OTP2 – loppuraportti

Ryhmä 4: MatkApp / Mika Iso-Ikala, Teemu Lindgren, Heikki Sillanpää & Eero Tuure

Lyhyt kuvaus mistä sovelluksessa on kyse (fokus on jo voinut muuttua siitä, mitä OTP1:ssa selititte)

> *MatkApp on budjetointia ja kulujen seurausta varten toteutettu ohjelmisto, jolla voidaan jakaa kuluja käyttäjien kesken. Ohjelmisto on suunniteltu käytettäväksi ryhmissä, joissa kulut halutaan jakaa osallistujien kesken tasan. Alkuperäinen visio tuotteelle oli retkeilyn suunnitteluohjelmisto, mutta fokus siirtyi OTP2:n myötä kulujen/budjetin käsittelyyn.*

Mitä toiminnallisuuksia on tähän mennessä toteutettu (tässä voi viitata Agilefantissa oleviin käyttäjätarinoihin)

> *Käyttäjätilit ja rekisteröinti, käyttäjätietojen muokkaus, ryhmien luonti ja niihin liittyminen sekä kutsuminen (sähköposti-ilmoitukset), kulujen kirjaus ja jakaminen ryhmäläisten kesken, budjetin määrittely sekä lokalisaatio kolmelle kielelle.*

Kuvaus tuotteenne arkkitehtuurista (näkökulma sellainen, että jos joku uusi kehittäjä tulisi mukaan kehitystyöhön, niin tämä opastaisi häntä, miten pääsisi alkuun kehitystyössä halutessaan tehdä uusia toiminnallisuuksia tuotteeseen)

> *Ohjelma on toteutettu MVC-mallia noudattaen ja hyödyntäen JavaFX:n FXML-formaattia.*
>
> *Näkymän rakenne on suunniteltu hyödyntäen Composite-mallia. On olemassa juuri näkymä, jonka sisälle ladataan FXML-tiedostosta näkymiä ja niihin liitetään vastaava kontrolleri. Näkymien kontrollerit perivät abstraktin luokan, jolla on tieto sen omistaja kontrollerista ja se sisältää kontrollerille yhtenäisiä toiminnallisuuksia. Kontrollerit keskustelevat toisilleen rajapintaa hyödyntäen.*
>
> *Data access objektit on sijoitettu omaan tasoon. Access objektien käyttöä ohjaa tietokannan sessiota ylläpitävä luokka.*

Selitys siitä, miten tuotettanne on testattu (manuaalinen testaus, automaattitestaus Jenkins; yhteenvetotaulukko testitapauksista esimerkiksi)

Manuaalinen testaus:

Pyysimme täysin ulkopuolista henkilöä testaamaan sovelluksen käytettävyyttä ja saimme paljon vinkkejä, mitä voi tehdä paremmin ja mitkä asiat ovat turhia jne. Ulkopuolisen silmin tulee uusia ideoita sovelluksen toimintaan ja ulkoasuun liittyen. Sovellusta testasimme jatkuvasti myös itse ja yritimme löytää bugeja ja varmistaa, että kaikki toiminnot toimivat halutulla tavalla.

Automaattinen testaus:

Testejä on toteutettu data access objekteille ja tärkeimmille malli -tason luokille.

Automaattiset käyttöliittymän testit ovat jääneet vähemmälle, mutta niitäkin muutamille luokille löytyy. Jenkins palvelimella oli vaikeuksia suorittaa testFX testejä - joko testit jäivät pyörimään tai testFX tuki puuttuu. Tästä syystä priorisointi käyttöliittymän testeille jäi vähäiselle.