

## 1. Johdanto

- **Projektin nimi:** COMP.SE.110 Äkkii lomalle halvalla.
- **Projektin tiimi:** Otto Haarahiltunen, Vilppu Tupeli, Eero Hakanen, Niilo Jaakkola
- **Yhteenveto:** Äkkilähtö skanneri hinnan, päivämäärien ja halutun sään perusteella.

### 1.1 Tarkoitus ja laajuus

Ohjelmiston tarkoituksena on ehdottaa halvimpia suoria lentoja eri kohdemaihin käyttäjän valitsemalta lentokentältä ja aikaväliltä. Löydetyt lennot järjestetään hinnan ja paikallisen sään mukaan. Näistä käyttäjä voi valita mieluisimman ja siirtyä verkkosivulle, josta lennon voi varata itselleen.

### 1.2 Tavoitteet

- Lyhyen aikavälin tavoitteina on saada ohjelma tarjoamaan kohtuullisia vaihtoehtoja
- 

## 2. Vaatimusmäärittely

### 2.1 Toiminnalliset vaatimukset

- **Hakuehtojen valitseminen:** Käyttäjän tulee pystyä valitsemaan reissun budjetti, lähtöpäivä, paluupäivä ja kenttä.
- **Tietojen visualisointi:** Käyttäjän tulee voida nähdä karttapohjaisia visualisointeja matkakohteista, säistä ja hinnoista.

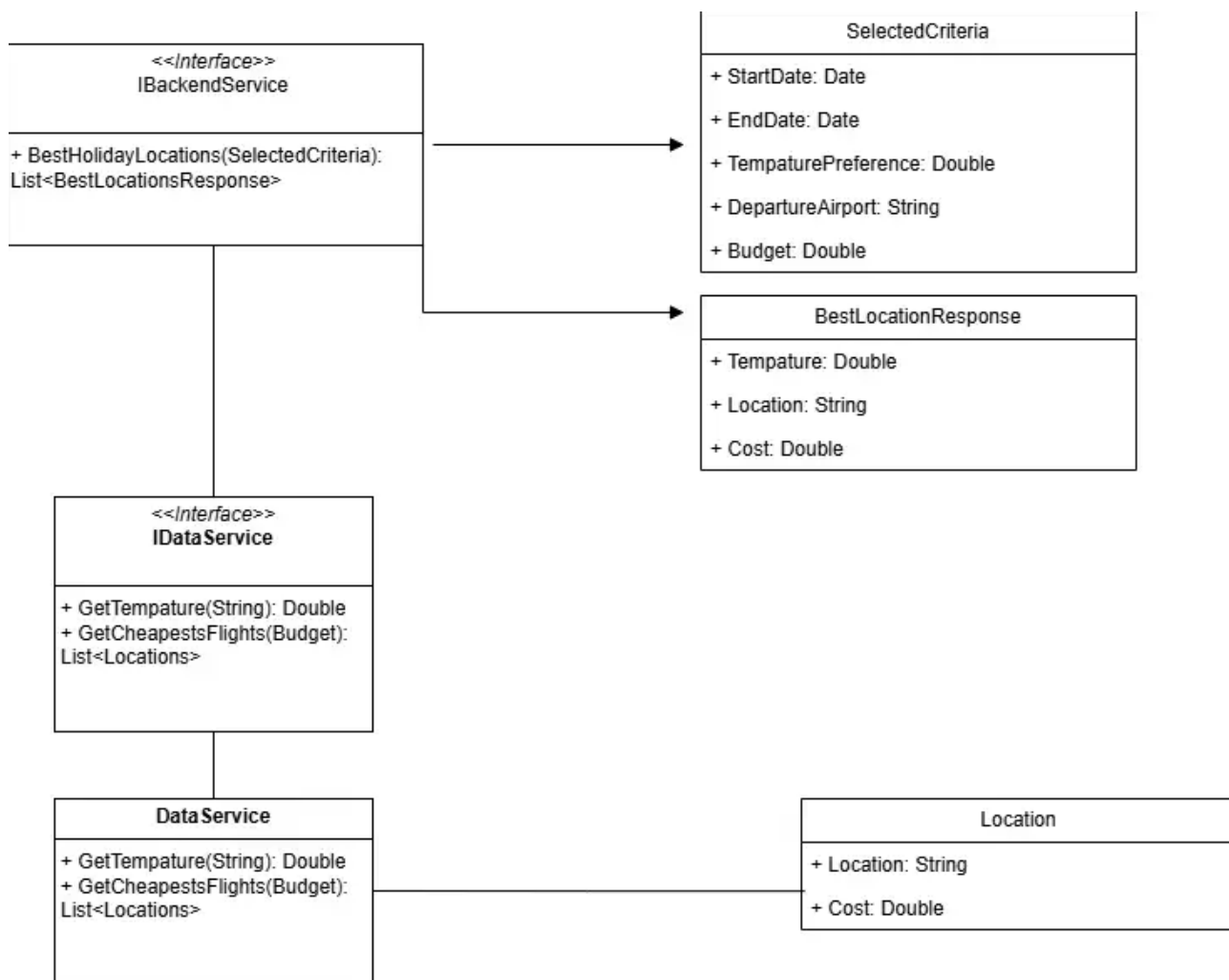
### 2.2 Ei-toiminnalliset vaatimukset

- **Suorituskyky:** Sovelluksen tulee toimia riittävällä nopeudella.
- 

## 3. Arkkitehtuurisuunnitelma

### 3.1 Yleiskuva arkkitehtuurista

Sovellus toteutetaan web-sovelluksena MVC periaatetta noudattaen, jossa Front-End vastaa vain datan visualisoinnista ja käyttöliittymän toiminnallisuuksista. Sovelluksen liiketoimintalogiikka tapahtuu Back-Endissä, jossa hyödynnetään kerrosarkkitehtuuria selkeyttämään rakennetta ja helpottamaan työn jakamista sekä testausta. Alla oleva luokkakaavio havainnollistaa Back-Ending rakenteen ja eri kerrosten väliset suhteet.



Kuva 1.1: Luokkakaavio Back-endistä

<https://drive.google.com/file/d/1OiQVKGx7B5HnSlQVaulCjwz9y3TbM1ZT/view?usp=sharing>

### 3.2 Teknologiat ja työkalut

- **Frontend:** Vanilla JS
- **Backend:** Java

## 5. Käyttöliittymän suunnittelu

### 5.1 Käyttötapaukset

Yksityiskohtaiset käyttötapaukset, joissa selitetään käyttäjän toiminnot sovelluksessa, esim.:

- **Karttatietojen tarkastelu**
- **Datan lataaminen ja käsittely**

### 5.2 Käyttöliittymän prototyyppi

[Käyttöliittymän prototyyppi](#)

## 6. Integraatio ulkoisten palveluiden kanssa

## 6.1 API-kuvaus

Miten sovellus integroituu ulkoisiin palveluihin (esim. OpenStreetMap API tai säädata-API). Sisällytä esimerkkejä API-kutsuista ja niiden vasteista.

Käytettävät ulkoiset rajapinnat

- [Weather API - OpenWeatherMap](#)
- [Flight Offers Search API - Find Cheap Flights, 400+ Airlines | Amadeus for Developers](#)
- <https://www.flightsfrom.com/>

## 6.2 Tietojen yhdistäminen ja visualisointi

Sovelluksen tarkoituksena on yhdistää edellä mainituista API-kutsuista saatava tieto ja näyttää se alla olevan käyttöliittymä suunnitelman perusteella.

---

## 7. Testaus

### 7.1 Testausstrategia

Testausstrategia Front-Endille on loppuvaiheessa suoritettava manuaalinen hyväksymistestaus, jota täydennetään tarvittaessa yksikkötesteillä. Testauksen pääpaino on Back-Endissä, jossa jokaiseen kerrokseen toteutetaan yksikkötestit sekä integraatiotestit kerrosten välisten rajapintojen toiminnan varmistamiseksi.

---

## 8. Projektinhallinta ja aikataulu

### 8.1 Projektinhallintatyökalut

Projektin etenemisen seurantaan ja tehtävienjakoon käytetään Trelloa. Ryhmän sisäiseen kommunikointiin käytetään Telegramia ja Discordia. Koodin versionhallintaan käytetään Gitiä ja GitHubia. Käyttöliittymän suunnitteluun ja prototyypin luomiseen käytetään Figma-työkalua.

## 9. Tekoälyn käyttö projektin suunnittelussa ja toteuttamisessa

Tekoälyä käytettiin ideoinnissa ja mahdollisten apien etsimisessä. Tekoälyllä myös luotiin dokumentaation pohja.