Olio-ohjelmointi harkkatyö

- 1. Itse tein kaiken
- 2. Ohjelman suunnitteluun käytin harkkatyön kuvauksessa olevia kuvia ja tein sen niiden pohjalta. Aloitin projektin sijoittamalla scene builderiin palikoita kohdilleen ja tämän jälkeen kirjottelin niille koodit. Aina kun tuli uusia toimintoja mieleen niin lisäilin ne ekaksi scene builderiin ja tämän jälkeen kirjoitin koodia.
- 3. Toiminnallisuudet:
 - -Etuivulle ladataan kartta
 - -Vasemmalla olevasta comboboxista voi valita kaupungin (kaupungit haetaan excelistä) ja painamalla nappia "lisää kartalle" niin valitun kaupungin smartpost automaatit lisätään kartalle.
 -"Luo Paketti" –napista pääsee uuteen ikkunaan jossa voi luoda uuden esineen tai valita listasta itse määrittelemäni esineen. kun tämä on tehty niin esineelle valitaan pakettiluokka (tiettyjä paketteja ei voi laittaa joihinkin pakettiluokkiin) esim esine pesukone ei mene pakettiluokkaan kaksi. "Infoa
 - ei voi laittaa joihinkin pakettiluokkiin) esim esine pesukone ei mene pakettiluokkaan kaksi. "Infoa luokista" pääsee uuteen ikkunaan, jossa on jotain infoa näistä kolmesta eri luokasta. Jos "Särkyvää" checkboxin aktivoi ja lähettää tuotteen pakettiluokassa yksi niin posteljooni heittelee pakettia niin paljon, että paketti särkyy. Ykkösluokan pakettia ei voi lähettää yli 150km päähän.
 - -Pakettien lähetystietoihin eka valitaan kaupunki(haetaan excelistä) ja tämän jälkeen alapuolella oleva combobox fetchaa excelistä valitulle kaupungille smartpost automaatit. Siitä valitaan siis lähtö ja määränpää ja tämän jälkeen painamalla "Luo paketti" paketti luodaan.
 - -Tämän jälkeen menemällä takaisin etusivulle ja painamalla "Päivitä pakettilista" saa äsken tehdyt paketti näkyviin "Valitse lähetettävä paketti" comboboxiin. Siitä valitaan sitten haluttu paketti ja painetaan "Lähetä paketti" jolloin paketti lähtee liikkeelle ja karttaan tulee näkyviin reitti, jota pitkin paketti menee.
 - -"Poista reittit kartalta" -napista voi poistaa piirretyt reitit
 - -Etusivun vasempaan alakulmaan tulee tieto matkan pituudesta kun paketti lähetetään ja alakulmaan tulee virheilmoituksia esim. jos 1. luokan pakettia yritetään lähettää yli 150km päähän. -Reitin väri määrittyy, millä pakettiluokalla se on lähetetty: 1. luokka on sininen, 2. luokka on punainen ja 3. luokka on vihreä.
 - -jos haluaa tehdä uuden esineen niin riippuu esineen koosta, mihin luokkaan sen voi laittaa. Pakettiluokka 2 esine täytyy olla pienempi kuin 30*30*30cm sekä painaa alle 15kg ja pakettiluokan 3 esine täytyy olla suurempi kuin 30*30*30cm sekä painaa yli 35kg.
 - -aloitussivulla on myös raha-automaatti, johon täytyy lisätä rahaa jos haluaa, että paketti lähtee liikkeelle. Paketin lähetys maksaa matka jaettuna 10 eli 100km matka maksaa 10euroa. Sliderista valitaan rahasumma ja alapuolella olevasta napista painamalla saa lisättyä rahaa. Kun paketin lähettää matkaan niin omista rahoista miinutetaan matkan kustannus. Jos rahat eivät riitä niin paketti ei lähde liikkeelle
- 4. Mielestäni tässä on ominaisuuksia ainakin 15 pisteen verran. Perustelut voisivat olla vaikka yläpuolella luetellut ominaisuudet.
- 5. Luokkakaavio:

Application

mainClass

- + main(args: String[]): void
- start(stage: Stage): vold

Initializable

FXMLDocumentController

- addMoneyBtn: Button
- alldata: String
- button1: Button
- buffon11: Buffon
- ~ citycheck: String
- cityComboBox: ComboBox<String>
- citylistwo: ObservableList<String> = FXCollections.o...
 oostLabel: Lahel
- Indexofselectedpacket: Int
- InfoLabel: Label
- label: Label
- lationtoindex: ArrayList<String> = new ArrayList<>().
- money: float
- moneyAmount: Label
- moneyLabel: Label
- moneySilder: Silder
- otherDataFromPackets: ObservableList<String> = FXCollections.o...
- packetcheckernumber: Int = 0
- packetolassnumber: Int
- packetComboBox: ComboBox<String>
- + packetinfodata: String
- packetinfodatalist: ObservableList<String> = FXCollections.o...
- packetSizeList: ObservableList<integer> = FXCollections.o...
- warningLabel: Label
- webview: WebView
- addMoney(event: ActionEvent): void
- addToMap(event: ActionEvent): void
- deleteRoutes(event: ActionEvent): void
- + getioniat(a: float, b: float, c: float, d: float, packetclass: float): vold + getOtherDataFromPackets(breakable: String): vold
- + getPacket(packet: String): String
- Initialize(url: URL, rb: ResourceBundle): vold
- newPacket(event: ActionEvent): vold refreshPackets/event; ActionEvent); void
- sendPacket(event: ActionEvent): vold
- showIndexNumber(event: Event): vold
- silderChanged(event: MouseEvent): vold

Initializable

FXMLPacketinfoController

+ Initialize(url: URL, rb: ResourceBundle): vold

Initializable

FXMLpacketMakerController

- addresschecker: String
- breakable: CheckBox
- citycheck: String
 citychecker: Strin citychecker: String
- ~ cit/listtwo: ObservableList<String> = FXCollections.o... endAutomat: ComboBox<String>
- endAutomat: ComboBox<String>
- endCity: ComboBox<String>
- endlat: float
- endion: float firstClass: RadioButton
- ItemList: ObservableList<String> = FXCollections o____
- ItemsListCombo: ComboBox<String>
- packetDoneLabel: Label
 packetInfostring: String
- packetinfostring: String
- packetName: TextFleid
- packetSize: TextFleid
- packet//elght: TextFleld
- postcodechecker: String secondClass: RadioButton
- startAutomat: ComboBox<String>
- startCity: ComboBox<String>
- startlat: float
- startion: float
- thirdClass: RadioButton
- checkfirstautomat(event: Event): vold
- checklonlatend(event: Event): vold.
- checklonlatstart(event: Event): vold
- checksecondautomat(event: Event): vold
- createPacket(event: ActionEvent): void enableDisableRadioBtns(event: Event): vold
- + getPacket(packet: String): String + Initialize(url: URL, rb: ResourceBundle): vold
- packetinfo(event: ActionEvent): vold