Sprint 1

Software Engineering og testing

Piotr Husak

Innholdsfortegnelse

US2: A → B med ankomsttid (F6)

Brukerhistorie

Som reisende vil jeg søke reiser fra A til B med ønsket ankomsttid slik at jeg kan planlegge når jeg må dra for å komme frem i tide.

Akseptansekriterier og krav

Grunnleggende søkefunksjonalitet

Systemet skal akseptere input bestående av startpunkt, endepunkt og ønsket ankomsttid, og returnere minimum ett gyldig reisealternativ.

Sortering

Hvis flere alternativer eksisterer, skal systemet sortere resultatene etter tidligst ankomst.

Sekundære kriterier er færrest antall bytter og kortest total reisetid.

Sanntidsintegrasjon

Løsningen skal inkorporere sanntidsdata for forsinkelser og avvik i resultatet.

Feilhåndtering

Ved ugyldige søk skal systemet gi brukeren en tydelig feilmelding uten at applikasjonen krasjer.

Eksempler på hendelser som krever feilhåndtering:

- Ankomsttid i fortid
- Identisk start og sluttpunkt

Brukerfeedback

Ved manglende reisealternativer skal systemet presentere konstruktive forslag til søkejustering.

Teknisk arkitektur og implementering

Datamodell

Løsningen bygger på eksisterende systemarkitektur med følgende kjernekomponenter.

- SearchQuery: Kapsler søkeparametre inkludert start-/endepunkt og ankomsttid
- JourneyPlan: Representerer beregnede reisealternativer
- ItineraryLeg: Modellerer individuelle reisesegmenter med temporal informasjon
- RealTimeFeed: Integrerer sanntidsdata for dynamisk oppdatering
- Disruption: Håndterer forstyrrelser og avvik

Algoritmer

Implementeringen anvender baklengs søkealgoritmer som starter fra ønsket ankomsttid og beregner nødvendig avgangstid basert på ruteinformasjon og historiske reisetider. Sanntidsdata integreres for å justere beregninger dynamisk.

Kvalitetssikring og testing

Definition of Done

- Fullstendig akseptansekriterieoppfyllelse
- Minimum 80% kodedekning gjennom enhetstester
- Implementert og testet sanntidsintegrasjon med mock-data
- Brukergrensesnittvalidering gjennom tilgjengelighetstesting
- Implementert logging for ytelse- og stabilitetsevaluering
- Gjennomført demonstrasjon og dokumentasjon

Testplan

Enhetstester

Validering av algoritmer, input-validering og sorteringslogikk

Integrasjonstester

Verifikasjon av sanntidsdata-integrasjon

Scenariotester

End-to-end validering av brukerhistorier

Demonstrasjon og validering

Funksjonalitetsvalidering gjennomføres via kontrollert demonstrasjon hvor bruker spesifiserer reise fra A til B med ankomsttid xx:yy. Systemet skal presentere realisable alternativer, håndtere simulerte sanntidsforsinkelser (+3 minutter), og demonstrere kraftig feilhåndtering ved ugyldig input.

Konklusjon

Implementeringen av US2 representerer en kritisk utvidelse av systemets planleggingskapasitet gjennom introduksjonen av ankomsttidsbasert søkefunksjonalitet. Løsningen adresserer reelle brukerbehov og øker systemets konkurransedyktighet innen kollektivtransportplanlegging. Den tekniske kompleksiteten, primært relatert til baklengs planlegging og sanntidsintegrasjon, krever robust arkitektur og omfattende testing for å sikre pålitelighet og ytelse.