Testning av Secureme CRM

Förstudie

# Översikt

CRM-systemet har till syfte att kunna hantera kundärenden, som initieras genom att kunden fyller i och skickar ett formulär med sitt problem, som sedan kommer besvaras i en chatt. Chatten blir för kunden tillgänglig via ett mejl och för kundtjänstmedarbetaren via ett separat gränssnitt, i vilket kundtjänstmedarbetaren kan se samtliga ärenden samt sina egna. Administratören hanterar användare, så som kundtjänstmedarbetare.

## Implementerad funktionalitet utifrån projektets kravspecifikation

|  |
| --- |
| Implementerad funktionalitet utifrån kravspecifikationerna |
| Kunder kan skriva ett meddelande och ange sin mailadress. |
| Formuläret kan komma ur ett sammanhang där vi vet exakt vilket produkt/tjänst kunden kommenterar. |
| Efter att kunden bekräftat kan kunden ombes fylla i mer uppgifter om sig själv eller produkten/tjänsten. |
| Ett första svar från kundtjänst kan automatiseras. |
| Kunden ska inte behöva skapa ett lösenord för att kunna interagera med kundtjänsmedarbetare. |
| En admin måste finnas, som administrerar produkter/tjänster/medarbetare. |
| Kundtjänstmedarbetare behöver läggas till manuellt av en admin. |

# Funktionell analys

|  |  |
| --- | --- |
| Huvudfunktioner | Beskrivning |
| Kundmeddelande | En kund kan genom ett formulär fylla i ärendekategori, mejladress, förnamn, efternamn samt meddelande med rubrik |
| Kundchatt | En kundchatt finns tillgänglig via token för kunden i ett enskilt gränssnitt och för kundtjänstmedarbetaren i "Mina ärenden" genom att klicka på ett ärende. |
| Visa alla ärenden | En lista genereras med alla inskickade ärenden under sidan "Alla ärenden" |
| Autogenererat mejl vid nytt ärende | Ett autogenererat mejl skickas till kundens angivna mejladress när hen skickar in ett nytt ärende |
| Visa ärenden för kundtjänstmedarebetare | En lista genereras med en enskild kundtjänstmedarbetares tilldelade ärenden |
| Avsluta ärende | Kundtjänstmedarbetare kan avsluta ett ärende genom en knapp i chatten |
| Skapa ny användare | En ny användare kan skapas från menyn i medarbetargränssnittet |
| Visa alla användare | En lista genereras med alla användare i systemet under sidan "Alla användare" |
| Ta bort användare | Tar bort en användare (permanent/softdelete?) |

# Kvalitetsanalys

|  |  |
| --- | --- |
| Viktiga kvalitetsattribut | Beskrivning |
| Tillgänglighet | ?? |
| Användbarhet | Tydligt gränssnitt för kund och kundtjänstmedarbetare |
| Säkerhet | Endast kunden kan komma åt sin egen chatt |
| Stabilitet | Ingen förlust av data för ärende eller chatt |
| Tillförlitlighet | Felhantering och hantering av oväntad indata |
| Realtidskommunikation | Korrekt och uppdaterad data av meddelanden i chatten |
| Skalbarhet | Med så lite implementering som möjligt ska flera företag kunna hanteras |
| Underhållbarhet | Koden ska vara tydligt strukturerad och lätt att förstå utifrån namngivning |

# Testmål

|  |
| --- |
| **Säkerställa att:** |
| -          systemet hanterar felmeddelanden och felaktig input |
| -          att endast kunden kommer åt sin chatt |
| **Verifiera att:** |
| -          kunden får ett mejl med rätt unik chat token för sitt ärende |
| -          en unik chatt uppdateras med meddelande från både kund och kuntjänstmedarbetare |
| -          att ett ärendes status uppdateras från påbörjad till avslutad. |
| **Validera att:** |
| -          ärendedata sparas korrekt i databasen |
| -          meddelandedata sparas korrekt i databasen |
| -          att nya användare sparas korrekt i databasen |
| **Undersöka:** |
| -          om det finns kod som inte används |
| -          om användargränssnittet är tydligt |
| -          vad som krävs för att uppfylla skalbarhetskravet mot flera företag |
| -          i vilken utsträckning koden upplevs strukturerad samt förståelig utifrån namngivning |
| - om API för login i backend är uppbyggd för att kunna hantera vidare påbyggnad för login i frontend |
| **Bevisa att:** |
| -          API:er returnerar rätt statuskoder |

# Testomfattning

Testningen omfattar ärendeformulär, mejl-, och ärendehantering, nya användare samt chattfunktionalitet.

# Testnivåer

* Enhetstestning: Logik inne i API end points, eventuellt logik i frontend.
* API-testning: endpoints för ärendehantering, chatt och användare
* Gränssnittstestning: formulärflöde, ärendevy och chattkomponent

# Testverktyg

* xUnit för testning backendlogik och Jest för frontendlogik
* Postman för att testa API
* xUnit och SpecFlow tillsammans med Playwright för gränssnittstestning

# Roller och ansvar

Me

Me

Me

# Risker och antaganden

inga