|  |  |
| --- | --- |
| Roameo   * Your personal travel planner | Produktrapport  Aarhus Tech  Jonas Wittstrøm  Svendeprøve – Datatekniker, programmering |

Indhold

[Indledning 2](#_Toc193203806)

[Casebeskrivelse 2](#_Toc193203807)

[Formål 2](#_Toc193203808)

[Funktionalitet 2](#_Toc193203809)

[Kravspecifikation 3](#_Toc193203810)

[Funktionelle krav 3](#_Toc193203811)

[Ikke-funktionelle krav 3](#_Toc193203812)

[Produktbeskrivelse 3](#_Toc193203813)

[Teknisk oversigt 3](#_Toc193203814)

[Frontend 3](#_Toc193203815)

[Backend 4](#_Toc193203816)

[Database og data persistens 4](#_Toc193203817)

[Diagrammer 4](#_Toc193203818)

[Flowchart (Bilag 1) 4](#_Toc193203819)

[Vejledning 4](#_Toc193203820)

[Test og validering 4](#_Toc193203821)

[Konklusion 4](#_Toc193203822)

[Kilder og referencer 4](#_Toc193203823)

[Bilag 5](#_Toc193203824)

[Bilag 1 – Flowchart 5](#_Toc193203825)

[Bilag 2 – ER Diagram over de mest centrale modeller 6](#_Toc193203826)

# Indledning

I dette afsnit vil jeg kort gennemgå følgende: En kort beskrivelse af min case, produktets formål og en overordnet beskrivelse af produktets funktionalitet.

## Casebeskrivelse

Formålet med denne app, er at hjælpe rejsende med at planlægge deres ture på en struktureret, overskuelig og visuel måde. Den kombinerer en dag-til-dag rejseplanlægger med integration af offentlig transport og brugerens personlige præferencer for mad og aktiviteter

## Formål

Formålet med mit produkt, er at gøre rejseplanlægning intuitiv, nemt og hurtigt ved at tilbyde en overskuelig dagsplan med planlagte aktiviteter, spisesteder og transportmuligheder. Appen tilbyder også anbefalinger baseret på brugerens præferencer.

## Overordnet Funktionalitet

Appen hjælper brugeren med at planlægge en personlig rejse ud fra egne præferencer. Brugeren vælger først destination samt datoer for rejsen. Derefter indtastes ønsker om mad, aktiviteter og generelle præferencer.

Ud fra disse valg foreslår appen relevante aktiviteter og spisesteder, som brugeren kan tilpasse efter behov. Når brugeren har godkendt den foreslåede rejseplan, genererer appen en overskuelig dag-til-dag oversigt med aktiviteter og måltider.

Under rejsen kan brugeren starte sin valgte dag direkte fra appen, vælge transportform, og få rutevejledning til aktiviteterne via integration med Google Maps.

# Kravspecifikation

I dette afsnit vil jeg gennemgå både de funktionelle og ikke-funktionelle krav til min app.

## Funktionelle krav

Her kommer en liste over appens funktionelle krav i punktform:

* Brugeren skal kunne vælge en ønsket destination for sin rejse.
* Brugeren skal kunne sætte start- og slutdato for sin rejse.
* Brugeren skal kunne sætte sine præferencer for spisning, oplevelser og aktivitetsniveau
* Appen skal kunne præsentere brugeren for forslag til spisesteder og oplevelser, baseret på brugerens valg.
* Brugeren skal kunne sætte et ønsket tidsrum for sine dagsplaner.
* Appen skal kunne præsentere brugeren for foreslåede dagsplaner baseret på brugerens valg.
* Brugeren skal have mulighed for at ændre sine planer inden endelig bekræftelse.
* Appen skal præsentere brugeren for valgte dagsplaner på en let og overskuelig måde.
* Appen skal gøre det nemt for brugeren at starte navigation via Google Maps fra sted til sted.

## Ikke-funktionelle krav

Ligesom ved de funktionelle krav, kommer her de ikke-funktionelle i punktform.

* Appen skal være nem og hurtig at starte.
* Brugerrejsen gennem appen skal være intuitiv.
* Appens brugergrænseflade (UI) skal være overskuelig.
* Appen skal leve op til GDPR-lovgivningen.
* Appen skal kunne køre på alle nyere Android og Apple enheder.
* Der skal være tydelig feedback til brugeren angående fejl i appen, ved manglende internetforbindelse og lign.

# Produktbeskrivelse

Dette afsnit deler jeg op i to underafsnit: En teknisk oversigt, der dækker over arkitektur, teknologier og platforme. Andet afsnit er diagrammer, som i dette tilfælde er en gennemgang af mit flowdiagram og et mindre ER-diagram for de mest centrale modeller i projektet.

## Teknisk oversigt

Dette afsnit dækker over mine valg af frontend, backend, databaser og datapersistens, integration af eksterne API ‘er og hvordan det hele i sidste ende hænger sammen.

### Frontend

Frontenden er lavet i .NET MAUI, som er en platform af Microsoft, der gør det muligt at udvikle apps til Android og Apple på samme tid, med en fælles samlet kodebase. Selve siderne er skrevet i .xaml format og den bagvedlæggende logik er skrevet i C#.

### Backend

Min backend er en ASP .NET Core Web API, som er deployed hos AWS (Amazon Web Services) som en AWS Lambda funktion. AWS Lambda er en serverløs tjeneste, som kan køre kode uden behov for serveradministration. Der vil blive fortalt mere om dette i procesrapporten.

Jeg bruger AWS API Gateway til at læse og hente data fra min database.

Backend kommunikerer med frontend gennem http kald.

### Database og data persistens

Til lokal lagring på brugerens enhed, bruger jeg SQLite, som er en letvægts, filbaseret database, som gemmer data lokalt. Dette gør den uden brug af servere.

Til styring af fjerndata bruger jeg AWS DynamoDB, som er en NoSQL database.

## Diagrammer

### Flowchart (bilag 1)

I mit endelige projekt nåede jeg ikke at få opsat brugerhåndtering (mere om dette i procesrapporten). Derfor endte mit diagram med at blive meget lineært. Det viser brugerens vej fra valg af destination til at man får sin rutevejledning i Google Maps. Jeg vil alligevel i punktform beskrive brugerens rejse gennem appen.

1. Brugeren starter på destinationsvælgersiden (DestinationPickerPage)
2. Her indtastes destination og start- og slutdato for rejsen
3. Brugerens input valideres:
   1. Hvis det ikke er gyldigt, skal der tastes igen
   2. Hvis det er gyldigt, fortsætter man
4. Navigerer til valg af præferencer (PreferencesPage)
5. Her vælger brugeren sine præferencer:
   1. Antal måltider om dagen
   2. Madpræferencer
   3. Oplevelsespræferencer
6. Præferencer gemmes
7. Appen genererer en midlertidig rejseplan
8. Navigerer til forslagssiden (SuggestionsPage)
9. Her vælger brugeren følgende:
   1. Tidsrum for sine aktiviteter
   2. Antal aktiviteter pr. dag
   3. Spisesteder
   4. Oplevelser
10. Disse valg gemmes og der genereres foreslåede rejseplaner.
11. Der navigeres til siden for foreslåede dagsplaner (SuggestedDayPlansPage)
12. Her kan brugeren vælge at bekræfte eller lave ændringer til sine planer.
    1. Ændr plan (Change Plan):
       1. Navigerer til EditPlansPage
       2. Brugeren foretager sine ændringer af tidligere valg.
       3. Navigerer tilbage til SuggestedDayPlansPage
    2. Bekræft plan (Confirm Plan):
       1. Fortsæt til siden for endelige dagsplaner (FinalizedDayPlansPage)
13. Brugeren gennemgår en oversigt over de genererede dagsplaner.
14. Vælger en bestemt dag og trykker på ”Start Day”
15. Navigerer til siden for de detaljerede dagsplaner (DetailedDayPlansPage)
16. Her vises en detaljeret oversigt over dagens planer.
17. Brugeren kan her vælge sin foretrukne transportmetode og trykke på ”Get Directions”
18. Appen tjekker tilladelse til lokation.
19. Brugerens aktuelle placering hentes.
20. Google Maps åbnes og brugeren kan nu tage hen til sin valgte aktivitet.

### ER-diagram (bilag 2)

Mit ER-diagram viser forholdet mellem de mest centrale datamodeller i min app. Siden jeg ikke fik brugerhåndtering til at virke, bliver det meste data håndteret af min Trip-entitet.

Trip repræsenterer selve rejsen og indeholder som destination, start-og slutdato (StartDate, EndDate). Mere komplekse detaljer som brugerpræferencer (UserPreferences) og Google Maps data, holder den som JSON-strenge.

UserPreferences har en 1:1 relation til Trip og beskriver brugerens præferencer som mad, måltider per dag, aktiviteter og energiniveau.

DayPlan (dagsplaner) har en 1-til-mange-relation med Trip og beskriver planlægningen for hver enkelt dag på turen. DayPlan indeholder et dagnummer, dato og en kort opsummering.

ScheduleItem (punkt på tidsplanen) er knyttet til DayPlan i en 1-til-mange relation, og holder detaljer om de enkelte aktiviteter som navn, tidspunkt, type af aktivitet (spisning eller oplevelse), og en lokationsforespørgsel til Google Maps.

# Vejledning

# Test og validering

- Testplan og testcases

- Resultater af test

- Dokumentation af fejlrettelser/optimeringer

# Konklusion

- Opsummering af produktets status og vurdering af løsningens kvalitet

- Forslag til videre udvikling

# Kilder og referencer

# 

# Bilag

## Bilag 1 – Flowchart

## Bilag 2 – ER Diagram over de mest centrale modeller

