missbruk.
- **Skapa, redigera och radera ärenden** – Denna funktion är kärnan i systemet. Om användarna inte kan skapa eller hantera ärenden på ett korrekt sätt, är hela systemets syfte undergrävt.
- **Tilldelning av ärenden** – För att effektivt kunna hantera ärenden och hålla reda på vem som ansvarar för vad, måste användare kunna tilldela ärenden till rätt personer. Detta är avgörande för att hålla systemet strukturerat och organiserat.
- **E-post och SMS-notifieringar vid statusförändringar** – Notifikationer är viktiga för att hålla användare informerade om statusändringar. Utan notifieringar riskerar ärenden att bli åsidosatta eller försenade.
- **Kryptering av lagrad och överförd data** – För att uppfylla säkerhetskraven och skydda känslig information måste data krypteras. Detta är också ett krav för att följa lagstiftning om dataskydd.

Vilka krav skulle kunna vänta till en senare release?

- **Statusuppdateringar och kommentarer på ärenden** – Det är bra att kunna lägga till uppdateringar och kommentarer på ärenden, men systemet kan börja fungera utan den här funktionen i början. Den kan komma senare.
- **Skalbarhet (10 000 användare samtidigt)** – Det här är också viktigt, men om systemet inte ska användas av så många från början, kan det byggas ut när fler användare kommer. Det kan vara något som förbättras i framtiden.

Del 2

User Stories

1- Must: Användare måste logga in med en säker autentiseringsmetod.

- Som en användare vill jag logga in med en säker autentisering så att systemet inte blir sårbart för intrång och missbruk.

2- Must :Användare kan skapa, redigera och radera ärenden.

- Som en användare vill jag skapa, redigera och radera ärenden så att systemet kan fungera effektivt och att jag kan hantera mina ärenden på rätt sätt.

3- Must: Ärenden kan tilldelas olika användare.

- Som en administratör vill jag kunna tilldela ärenden till olika användare så att varje ärende hanteras av rätt person och vi kan hålla ordning på alla ärenden i systemet.

Use Case-översikt

2- Must :Användare kan skapa, redigera och radera ärenden.

- Som en användare vill jag skapa, redigera och radera ärenden så att systemet kan fungera effektivt och att jag kan hantera mina ärenden på rätt sätt.

Namn: Skapa, redigera och radera ärenden.

Actör: Användare, Handledare, Anställd.

Förutsättningar: Användare är inloggad

Huvudflöde:

- 1- Användaren loggar in på systemet med säker autentisering.
- 2- Användaren navigerar till "Ärenden" sektionen.
- 3- Användaren väljer att skapa ett nytt ärende, redigera ett befintligt ärende eller radera ett ärende.
- 4- Vid skapande av ärende fyller användaren i relevant information som titel, beskrivning, kategori och prioritet.
- 5- Vid redigering av ärende uppdaterar användaren relevant information som beskrivning, status eller tilldelning av ansvarig.
- 6- Vid radering bekräftar användaren att ärendet ska tas bort och ärendet försvinner från systemet.

Alternativa flöden:

Flöde A1: Skapa ärende misslyckas

- Om användaren inte fyller i obligatoriska fält vid skapande, får användaren ett felmeddelande och kan inte fortsätta förrän alla fält är korrekt ifyllda.

Flöde A2: Radering av ärende misslyckas

Om användaren inte har rätt behörighet att radera ett ärende, får de ett meddelande om att de inte har tillåtelse att ta bort ärendet.

Eftervillkor:

- Ett nytt ärende har skapats och lagrats korrekt i systemet.
- Ett ärende har redigerats och de ändrade uppgifterna är uppdaterade i systemet.
- Ett ärende har raderats och försvunnit från systemet.

- Om ett ärende raderas, kan det inte återställas (om inte systemet har en återställningsfunktion).

Tilldela ärenden till olika användare

- **Krav:** "Ärenden kan tilldelas olika användare."
- **Tasks:**
 - Lägga till en fält i databasen för att lagra ansvarig användare.
 - Skapa en funktion i backend för att ändra tilldelad användare.
 - Bygga en dropdown-menyn i frontend där användare kan välja ansvarig.

Skillnaden mellan User Story och Use Case

- **User Story** är en kort beskrivning av vad en användare vill göra och varför. Den fokuserar på användarens behov och mål.
 - Exempel: *"Som en användare vill jag kunna skapa ett ärende så att jag kan få hjälp med mitt problem."*
 - En user story är enkel, fokuserar på resultatet och hjälper teamet att förstå vad som behövs.
- **Use Case** är en mer detaljerad beskrivning av hur systemet fungerar i olika steg. Det visar exakt hur användaren interagerar med systemet.
 - Exempel: *Steg 1: Användaren klickar på 'Nytt ärende'. Steg 2: Användaren fyller i formuläret. Steg 3: Systemet sparar ärendet i databasen.*
 - Use case är mer tekniskt och hjälper utvecklare att förstå exakt hur funktionen ska byggas.

Varför är det viktigt att kunna växla mellan båda?

- **User Stories** ger en enkel översikt och gör det lätt att förstå vad som är viktigt för användarna.
- **Use Cases** ger detaljerad information om hur något faktiskt ska fungera i systemet.

Genom att använda båda kan man först få en översikt med en user story och sedan gå djupare med ett use case för att säkerställa att alla detaljer är tydliga. Detta hjälper både utvecklare, testare och projektledare att bygga rätt funktioner på ett effektivt sätt.