Ryhmän jäsenet:

Tuomas Vehkala - 000534725 - tuomas.vehkala@student.lut.fi

Niilo Tallberg - 000466233 - niilo.tallberg@student.lut.fi

Aihe: ruuantilaus- / ravintolasovellus

Lähdekoodi- ja dokumenttirepository: https://github.com/niilotallberg/FoodOrderingApp

Esittelyvideo: https://youtu.be/DjFnODKTzOk

Yleinen kuvaus työstä

Harjoitustyön aiheena oli luoda ravintolasovellus, jonka kautta käyttäjä pystyy tilaamaan valitseman ruuan noudettuna tai toimitettuna. Sovellus sisältää monia samantyyppisiä ominaisuuksia, joita näkee oikeaan tarkoitukseen luoduissa sovelluksissa (esim. Wolt, Foodora). Käyttäjän on mahdollista luoda oma profiilinsa sovellukseen syöttämällä rekisteröimisnäkymään sähköpostin, käyttäjätunnuksen ja salasanan. Kirjautumalla sisään, käyttäjä pääsee tarkastelemaan sovelluksen kotisivuja. Sovelluksessa käyttäjän on mahdollista lisätä koriinsa monenlaisia eri tuotteita. Ostoskorin kautta sovelluksen käyttäjä voi vielä muuttaa tilaustaan. Tilauksen vahvistusnäkymässä valitaan toimitus- ja maksutapa. Nämä täytettyään käyttäjä pääsee vahvistamaan tilauksensa. Mikäli käyttäjä ilmoitti noutavansa ruuan, hänelle esitetään arvioitu aika siitä, milloin ruoka on noudettavissa. Toisaalta, mikäli käyttäjä valitsi kotiinkuljetuksen, hänelle esitetään arvio siitä, milloin ruoka on ovella. Tilauksen seurantanäkymässä käyttäjän on mahdollista arvostella sovelluksen käyttökokemus tähdillä. Apissa on mahdollista myös lisätä käyttäjän henkilökohtaiseen suosikkinäkymään eri ruokia. Tämä mahdollistaa sen, että käyttäjä voi kätevästi sisään kirjauduttuaan navigoida suoraan tilamaan lempituotteitaan. Sovellus sisältää myös käyttäjälle mahdollisuuden tarkistella usein kysyttyjä kysymyksiä FAQ-näkymän kautta, johon inspiraatiota olemme ottaneet samantyyppisistä sovelluksista. Lisäksi käyttäjä voi muokata oman profiilinsa tietoja erillisestä näkymästä. Käyttäjä voi vaihtaa esimerkiksi oman profiilikuvansa tai salasanansa.

Työnjako

Suunnitteluvaiheessa teimme hyvin tiivistä yhteistyötä. Luokkakaavioiden suhteen teimme erikseen Activity-luokkakaavion sekä normaalin luokkakaavion. Niilo vastasi normaalin luokkakaavion tekemisestä ja Tuomas Aktiviteetti-luokkakaavion.

Koodausvaiheessa Niilo rakensi sovelluksen kirjautumis- / rekisteröitymistoiminnallisuuden, ostoskorin toiminnallisuuden sekä favorites ja profile-settings-näkymät ja toiminnallisuuden. Koodausvaiheessa Tuomas taas toteutti tilauksen vahvistamiseen ja seurantaan liittyvät näkymät sekä toiminnallisuuden. Lisäksi hän rakensi sovelluksen FAQ-osion.

Sovelluksen ominaisuudet / toiminnallisuudet

1. Sovellukseen tulee kirjautua sisään (Mikäli ei ole vielä käyttäjää, tarjotaan mahdollisuus luoda sellainen ja kirjautuminen onnistuu vasta sen jälkeen)
2. Sovelluksessa voi selailla tilattavia ruokia etusivun tai omien suosikkien kautta
3. Ruokien tarkempia tietoja voi tarkastella halutessaan tarkemmin painamalla ruokien kuvakkeita (Tätä kautta pääsee myös lisäämään tuotteita omaan ostoskoriin tai omiin suosikkeihinsa)
4. Ruokia pystyy lisäämään eri määriä ostoskoriin ja määriä pystyy muuttamaan vielä oman ostoskorin puolella
5. Ostoskorissa ruokien hinnat + esimerkiksi toimituskustannukset ja verot näkyvät erikseen ja kootusti
6. Ostoskorissa ruokien määrää voi vielä muuttaa tai ne voi poistaa kokonaan ja nämä muutokset muuttavat myös hintoja reaaliaikaisesti
7. Oman profiilin asetuksia voi muuttaa (profiilikuvaa, sähköpostiosoitetta, osoitetta ja salasanaa) profile-settings-näkymässä
8. ”Tuki”-osiossa voi selata kätevästi usein kysyttyjä kysymyksiä, josta käyttäjä voi saada apua pieniin ongelmiin.
9. Ostoskorissa olleen tilauksen voi hyväksyä, jolloin sovellus pyytää antamaan maksutiedot ja valitsemaan toimitustavan kotiinkuljetuksen ja noudon väliltä. Mikäli osoite on tallennettuna profiiliin ja käyttäjä valitsee kotiinkuljetuksen, sovellus ehdottaa toimitusta tähän osoitteeseen. Jos osoitetta ei ole vielä syötetty profiilille tai sitä halutaan muuttaa, niin osoitteen voi syöttää / sitä voi muuttaa tässä vaiheessa.
10. Kun käyttäjä on valinnut toimitus- ja maksutavan ja vahvistanut tilauksen - > Sovellus avaa näkymän, jossa näkyy, milloin tilaus on noudettavissa / ovella.
11. Jokaisen käyttäjän viimeisen ostoskori ja suosikit ovat reaaliaikaisesti säilössä erilisissä tiedostoissa, jonka ansioista ne säilyvät, vaikka käyttäjä kirjautuisi esimerkiksi ulos tai joku muu kirjautuisi välissä sisään.

Pisteytys (anomus)

Pakolliset = 15p

Ohjelma on koodattu olioparadigman mukaisesti - Kyllä

Ohjelman kaikki koodi ja kommentit ovat englanniksi. - Kyllä

Ohjelman tulee toimia erilaisilla Android-puhelimilla - Kyllä

Lisäpisteet:

Listatut

- **1 p** - Satunnaisuus (Olemme käyttäneet sovelluksessa hyödyksi Math.random-metodia. Metodia käytetään toimitusajan määrittämisessä. Sovellus arpoo tällä hetkellä sovelluksen käyttäjälle näytettävän toimitusajan 10-30min väliltä ihan vain osoittaaksemme, että hallitsemme metodin käytön)

- **4 p** - Ohjelmassa käytetään RecycleView-komponenttia listattaessa ruokia ja niiden tietoja (Pyydämme tästä ominaisuudesta 1 p enemmän kuin tehtävänannossa oli listattuna, sillä olemme käyttäneet recyclerView-komponenttia hyvin monessa eri asiayhteydessä ja melko soveltavasti)

- **3 p** - Erilaiset ruuat on visualisoitu erilaisin kuvin (Pyydämme tästä ominaisuudesta 1p enemmän kuin tehtävänannossa oli listattuna, sillä olemme käyttäneet erilaisia kuvia hyvin runsaasti ympäri sovellusta, jotta sen visuaalinen ilme olisi poikkeuksellisen siisti)

- **3 p** - Tietojen tallentaminen tiedostoon + sieltä lukeminen (Käyttäjätietojen, käyttäjäkohtaisten ostoskorien ja käyttäjän suosikkien tallentaminen erillisiin tiedostoihin puhelimen muistissa - > mahdollistaa kirjautumis- / rekisteröitymis-toiminnallisuuden, käyttäjäkohtaisten ostoskorien säilymisen ja käyttäjäkohtaisten suosikkien säilymisen kirjautumisten välissä. Pyydämme ominaisuudesta 1 p enemmän kuin tehtävänannossa oli listattuna, sillä tiedostoja ja serialisointia hyödynnetään todella laaja-alaisesti tässä toteutuksessa)

Omat

- **2 p** - Kirjautumis- / rekisteröitymistoiminnallisuus (Sovellus sisältää hyvin kattavan rekisteröitymis- ja kirjautumistoiminnallisuuden. Rekisteröityneet käyttäjät serialisoidaan erilliseen tiedostoon puhelimen muistissa. Kun käyttäjä yrittää kirjautua sisään, ohjelma tarkistaa, löytyykö tiedostosta sähköpostia ja salasanaa vastaava käyttäjä. Jos ei löydy, niin ohjelma ilmoittaa, että kyseisellä sähköposti-salasana-parilla ei löytynyt käyttäjää ja pyytää rekisteröitymään. Toiminnallisuus sisältää runsaasti erilaisia tarkistuksia. Kun ohjelman käyttäjä esimerkiksi yrittää rekistöityä, ohjelma tarkistaa, että jokaiseen kenttään on syötetty pyydetyt tiedot, sähköposti tai käyttäjänimi ei ole kenenkään muun käytössä, sähköposti noudattaa virallisen sähköpostiosoitteen vaatimuksia, salasana on vähintään 8 merkkiä pitkä ja sisältää vähintään yhden merkin ja yhden numeron. Lisäksi salasanan ja salasanan vahvistuksen tulee olla samat tai muuten rekisteröityminen ei onnistu)

- **1 p** - Toast -ilmoitusten käyttäminen keinona informoida käyttäjää sovelluksen ajon aikana (Esimerkiksi väärin syötetyistä tiedoista informoidaan sovelluksen käyttäjää Toast-ilmoitusten avulla)

- **1 p** - Scroll View -näkymien käyttö laaja-alaisesti ympäri sovellusta (Todella moni sovelluksen näkymistä käyttää ScrollView-komponenttia, jonka ansiosta näytön koko ei rajoita komponenttien määrän näkymän sisällä. Esimerkiksi ruokia voidaan listata kategoriakohtaisesti allekkain ScrollView:n ansioista monta allekkain)

- **1 p** - Käyttäjää estetään tietyissä kohdin ohjelmaa painamasta puhelimen taaksepäin nappia (Kun käyttäjä on esimerkiksi maksanut tilauksensa ja siirtyy WaitingScreenActivityyn, hän ei voi enää palata tuohon edelliseen activityyn eli peruuttaa tilausta. Tämä on estetty käyttämällä erillistä onBackPressed-metodia, joka ilmoittaa käyttäjälle ”Toast”-ilmoituksen avulla ”You can’t go back…”. Lisäksi ohjelman backstack poistetaan, kun esimerkiksi tämä WaitinScreenActivity aloitetaan. Tämä tehdään näin:
”intent.addFlags(Intent.*FLAG_ACTIVITY_CLEAR_TOP* | Intent.*FLAG_ACTIVITY_CLEAR_TASK* | Intent.*FLAG_ACTIVITY_NEW_TASK*);”

- **2 p** - Käyttäjä voi muuttaa omia tietojaan ProfileSettingsActivityn kautta, painamalla esimerkiksi etusivulla olevan CoordinatorLayoutin profiilikuvaa (Käyttäjä voi vaihtaa profiilikuvaansa, sähköpostiaan, salasanaansa ja osoitettaan. Sähköpostin suhteen tarkistetaan, ettei se ole kenenkään muun käyttäjän käytössä. Mikäli käyttäjä muuttaa tietojaan, tiedot päivitetään ympäri sovellusta sis. esimerkiksi profiilikuvan sovelluksen etusivulla. Lisäksi tiedot päivitetään tiedostoon, johon käyttäjätiedot on tallennettu. Tämä mahdollistaa sen, että mikäli käyttäjä muuttaa esimerkiksi salasanaansa, hän voi kirjautua uudella salasanalla ensi kerran, kun hän kirjautuu sisään.)

- **2 p** - CoordinatorLayoutin käyttö mahdollistamassa laajalti ympäri sovellusta käytetyn nappivalikon Acitivtyjen alalaidassa (Käyttäjän käytettävissä on ympäri sovellusta valikko nappeja, joiden avulla tämä pääsee kotisivulle, omiin suosikkeihinsa, profiilin asetuksiin, tuki-osioon tai tarkastelemaan omaa ostoskoriaan)

- **2 p** - Ohjelmassa on käytetty todella laajasti ja monipuolisesti erinäisiä Adaptereita (esimerkiksi CategoryAdapter ja DetailCategoryAdapter muodostavat yhdessä melko monimutkaisen kokonaisuuden joka mahdollistaa sen, että käyttäjä voi selailla etusivulla eri kategorioita, joita sovelluksessa on. Lisäksi, mikäli käyttäjä painaa jotain tiettyä kategoriaa niin aukeaa recyclerView, joka sisältää vain tuon kyseisen kategorian ruokia.)

- **1 p** - Ohjelmasta on tehty visuaalisesti näyttävä. Esimerkiksi nappien reunoja on pyöristetty tekemällä oma drawable-tiedosto napin taustaksi

- **2 p** - Ohjelmaan on rakennettu favorite-foods toiminnallisuus. Toiminnallisuuden ansioista, kun käyttäjä tarkastelee jotain ruokaa sovelluksen sisällä, hän voi painaa mustaa tähteä, jolloin tähti muuttuu keltaiseksi ja se lisättiin käyttäjän suosikkeihin. Suosikkeja käyttäjä voi tarkastella omana sivunaan, kun tämä painaa tähden kuvaa sovelluksessa esiintyvästä alavalikosta. Omia suosikkeja voi poistaa mistä tahansa päin sovellusta, kun tarkastelee ruokien niin sanottua detail-näkymää. Mikäli jokin ruoka on käyttäjän suosikki, tähti näkyy joka puolella sovellusta keltaisena. Viimeisin tieto käyttäjän suosikeista on aina säilöttynä erilliseen tiedostoon puhelimen muistissa, joten käyttäjän suosikit säilyvät, vaikka tämä kirjautuisi ulos tai joku muu käyttäjä kirjautuisi välissä sisään.

- **3 p** - Ostoskorin toiminnallisuus. Käyttäjä voi lisätä haluamansa määrän tuotteita koriin ja voi myös muuttaa tuotteiden määrää vielä ostoskorin puolella. Mikäli käyttäjä yrittää lisätä moneen otteeseen saman tuotteen, tuotteen määrä vain lisääntyy korissa, eikä sinne ilmesty uutta samaa tuotetta erillisenä kenttänä. Ostoskori on toiminnallisuudeltaan kehittynyt. Ostoskorin sisällä, käyttäjä voi muuttaa lisäämiensä tuotteiden määrää painamalla jokaisen ostoskorissa olevan tuotteen kohdalla ”+” tai ”-” nappia. Ostoskorissa on listattuna recyclerView:n avulla jokainen sinne lisätty ruoka, oikeassa määrässä. Mikäli käyttäjä haluaa poistaa tuotteen kokonaan korista, voi tämä painaa ”-” niin monta kertaa, että se menee nollaan. Nollaan mennessään, tuote poistuu kokonaan ostoskorista ja recyclerView:stä. Ostoskori listaa myös jokaisen tuotteen kappalehinnan, määrästä muodostuvan hinnan, koko ostoskorin tuotteiden hinnan, verojen määrän, toimituskulut ja kokonaishinnan. Nämä hinnat myös muuttuvat reaaliajasssa, mikäli käyttäjä muuttaa ostoskorissa oleviensa tuotteiden määriä tai poistaa niitä.

- **1 p** - Sovellus sisältää FAQ-osion, jossa voi tarkastella usein kysyttäjä kysymyksiä ja vastauksia niihin. Mikäli käyttäjä painaa jotakin kysymystä, aukeaa kysymyskohtainen vastaus sen alle.

- **1 p** - Maksu- ja toimitustavan yhteydessä on melko laajat vaatimukset, jotka tulee täyttää, jotta tilauksen voi vahvistaa. Käyttäjä ei pääse vahvistamaan tilaustaan, tämä ei ole syöttänyt maksutapaansa onnistuneesti. Käyttäjän tulee joko syöttää uuden kortin tiedot (joille on myös omat vaatimukset ja tarkistukset) tai valita vaihtoehtoisesti maksutavaksi ”Mobilepay” tai ”PayPal”, jotta sovellus päästää eteenpäin vahvistamaan tilauksen. Käyttäjän tulee myös valita toimitustapa (nouto vai kotiinkuljetus) ja mikäli käyttäjä valitsee kotiinkuljetuksen niin ruudulle tulee näkyviin EditText kenttä, jossa on toimitusosoite. Osoite on oletusarvona osoite, jonka käyttäjä on asettanut mahdollisesti profiilin asetuksissa. Mikäli käyttäjä ei ole asettanut osoitetta aikaisemmin, tulee osoite täyttää tässä kohtaa. Mikäli käyttäjä jättää osoitteen tyhjäksi, tilauksen vahvistaminen ei onnistu. Osoitetta ei kuitenkaan kysytä, mikäli käyttäjä valitsee noutavansa itse tilauksen. Toiminnallisuudessa on käytetty eri näkyvyysmääreitä (.VISIBLE vs .GONE)

- **1 p** - ”Interface”:n eli rajapinnan hyödyntäminen osana ohjelman toteutusta. Koodissamme on käytössä tällainen abstrakti luokka, joka implementoidaan CartListAdapter-luokassa. Rajapintaa käytetään ostoskorissa olevien tuotteiden määrän muuttamisen yhteydessä.

- **1 p** - ”Enum” hyödyntäminen osana ohjelman toteutusta. Ohjelmassa lisätään jokaiselle käytettävälle foodDomain oliolle kategoria enum avulla. Tämä mahdollistaa erilaisten sovelluksen sisältämien ruokien sorttaamisen kategorian mukaan.

  yhteensä 15 p (vaatimukset) + 32 p lisäominaisuudet = **47 p**

Lopullista toteutusta kuvaava luokkakaavio