

Slutrapport – Testning av Swine sync CRM-system för ärendehantering

Jacob Westholm

April 19, 2025

Contents

1	Sammanfattning	1
2	Testade komponenter	1
3	Genomförda testtyper	2
4	Testresultat	2
5	Upptäckta buggar	2
6	Täckning och begränsningar	2
7	Rekomndationer	2
8	Reflektioner och lärdomar	2

1 Sammanfattning

Denna rapport sammanfattar test resultaten

2 Testade komponenter

- Login och session hantering
- Skapandet av ett ärende i systemet.
- Skapandet av Företag, Admin, Produkt , Ärende Kategori, Kundmedarbetar i systemet.
- Blockering och av blockering av Företag, Admin, Produkt , Ärende Kategori, Kundmedarbetar i systemet.
- radering av Företag, Admin, Produkt , Ärende Kategori, Kundmedarbetar i systemet.
- Chat funktionalitet mellan Kund och Kundmedarbetare
- Rolebaserad autentisering och auktorisering.

3 Genomförda testtyper

- Api-testning: automatiserade testning genomförd med Postman.
- Gränssnittstestning genomförd med Playwright implementerat i c#.

4 Testresultat

Totalt genomfördes 73 testfall. Av dessa

- 71 passerad utan anmärkningar
- 1 resulterade i en minder bug
- 1 resulterade i en viktig bug

5 Upptäckta buggar

minder bug: punkt 1.

viktig bug: punkt 2.

1. Att försöka skapa en admin som en admin skickar tillbaka ett missvisande fel meddelande om att session inte existerar trots att felet är att man saknar behörighet.
2. Att försöka skapa ett Företag när man är inloggade som admin lyckas och returnerar code 200 trots att man saknar behörighet.

6 Täckning och begränsningar

- 87.5% av identifierad användnarflöden täcktes av de automatiska testerna.
- e-post och rating testades ej.

7 Rekomndationer

- Se över implementering av Rolebaserad auktorisering och autentisering och åtgärda säkerhets luckor.
- Se över fel meddelande implementera mer korrekt hantering av fel koder.
- Se över testbarhet för meddelande genom att skapa möjlighet att ta bort meddelande så att tester kan städas upp.

8 Reflektioner och lärdomar

Automatisk testning är ett kraftfullt verktyg för att Upptäcka brister i din kod. Det är lätt att skriva tester för att kontrollera att korrekta användning fungerar korrekt men det är betydligt svårare att kunna täcka in felaktig användning av koden och att kunna skriva tester som validerar att koden hanterar dessa på ett korrekt set.

Api prövning med postman fungerar bra även om postman upplevs något klumpigt.

UI prövning med Playwright fungerar bra men detaljer som tid för svar kan var viktigt när man kör flera tester samtidigt samt när man drar ner slowmo tiden.