



jamk

Kuljetussovelluksen suunnittelu

Tomi Hyvärinen

Opinnäytetyö, AMK

Toukokuu 2024

Tieto- ja viestintätekniikan tutkinto-ohjelma

Hyvärinen, Tomi

Kuljetussovelluksen suunnittelu

Jyväskylä: Jyväskylän ammattikorkeakoulu. Toukokuu 2024, 32 sivua.

Tieto- ja viestintätekniikan tutkinto-ohjelma. Opinnäytetyö AMK.

Julkaisun kieli: suomi

Julkaisulupa avoimessa verkossa: kyllä

Tiivistelmä

Sovellussuunnittelu on iso osa onnistuneen sovelluksen toteutumista. Sovellussuunnitelmalla onkin tarkoitus auttaa sovelluskehitystä toteuttamaan toimiva sovellus, jossa on huomioitu sovelluksen toteutumisen aikataulu, sovelluksen ulkoasu ja toiminnot, tekninen suunnittelu kuten tietokanta. Sovellussuunnitelmaa on tärkeä dokumentoida sillä, sovelluskehityksen aikana sovellussuunnitelma päivittyy sovelluksen muutosten ja uusien ideoiden myötä.

Sovellussuunnittelu toteutettiin tulevalle startup-yritykselle, jonka on tarkoitus helpottaa paketin lähettämistä ja vastaanottamista. Sovellussuunnittelun seuranta on toteutettu päiväkirja muodossa, jonka avulla pyritään tuomaan esille sovellussuunnittelun tärkeys, miten sovellussuunnittelu on tapahtunut ja toteutunut. Seurannassa on myös pyritty selvittää käytettyjä teknologioita, niiden toimintaa ja miten niitä voitaisiin käyttää.

Seurannassa havaittiin että, jos sovellussuunnittelu on tehty huolella, tämä helpottaa sovelluksen seuraavan vaiheen toteuttamista. Seuranta auttaa ymmärtämään miten voidaan toteuttaa sovellussuunnitelma ja mitä teknologioita suunnittelussa voidaan hyödyntää. Myös sovelluksen suunnittelun aikataulutusta ja sen ymmärtäminen on tärkeää. Seurannan avulla voitiin huomata, että aikataulu, mikä oli suunniteltu ei aina pitänyt paikkaansa, ja tämä tulisi ottaa huomioon sovellussuunnittelussa.

Sovellussuunnittelun tärkeys on hyvä ymmärtää ja sen tekemiseen kannattaakin käyttää aikaa ja resursseja. Hyvän sovellussuunnitelman avulla ohjelmisto on mahdollista toteuttaa vaivattomasti ja aikataulussa. Hyvin toteutettu sovellussuunnittelu auttaa myös sovelluskehitystä ja sovelluksen valmistumista: hyvin dokumentoidun sovellussuunnitelman avulla sovelluksen parissa työskentelevät pystyvät helposti seuraamaan missä vaiheessa sovelluksen kehitys on, ja tarvittaessa palata jo sovellukseen toteutettuihin tapahtumiin.

Avainsanat (asiasanat)

Sovellussuunnittelu, ketterä kehitys, tietokannat, käyttöliittymä, projektin hallinta, GitLab, Figma

Muut tiedot (salassa pidettävät liitteet)

-

Hyvärinen, Tomi

Package carrier planning

Jyväskylä: Jamk University of Applied Sciences, May 2024, 32 pages.

Degree Programme in Information Technology. Bachelor's thesis.

Permission for open access publication: Yes

Language of publication: Finnish

Abstract

Application design is a significant aspect of successful application development. The purpose of it is to assist application development in implementing a functional application, considering aspects such as the application's development timeline, its appearance and features, and technical design such as the database. It is important to document the application design, as it evolves during application development due to changes in the application and new ideas.

The application design was implemented for a forthcoming startup company. The application aims to facilitate package sending and receiving. Monitoring was conducted in diary format, intended to highlight the importance of application design and how designing has occurred and been implemented. The monitoring also aimed to clarify the technologies used and their functionalities, as well as how they could be utilized in application design.

The results of the monitoring showed that careful application design facilitates the implementation of the next phase of the application. Monitoring helps understand how to implement the application design and what technologies can be utilized in it. Understanding and adhering to the schedule while designing the application is also important. Monitoring allowed us to observe that the planned schedule did not always hold true, and this should be considered in the planning.

Understanding the importance of application design and investing time and resources in it is advisable. With a good application design, software can be implemented effortlessly and on schedule. A well-documented application design also aids in application development and the completion of the application: with a well-documented application plan, those working on the application can easily track the development stage and, if necessary, revisit events already implemented in the application.

Keywords/tags (subjects)

Application design, agile development, database, user interface, project management, GitLab, Figma

Miscellaneous (Confidential information)

-

Sisältö

1	Johdanto	2
2	Sovelluksen suunnittelu	3
2.1	Ketterä kehitys	3
2.2	GitLab	4
2.3	Figma	6
2.4	Tietokanta	7
3	Päiväkirjaraportointi	8
3.1	Seurantaviikko 1	8
3.2	Seurantaviikko 2	12
3.3	Seurantaviikko 3	14
3.4	Seurantaviikko 4	17
3.5	Seurantaviikko 5	19
3.6	Seurantaviikko 6	21
3.7	Seurantaviikko 7	22
3.8	Seurantaviikko 8	25
4	Pohdinta	28
4.1	Tavoitteet ja tulokset	28
4.2	Sovellussuunnittelun onnistuminen ja kehityskohteet	28
4.3	Tulosten luotettavuus ja käytettävyys	30
4.4	Kehittyminen suunnittelijana.....	30
	Lähteet	32

Kuviot

Kuvio 1.	Mermaid kaavio	5
Kuvio 2.	Markdown tekstin muokkaus.....	6
Kuvio 3.	Sprintti.....	9
Kuvio 4.	Tunniste	9
Kuvio 5.	Tehtävätaulu	10
Kuvio 6.	Sovellussuunnitelma kaavio.....	11
Kuvio 7.	Figma kirjautuminen	13
Kuvio 8.	Figma lähetyksesiä mobiilinäkymä.....	14

1 Johdanto

Sovelluksen suunnittelu on iso osa onnistunutta projektia. Sovelluksen suunnitteluun kannattakin panostaa aikaa ja resursseja, jotta sovellussuunnitelmasta saadaan mahdollisimman tarkka ja sovelluksen toteutus olisi vaivattomampaa. Sovellussuunnittelun yksi tärkeimpiä osia on käyttöliittymän suunnittelu. Käyttöliittymän suunnittelu mahdollistaa suunnittelijoiden ja mahdollisten asiakkaiden osallistumisen sovelluksen suunnittelemiseen ja toteutukseen. Tällöin voidaan jo sovelluksen alkuvaiheessa huomata mahdolliset puutteet ja muutoskohteet, mitä sovellus voisi tulla tarvitsemaan. Hyvä käyttöliittymäsuunnitelma mahdollistaa myös työmäärä arvioinnin tarkemman laskemisen. Hyvin suunniteltu sovellus onkin jo puoliksi tehty sovellus.

Tämän opinnäytetyön tarkoituksena oli seurata, miten sovellussuunnittelu on toteutettu oikeassa projektissa. Sovellussuunnittelua seurattiin päiväkirjan muodossa. Päiväkirjaa kirjoitettiin joka päivä, jotta sovelluksen suunnittelua ja toteutusta voitiin seurata. Jokaisen viikon alussa tehtiin myös suunnitelma ja tavoitteet tulevalle viikolle ja viikon lopussa arvioitiin, miten tavoitteissa oli onnistuttu. Työssä on myös kerrottu tarkemmin sanastoa, mitä päiväkirjaa kirjoittaessa on käytetty, niin että sovelluksessa tehdyt asiat eivät jäisi epäselviksi. Sovelluksen aikataulutusta ja töiden järjestelyä toteutettiin ketterää kehitystä hyödyntäen ja sovelluksen suunnittelutyökaluina toimivat GitLab, Figma ja DBeaver.

Opinnäytetyön toimeksiantajana toimi tulevaisuuden yritys. Yritykselle suunniteltiin ja kehiteltiin sovellusta. Sovellus oli tulevaisuuden yrittäjän pitkäaikainen haave oman yrityksen perustamisesta, ja opinnäytetyö ja työharjoittelu sovelluksen parissa mahdollistavat tämän haaveen toteuttamisen. Kokemusta sovelluskehityksestä oli yrittäjälle kertynyt koulun kautta, ja tämä auttoi huomattavasti sovellussuunnittelussa. Valitettavasti sovelluksesta ei voitu esittää jokaista yksityiskohtaa sovelluksen suunnittelusta, toiminnoista ja toteutuksesta. Tämä johtui siitä, että yritysideoissa oli salassa pidettäviä toimintoja, joiden avulla yritys tulee erottumaan kilpailijoista. Tulevaisuuden tavoite yrittäjällä onkin saada toimiva sovellus, jota voidaan myös jatkokehittää ja saada sovellukselle mahdollisimman paljon käyttäjiä.

Sovellus on kuljetussovellus, joka mahdollistaa paketin lähettämisen ja vastaanottamisen. Sovelluksen tarkoituksena on tehdä paketin lähettamisestä ja vastaanottamisesta helppoa ja vaivatonta. Paketinlähetyksen suunnittelu tapahtuu niin, että kun asiakas kirjautuu sovellukseen, hän

täyttää sovellukseen sen tarvitsemat tiedot. Pakollisia tietoja ovat lähettäjän ja vastaanottajan tiedot, paketin tiedot sekä lähtöaika. Tämän jälkeen lähetys maksetaan, ja tilaus on valmis. Paketin kulkua voidaan seurata GPS:n avulla reaaliajassa. Paketin vastaanottaminen on taas huomattavasti vaivattomampaa, sillä vastaanottajan tarvitsee vain kirjautua sovellukseen. Vastaanottaja voi myös halutessaan seurata paketin kulkua GPS:n avulla ja noutaa paketti sovitusta toimituspäikasta.

2 Sovelluksen suunnittelu

2.1 Ketterä kehitys

Ketterä kehitys (agile development) on joustava tapa toteuttaa ohjelmistosuunnittelua. Ketterän kehityksen tarkoitus on mahdollistaa projektin jakamisen pienempiin osiin eli sprintteihin (sprint), joita on helpompi hallita projektia suunnitellessa ja projektin edetessä. (Agile / Ketterä kehitys - mitä ja miksi? 2024.) Luotio (2023) mukaan ketterä kehitys mahdollistaa projektiin tulevat muutokset ja jo toteutettujen osien uudelleenkäsittelyn, toisin kuin vesiputousmalli, jossa ohjelmiston elinkaari ja sen vaiheet ovat ennalta suunniteltu. Ketterässä kehityksessä käytetään erilaisia termejä kuvaamaan työn eri osia ja vaiheita, ja ne muodostavat tehtävälle työlle helposti lähestyttävän viitekehyksen. Näitä termejä ovat sprintti, kehitysjojo (backlog), issue ja käyttäjätarina (user story).

Sprintit ovat lyhyitä kehitysjaksoja, jotka kestävät viikosta jopa neljään viikkoon. Sprinttien tarkoitus on toimia asiakkaan ja projektin toteuttajien välisenä kehityskehyksenä. Asiakas ja toteuttajat pystyvät näin seuraamaan projektin kehitystä, ja sprintin jälkeen voidaan käydä keskustelu asiakkaan kanssa siitä, miten on onnistuttu ja mitä voitaisiin tehdä toisin. Sprintille asetetut tavoitteet pyritäänkin saamaan valmiiksi sprintin aikana niin, että ne olisivat testattu ja valmiina julkaistavaksi. (Luotio 2023.) Sprinteillä projekti saadaan pilkottua hyvin pienempiin osiin, ja näitä osia on helpompi hallita sekä kehittäjän että asiakkaan näkökulmasta.

Kehitysjojoissa on projektiin liittyviä toimintoja, jotka tulevaisuudessa nostetaan työn alle. Luukkaisen (2022) mukaan kehitysjojoon kerätään ohjelmistolle asetettuja vaatimuksia ja toimintoja. Kehitysjojoon voidaan myös lisätä ohjelmistoon muuten liittyviä asioita, kuten bugit tai asiakkaan esittämät sisältö-, tai toiminnallisuustoiveet. Kehitysjojo toimii eräänlaisena tehtävälistanana, josta

nostetaan työn alle seuraavalla sprintillä tehtävät toiminnot niiden arvioidussa tärkeysjärjestyksessä.

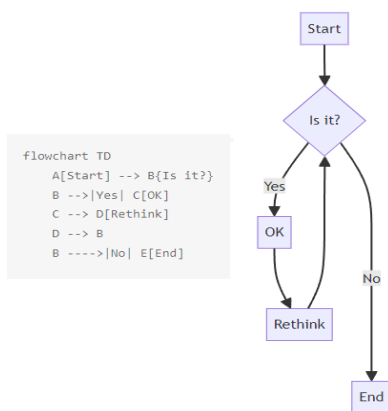
”Issues”, eli sprintillä käsiteltäviä tehtäviä, ongelmia, tai ominaisuuksia käytetään ideoiden keräämiseen, ongelman ratkaisuihin ja työn suunnitteluun (Issues 2024). Projektin teko tulisi aina aloittaa näiden läpikäynnillä: uusi projekti aloitetaan luomalla issueita, ja vanhassa projektissa käydään jo olemassa olevat issueit läpi, jottei vahingossa luoda duplikaatteja. Pienissä projekteissa ei välttämättä tarvitse luoda issueita lainkaan. Issueiden ja kehitysjonon avulla tiimi saa kerättyä tehtävät yhteen paikkaan, ja myös mahdollinen asiakas pääsee vaikuttamaan ja seuraamaan ohjelmiston kehitystä. (Always start with an issue 2016.)

Käyttäjätarinalla kuvataan tarkemmin yhtä ohjelmistoon tehtävää toiminnallisuutta. Käyttäjätarina voi toimia tapaus, jossa käyttäjä haluaa esimerkiksi kirjautua sovellukseen pankkitunnuksilla. Käyttäjätarinoilla ei kuitenkaan oteta kantaa siihen, miten toiminto tulisi teknillisesti toteuttaa. Hyvin tehdystä käyttäjätarinasta saa selkeän kuvan siitä, mitä tullaan tekemään, mutta se antaa myös vapaammat kädet ohjelmiston suunnittelijalle. (Virtanen 2022.)

2.2 GitLab

GitLab on versionhallintapalvelu, jota käytetään yhdessä Git-versionhallintajärjestelmän kanssa. GitLab mahdollistaa projektien ylläpidon niin ryhmille, pienille tiimeille kuin yksityishenkilöille. GitLab auttaa projektin seurannassa, sillä sen avulla voidaan helposti seurata mitä projektissa on tapahtunut, ja mitä tullaan tekemään seuraavaksi. GitLabin avulla voidaan myös helposti tarkastella projektin historiaa: mitä muutoksia on tehty ja milloin, ja kenen muutokset ovat. Lisäksi muutosten tekijöiden kommentit auttavat ymmärtämään miksi nämä muutokset ovat tehty. GitLabia voidaan käyttää verkossa, tai sen voi ladata omalle koneelle. (Kaukojarju 2017.) Yleisimmin GitLabia käytetään verkon kautta, jotta projektin jakaminen olisi helpompaa. GitLabiin voidaan luoda issueita kehitysjonoon, josta ne nostetaan tärkeysjärjestyksessä sprintille, josta ne halutaan toteuttaa. Kun issue nostetaan sprintille, issuen elinkaarta on helppo seurata tehtävätaulusta (issueboard). Tehtävätaulusta on sovelluskehittäjän helppo aloittaa uusi issue. GitLabia käytetään myös sovelluksen lähdekoodin tallentamiseen ja jakamiseen, jotta jokainen projektiin osallistuja saisi käsiinsä uusimman version sovelluksesta. GitLabin avulla on myös mahdollista julkaista sovellus GitLabin omien työkalujen avulla.

GitLabin dokumentointia kirjoitetaan yleensä markdownina, sillä GitLabissa on sisäänrakennettu markdown-kääntäjä, joka mahdollistaa selkeät dokumentaatiot. Markdown on yksinkertainen tekstipohjainen merkintäkieli. Sen avulla on mahdollista muuttaa teksti XHTML- tai HTML-muotoon web-kirjoittajille. Markdown-kirjoitusta on helppo kirjoittaa ja lukea. (Gruber 2004; Ifloadfocus n.d.) Markdownin kaksi tärkeintä ominaisuutta ovat yksinkertainen tyyllittely, kuten lihavointi, kursivointi ja luettelomerkinnät, sekä muuntaminen HTML:ksi, joka helpottaa verkkosisällön kirjoittamista, jos verkkosivujen kirjoittaminen ei ole muuten tuttua HTML:llä. (Ifloadfocus n.d.) GitLabissa käytetään GitLab Flavored Markdownia, joka eroaa standardi markdownista jonkin verran. GitLabin käytössä olevassa markdownissa on muun muassa erikseen tehtyjä lisäosia, jonka avulla voidaan käyttää esimerkiksi Mermaid -työkalua, joka mahdollistaa kaavioiden lisäämisen markdown-tyyppisen kielen avulla ilman erillisiä kaavion piirto-ohjelmia. (GitLab Flavored Markdown (GLFM) 2024.) Kuviossa 1 nähdään mermaid-kaavio (diagram), jossa on nähtävissä sekä se, miten markdownia kirjoitetaan, sekä se, miltä kaavio näyttää lukijalle lopullisessa muodossaan (ks. kuvio 1). Yleisesti jokaisessa GitLab-projektissa on README.md tiedosto, joka sijaitsee projektin etusivulla ja se sisältää yleiskuvauksen projektista ja sen mahdollisista toiminnoista.



Kuvio 1. Mermaid kaavio (Flowcharts - Basic Syntax 2024, muokattu)

Toisessa esimerkissä (ks. kuvio 2) näytetään joitain markdown tekstin tyylivaihtoehtoja, ja siitä, miten yksinkertaisilla muutoksilla markdown mahdollistaa tyylien muokkaamisen.

Style	Syntax	Keyboard shortcut	Example	Output
Bold	** ** or __ __	Command + B (Mac) or Ctrl + B (Windows/Linux)	**This is bold text**	This is bold text
Italic	<i>* *</i> or <i>_ _</i>	Command + I (Mac) or Ctrl + I (Windows/Linux)	<i>_This text is italicized_</i>	<i>This text is italicized</i>
Strikethrough	~ ~	None	~This was mistaken text~	This was mistaken text
Bold and nested italic	<i>** **</i> and <i>_ _</i>	None	<i>**This text is extremely important**</i>	This text is extremely important
All bold and italic	<i>*** **</i>	None	<i>***All this text is important***</i>	<i>All this text is important</i>
Subscript	<code><sub> </sub></code>	None	This is a <code><sub>subscript</sub></code> text	This is a _{subscript} text
Superscript	<code><sup> </sup></code>	None	This is a <code><sup>superscript</sup></code> text	This is a ^{superscript} text

Kuvio 2. Markdown tekstin muokkaus (Basic writing and formatting syntax 2024)

2.3 Figma

Figma on verkkosivustojen ja mobiilisovellusten käyttöliittymän suunnitteluun tarkoitettu ohjelmisto, jonka avulla voidaan toteuttaa käyttöliittymän suunnittelu ilman ohjelmointia. Figma mahdollistaa interaktiivisen sovelluksen toteuttamisen alustavan suunnitelman avulla: Figmasta löytyvällä esitystilalla voidaan havainnollistaa miltä tuleva sovellus voisi tulla näyttämään, ja mitä toimintoja se voisi pitää sisällään. Sovelluksen käyttöliittymän suunnittelu kannattaakin tehdä ennen varsinaisen sovelluksen toteuttamista, sillä siten saadaan kaikille sovelluksen suunnitteluun osallistuville selkeä kuva mitä ollaan tekemässä ja varmistetaan, että käyttökokemus sovelluksen parissa olisi laadukasta. (Piispanen 2021.)

Figma-sovellusta voi käyttää selaimessa tai omalle koneelle ladattavalla sovelluksella. Figmasta on olemassa maksullinen ja ilmaisversio. Ilmaisversiolla pärjää pienissä projekteissa ja se on kätevä Figman käytön opetteluun, mutta jos suunnitteluun osallistuu enemmän tekijöitä, tarvitaan jo maksullinen versio. Figman maksullinen versio onkin hyvä silloin, kun sovelluksen suunnitteluun osallistuu monta suunnittelijaa. Se myös mahdollistaa asiakkaiden osallistumisen sovelluksen suunnitteluun, jolloin asiakas voi antaa palautetta ja ehdottaa mahdollisia muutosideoita jo suunnitteluvaiheessa. (Mts.)

Figmassa suunnittelu tapahtuu näkymissä (frame), joiden sisälle voidaan tuoda valmiita komponentteja kuten painikkeet. Painike on interaktiivinen elementti, ja näiden avulla voidaan liittää

näkymiä toisiinsa. (Mts.) Esimerkiksi kirjautuminen: luodaan kirjautumisnäkyvä, johon tuodaan "kirjaudu"-painike, joka liitetään "kirjautunut käyttäjä"-näkymään. Esitystilassa "kirjaudu"-painikkeesta painaessa kirjautumisnäkyvä vaihtuu "kirjautunut käyttäjä"-näkymään.

2.4 Tietokanta

Hyvin suunniteltu ja toteutettu tietokanta on sovelluksen yksi tärkeimpiä ominaisuuksia. Tietokannan suunnittelu ja toteutus kannattaa tehdä huolella, sillä huonosti suunniteltuun tietokantaan voidaan joutua tekemään isojakin muutoksia vielä ohjelmointi vaiheessa, ja tämä hidastaa sovelluksen valmistumista. Pahimmassa tapauksessa sovelluksen valmistumisen jälkeenkin voidaan joutua muokkaamaan tietokantaa esimerkiksi huonon suorituskyvyn takia.

Aaltonen (2023) kuvailee tietokannan olevan järjestelmä, jonne tallennetaan haluttua tietoa järjestelmällisesti, jotta tiedon hakeminen tietokannasta olisi helppoa ja nopeaa. Tietokannassa olevaa tietoa voidaan lisätä, poistaa ja muokata. Tietokanta sisältää tauluja, jotka muodostuvat sarakkeista ja riveistä. Kilpeläinen (n.d.) kertoo tarkemmin, miten tietokantataulut ja sarakkeet tulisi luoda. Tietokantataulun sarakkeelle määritellään minkä tyyppistä tietoa voidaan tallentaa. Eri tietotyyppejä ovat esimerkiksi numeerinen, merkkijono ja aika. Jokaisessa taulussa tulee olla myös perusavain (primary key). Taulut voivat myös sisältää viiteavaimen (foreign key), jota käytetään tietokantataulujen yhteyksien luomiseen, jolloin puhutaan taulujen välisestä relaatiosta. (Mts.) Relaatioista koostuvaa tietokantaa kutsutaan relaatiotietokannaksi. Tietokantojen luomiseen ja tiedon hakemiseen käytetään SQL-kieltä, joka on yksi ohjelmointikielistä, jota käytetään lähes kaikissa relaatiotietokannoissa. Tietokannan hallintaan käytetään yleensä jotain tietokannan hallintajärjestelmää (DBMS). Suositujia hallintajärjestelmiä ovat MySQL, Microsoft Access, Microsoft SQL Server, FileMaker Pro, Oracle Database ja dBASE. (Mitä tietokanta tarkoittaa? n.d.) Tietokantoja voi myös luoda ja hallita graafisen ohjelmiston avulla.

Peterson (2023) kuvaa, miksi tietokannan suunnittelu on tärkeää ja mitä tietokannan suunnitteluun kuuluu. Oikein suunniteltu tietokanta on helppo päivittää ja se ei vaadi paljon tallennustilaa, joten tallennuspaikan pitäminen on halpaa. Tietokantajärjestelmän on tarkoitus palvella käyttäjien vaatimuksia, ja sen on tarkoitus olla suorituskykyinen. Tietokannan luomiseen on kolme eri vaihetta, ja näitä vaihteita kutsutaan tietokannan elinkaareksi. Elinkaareen kuuluvat vaatimusana-

lyysi, tietokannan suunnittelu ja toteutus. Vaatimusanalyysi sisältää tietokannan elinkaaren suunnittelun ja järjestelmän määrittämisen. Ensimmäisen vaiheen suunnittelun avulla luodaan tietokannan elinkaarelle suunnitelma siitä, mitä tehdään missäkin elinkaaren vaiheessa ja huomioidaan projektin tietojärjestelmästrategia. Järjestelmäanalyysin avulla määritellään tietokannan laajuuden ja rajat. Tietokannan suunnitteluvaiheeseen kuuluu kahden eri mallin suunnittelu: looginen ja fyysinen. Loogisen mallin suunnittelussa suunnitellaan vaatimukseen perustuva tietokantamalli, joka yleensä tehdään ensin kynän ja paperin avulla. Tarkoituksena on luoda ER-kaavio (Entity-Relationship-diagram), joka määrittelee tietokantataulut ja niiden väliset mahdolliset riippuvuudet. Fyysisessä mallissa käytetään apuna loogisen mallin suunnitelmaa, johon otetaan mukaan tietokannan hallintajärjestelmä ja mahdolliset fyysiset toteutustekijät. Viimeisenä tietokannan luomisen elinkaaren vaiheena on täytöntöönpano, johon kuuluvat tietojen muuntaminen ja lataus, sekä testaus. Tietojen muuntamisella ja latauksella tarkoitetaan suunnitellun relaatiotietokannan tuomista käytäntöön ja mahdollisesti muuntamista vanhasta järjestelmästä uuteen tietokantaan. Testaus nimensä mukaan tarkoittaa luodun tietokannan testausta mahdollisten virheiden ja vaatimusten täyttämiseksi.

3 Päiväkirjaraportointi

3.1 Seurantaviikko 1

Ensimmäisen viikon tavoitteena olisi saada suunnitteluympäristö pystyyn GitLabiin, sekä toteuttaa myös alustava suunnitelma toteutettavasta ohjelmistosta. Tässä hyödynnetään ketterän kehityksen menetelmää. Tarkoituksena olisi myös luoda markdown-dokumentti, johon kerätään tietoa, miten ja miksi ohjelmistoa kehitetään, ja kuka on ohjelmiston kehittäjä. Jos aikaa jää, luodaan kehitysjonoon issueita, joita tullaan nostamaan sprinteille tärkeysjärjestyksessä.

Maanantai 25.03.2024

Päivä alkoi GitLab-ympäristön suunnittelulla. Miten lähdän rakentamaan GitLab ympäristöä järkevästi? Tämä täytyisi tehdä niin, että kuka vain pystyisi tulemaan projektiin mukaan ja saisi helposti selville mikä sovellus on, ja missä vaiheessa projektia ollaan menossa. Aloitin suunnittelun luomalla GitLabiin virstanpylväitä (milestone) (ks. Kuvio 3), joiden avulla projekti voidaan aikataulut-

taa. Nämä ovat yleensä isompia etappeja projektissa, esimerkiksi sovelluksen valmiin käyttöliittymäsuunnitelman tarkastus. Seuraavana oli vuorossa sprintit, jotka loin samalla tavalla kuin virstanpylväät. Sprintin pituudeksi asetin alustavasti kaksi viikkoa. Sprinteillä onkin tarkoitus jakaa virstanpylväs pienempiin osiin ja sprinttien valmistumisen myötä saavutetaan virstanpylväälle asetettu tavoite. Tein myös kehitysjono-nimisen virstanpylvään. Tähän on tarkoitus kerätä kaikki issueet, jotka nostetaan kehitysjonosta sprintille issueen tärkeysjärjestys huomioiden.

Sprint 9 Jul 15, 2024–Jul 28, 2024 Upcoming a1t0 / Autax	0 Issues · 0 Merge requests 0% complete	↑ Close Milestone
Sprint 10 Jul 29, 2024–Jul 31, 2024 Upcoming a1t0 / Autax	0 Issues · 0 Merge requests 0% complete	↑ Close Milestone
Backlog Open a1t0 / Autax	6 Issues · 0 Merge requests 0% complete	↑ Close Milestone

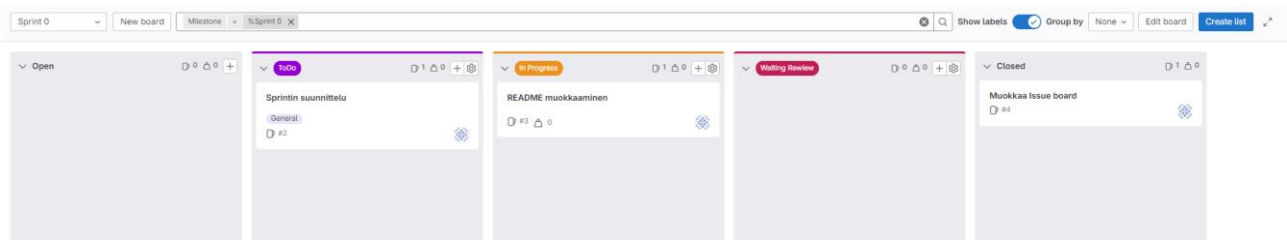
Kuvio 3. Sprintti

Virstanpylväiden jälkeen vuorossa oli tunnisteet (label) (ks. Kuvio 4). Tunnisteet auttavat määrittämään, mihin kategoriaan issue tulee sijoittumaan. Tunnisteita loin: ToDo-tunniste, jonne tullaan sijoittamaan seuraavana vuorossa olevat työt, In Progress -tunniste, josta käy ilmi tekeillä olevat työt ja Waiting Review -tunniste, jossa työn valmistuttua toinen osapuoli tarkistaa, että kaikki on kunnossa ja hyväksyy työn tehdyksi. Edellä mainittuja tunnisteita tarvitaan, jotta voidaan hahmottaa missä vaiheessa issue on menossa. Loin myös backlog-tunnisteen, jotta issueet saadaan ensiksi kehitysjonoon, General-tunnisteen, joka voidaan määrittää yleisille tehtäville, sekä Figma-tunnisteen, joka on tarkoitettu ulkoasun suunnitteluun. Lopuksi loin vielä erikseen Task-tunnisteen, joka on sovelluksen suunnitteluun ja User Story -tunnisteen, joka on käyttäjätarinoita varten.

Backlog a1t0 / Autax	Issues Merge requests ☆ Subscribe ⋮
Figma UI suunnittelu a1t0 / Autax	Issues Merge requests ☆ Subscribe ⋮
General a1t0 / Autax	Issues Merge requests ☆ Subscribe ⋮
In Progress a1t0 / Autax	Issues Merge requests ☆ Subscribe ⋮
Task Suunnitteluun ja toteutukseen liittyvät tehtävät a1t0 / Autax	Issues Merge requests ☆ Subscribe ⋮
ToDo a1t0 / Autax	Issues Merge requests ☆ Subscribe ⋮
User Story Käyttäjänä näkökulmasta, käyttäjänä minä haluaisin... a1t0 / Autax	Issues Merge requests ☆ Subscribe ⋮
Waiting Review a1t0 / Autax	Issues Merge requests ☆ Subscribe ⋮

Kuvio 4. Tunniste

Tämän jälkeen pääsin issueiden luomiseen. Issue sisältää kuvauksen työstä, mille virstanpylväälle työ kuuluu ja mikä on sen tunniste. Työn toteuttaja määritellään myös mahdollisesti tässä vaiheessa. Loin vain muutaman issueen alkuun, jotka tiesin tulevani tekemään ensimmäisen viikon aikana, kuten Readme-tiedoston muokkamisen ja sprintin suunnittelun. Issues sijoitin kehitysjonon kautta sprintille 0, jotta saisin ensimmäiselle viikolle suunnitellut työt näkyviin tehtävätauluun (ks. Kuvio 5). Tehtävätaulu osoittautui todella hyväksi työkaluksi, sillä sen avulla jokainen projektissa toimiva voi seurata mitä töitä on suunniteltu ja missä vaiheessa työ on menossa.



Kuvio 5. Tehtävätaulu

Tiistai 26.03.2024

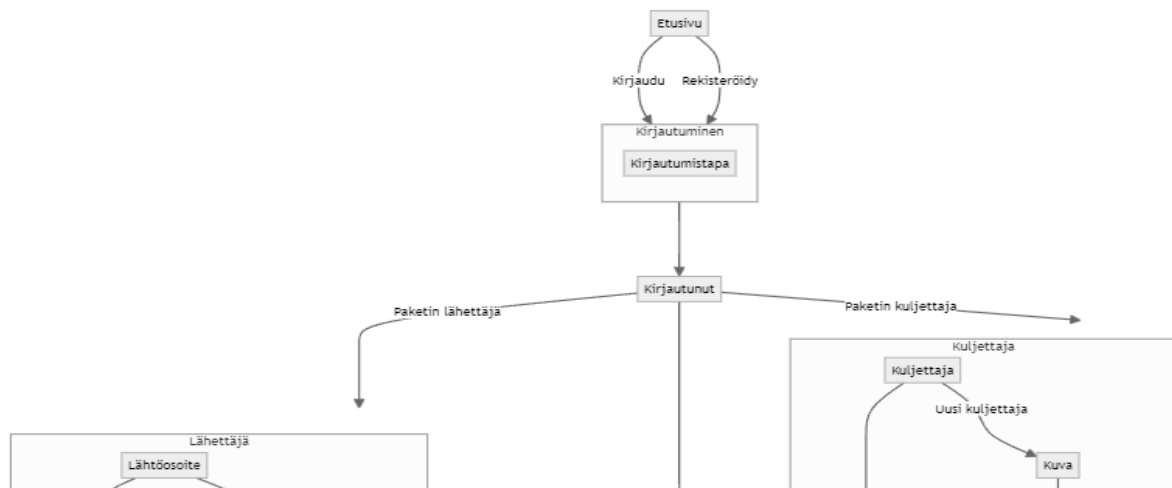
Tämän päivän tavoitteena oli kirjoittaa GitLabiin yleinen kuvaus siitä, mikä sovellus on ja miksi sovellusta tulisi käyttää. Sovelluksen esittely näkyy GitLabin etusivulla, jotta sovelluksen toteutukseen osallistuvat näkevät ensimmäisenä mikä sovellus on ja miten tämän on tarkoitus toimia. Suunnitelma kirjoitettiin GitLabin etusivulla olevaan README.md tiedostoon käyttäen markdown-kieltä. Sain kirjoitettua sovelluksen kuvauksen hyvin auki, ja avasin siinä enemmän sitä, mikä sovellus on ja miksi sitä tulisi käyttää. Keräsin myös sovelluksen hyvät ja huonot puolet listaksi, jota tulevaisuudessa voidaan päivittää ja mahdollisesti käyttää sovelluksen kehittämiseen. Sain myös luotua parit uudet issues valmiiksi seuraavia sprinttejä varten.

Keskiviikko 27.03.2024

Tänään vuorossa oli suunnitella kaavio siitä, miten sovellus tulisi toimia. Kaavion suunnittelu toteutettiin Mermaidin avulla. Kaaviossa kävin mahdollisimman tarkasti läpi sovelluksen toimintoja, mitä tulee tapahtumaan minkäkin valinnan jälkeen. Tekemäni kaavio tulee auttamaan suunnitteli-

joita jatkossa luomaan käyttäjätarinoita ja hahmottamaan mitä kaikkea tulee ottaa huomioon sovellussuunnittelussa. Kaavion tekoon meni koko päivä. Valitettavasti en voi esittää kokonaisuudessaan kyseistä kaaviota (ks. Kuvio 6), koska kaaviossa on salassa pidettävää tietoa, koskien sovelluksen toimintointoja.

Paketin lähettäminen ja toimittaminen



Kuvio 6. Sovellussuunnitelma kaavio

Torstai 28.03.2024

Pääsiäislomaa edeltävänä päivänä kirjoitin sovelluksen historiaosuuden. Historiaosuudessa kerroin mistä sovelluksen idea on peräisin, ja miten olen suunnitellut sovelluksen toimivan. Kerroin myös vähän omasta taustastani, kuten työelämästä ja koulutuksista, koska nämä vaikuttivat sovellusidean keksimiseen. Siistin myös README.md tiedostoa, jotta tulevaisuudessa projektiin osallistuvat saisivat hyvän ensisilmäyksen projektista.

Viikon 1 yhteenveto

Ensimmäinen viikko meni suunnittelun merkeissä, miten sovellus tulisi toimimaan ja mitä seuraavaksi on vuorossa. Kertasin myös markdownin ja GitLabin käyttöä, kuten miten luodaan sprintit,

jotta työn suunnittelu ja seuraaminen olisi helpompaa. Jatkossa GitLab tulee olemaan mukana sovelluskehityksessä aikataulutuksessa, työn seurannassa, suunnittelussa ja lähdekoodin tallennuspaikkana.

3.2 Seurantaviikko 2

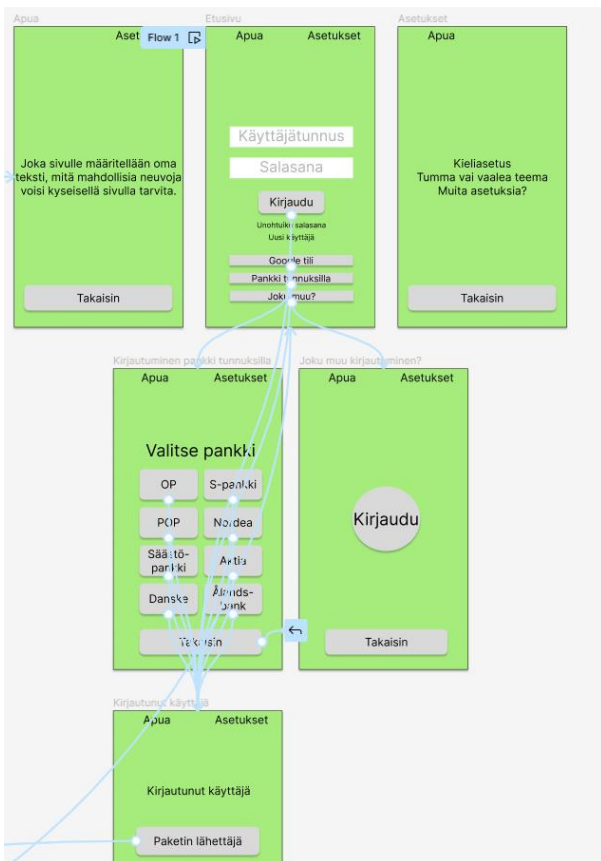
Toisen viikon tavoitteena on opetella Figmaa käyttöä ja suunnitella miltä mobiilisovellus voisi tulla näyttämään. Tavoitteena on myös suunnitella, mitä toimintoja sovellus tulisi tarvitsemaan, jotta käyttäjän tai käyttäjien olisi helppo käyttää sovellusta. Käyttöliittymän (user interface) suunnitteluun tulee menemään aikaa, koska sovelluksessani on paljon ”liikkuvia osia”.

Tiistai 02.04.2024

Tänään opettelin Figmaa käyttöä videoiden ja verkkomateriaalien avulla. Selvitin mihin kaikkeen Figmaa voidaan käyttää, ja mitä kaikkea sillä voidaan toteuttaa. Pääsin myös alkuun omassa sovelluksessani. Suunnittelin vähän miltä kirjautumisikkuna tulisi näyttämään, ja miten kirjautuminen tulisi tapahtumaan. Tein myös komponentteja muun muassa painikkeista, jotta voisin käyttää näitä uudelleen, eikä tarvitsisi aina tyyliä uudelleen monesti käytettyjä käyttöliittymän osia. Mietin, mitä kirjautumisvaihtoehtoja voisi olla ja mitä kaikkea tietoa käyttäjistä tarvitaan, jotta voitaisiin kirjautua sovellukseen.

Keskiviikko 03.04.2024

Käyttöliittymän suunnittelu Figmalla jatkuu. Sain suunniteltua kirjautumisikkunoiden ulkoasun ja toiminnot alkuvaiheeseen (ks. Kuvio 7). Rupesin myös suunnittelemaan sovelluksen lähetysosoiteen tarvittavaa käyttöliittymää. Mietin mitä tietoa voitaisiin tarvita, jotta lähetysosoite onnistuisi, kuten etunimi, sukunimi, osoite ja puhelinnumero. Tässä täytyi myös miettiä sovelluksen eri käyttäjäryhmiä. Pääsin jo hyvin vauhtiin lähetysosoiteen ulkoasun ja toimintojen suunnittelussa.



Kuvio 7. Figma kirjautuminen

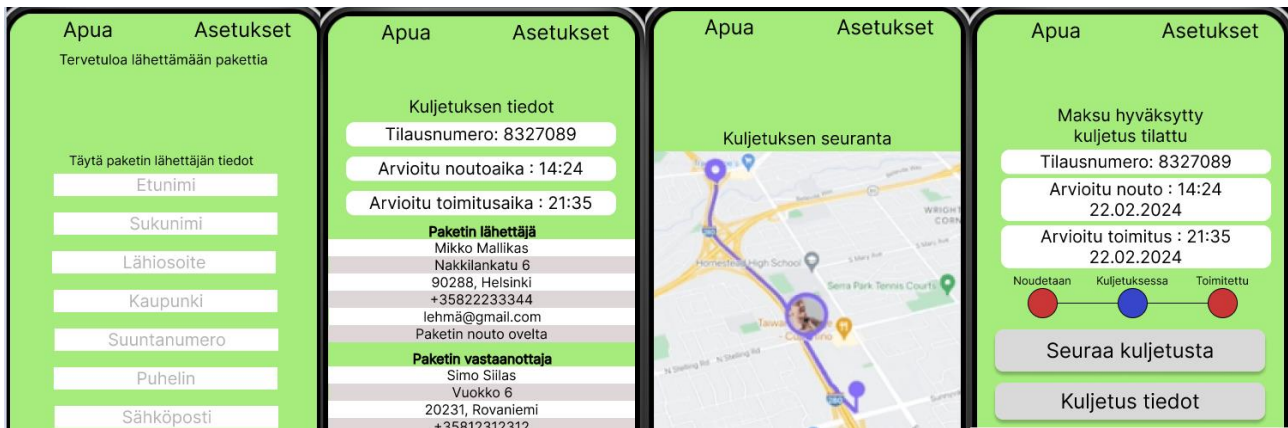
Torstai 04.04.2024

Tänään tavoitteena oli saada lähetysoisuuden käyttöliittymän suunnittelu valmiiksi Figmalla. En ihan päässyt päivän tavoitteeseen, mutta sain paljon suunniteltua ulkoasua ja toimintoja lähetysoisuutta koskien. Toteutin käyttöliittymäsuunnittelun lähetysoision osalta ja määrittelin, mitä tietoja tulisi käyttäjältä kerätä, mitkä ovat tarvittavat tiedot vastaanottajasta, miten seurata lähetyksen kulkua, miten tarkastella lähetyksen tietoja, miltä mahdollisesti tulisi näyttämään tilausvarmistusikkuna ja miten mahdollinen maksaminen tapahtuisi.

Perjantai 05.04.2024

Kohti viikonloppua, tänään suunnittelin loppuun miltä lähetysoisio tulisi näyttämään mobiilissa (ks. Kuvio 8), ja mitä mahdollisia toimintoja lähetysoisio tulisi tarvitsemaan. Suunnittelin lähetysoisioon

myös arvioinnin lähetyksestä, toimitettujen lähetysten tarkastelun, kuljettajan tarkemmat tiedot ja mahdollisten muiden omien kuljetuksien seuranta mahdollisuuden.



Kuvio 8. Figma lähetykskehyksiä mobiilinäkymä

Viikon 2 yhteenveto

Tämä viikko sujui leppoisasti, vaikka en kerennytkään saada sovelluksen käyttöliittymän suunnittelua Figmalla loppuun asti. Sovelluksen toimintojen suunnittelussa tuli aina jotain uutta mieleen, mitä tarvittaisiin sovellukseen, jotta kaikki tarvittava tieto saataisiin kerättyä. Lähetyksessä käytettävät toiminnot ja ikkunat paisuivat kuin ”pullataikina”. Tällä viikolla opin käyttämään Figmaa hyvin ja sain käsityksen mitä kaikkea Figmalla voidaan toteuttaa: linkitykset sivujen välillä nappien ja tekstien avulla, ulkoasun suunnittelun ja komponenttien tärkeyden, joka auttaa käyttöliittymän suunnittelussa. Sovelluksen käyttöliittymän suunnittelu onkin jo todella hyvässä vaiheessa, ja ensimmäinen versio kirjautumisesta ja lähetysoasiosta ovat nyt valmiina. Näihin tulee kuitenkin varmasti vielä matkan varrella muutoksia.

3.3 Seurantaviikko 3

Tämän viikon tavoitteena olisi saada ohjelmiston käyttöliittymän suunnittelu loppuun Figmalla. Viikon tavoitteet ovatkin kuljettajan ja vastaanottajan roolin liittyvät toiminnot ja toimintojen sivujen ulkoasun suunnittelu. Tarkoituksena on myös aloittaa suunnitteleminen, ja kartoittaa sovelluksessa käytettäviä eri teknologioita.

Maanantai 08.04.2024

Tänään aloitin vastaanottajan ominaisuuksien suunnittelun Figmalla. Tämä oli huomattavasti pienempi kokonaisuus kuin lähettäjän ominaisuuksien tekeminen. Sain vastaanottajan käyttöliittymäsuunnitelman valmiiksi. Osa vastaanottajan komponenteista on samoja kuin lähettäjällä, joten tämä helpotti vastaanottajan ominaisuuksien toteutusta.

Tiistai 09.04.2024

Tänään oli vuorossa Figmassa kehysten siistimistä. Jaoin kehukset omiin ryhmiin, mitkä kuuluvat kirjautumiseen, mitkä lähetykseen, mitkä vastaanottoon ja mitkä ovat yhteisiä. Tämä auttaa hahmottamaan ja muokkaamaan suunnitelmaa tulevaisuudessa. Poistin myös kehyksistä ylimääräisiä yhteyksiä: näitä kertyy, kun käyttää samoja komponentteja uudelleen ja luo uusia yhteyksiä kehysten väleillä. Kävin läpi myös nimeämisiä, jotta jokainen nimi kuvaisi paremmin mikä komponentti on kyseessä.

Keskiviikko 10.04.2024

Työskentely Figman parissa jatkui, tänään vuorossa oli paketinkuljettajan toimintojen suunnittelu. Sain paketin kuljettajan kirjautumisen tehtyä ja mallinnettua miltä kuljettajan henkilökortti tulisi näyttämään. Lisäsin jo aikaisemmin suunniteltuun kirjautumiskehykseen kuvan ottamisen ja uuden kentän, jossa pyydetään henkilötunnusta, jotta voidaan varmistaa, että sama henkilö pysty toimimaan kuljettajana yrityksessä. Henkilötunnuksen avulla pyritään myös estämään sovelluksen väärinkäyttöä, jos vaikka kuljettaja on aikaisemmin rikkonut yrityksen sääntöjä, jolloin hänellä ikuinen porttikielto yrityksen kuljettajaksi. Henkilökortti luodaan kuljettajalle, jotta paketinlähettäjä ja vastaanottaja pystyisivät tunnistamaan paketinkuljettajan ja kuittaamaan QR-koodin avulla paketin lähteneeksi tai toimitetuksi.

Torstai 11.04.2024

Lähettäjän toimintojen suunnittelu Figmalla jatkui. Tänään oli vuorossa pakettien valinta sekä valittujen pakettien tarkastelu ja poistaminen. Kuljettajalle jaetaan paketit ja tiedot paketeista mitkä

hänen tulee toimittaa. Kuljettajan on mahdollista tarkastella pakettien tietoja, kuten lähtöpaikka, toimitusosoite ja paketin fyysiset tiedot. Kuljettaja voi hyväksyä paketit kuljetettaviksi tai hylätä paketin kuljetukset.

Perjantai 12.04.2024

Tänään vuorossa oli pakettien reitin esittäminen, kuljetuksen arviointi ja jos paketin toimituksessa tulee ongelmia, mahdollinen yhteydenotto paketin lähettäjään tai vastaanottajaan. Tein kehityksen, joka näyttää nopeimman reitin paketin toimituksille, niin nouto kuin toimitustilanteessa. Loin myös erillisen kehityksen, josta kuljettaja näkee osoitteet ja arvioidun aikataulun paketin noudoille ja toimituksille. Nämä toiminnot tulevat auttamaan kuljettajaa toimitusten toimittamisessa ja antavat aikataulun pakettien toimituksille. Tein myös uuden kehityksen arviointiin, jotta paketin kuljettaja pystyy arvioimaan paketin toimituksen sujuvuuden, kuten onko paketin lähettäjä tai vastaanottaja ollut paikalla sovittuun aikaan tai onko jotain muuta ilmennyt paketin toimituksen yhteydessä. Tein myös yhteydenottokehityksen paketin tietojen yhteyteen, jotta kuljettaja voi ongelmatilanteessa soittaa paketinlähettäjälle ja vastaanottajalle.

Sunnuntai 14.04.2024

Tänään oli vuorossa kokonaismarkkinoinnin kartoitusta (ks. Liite 1). Tarkoituksena oli selvittää, mitä mieltä läheiset ja sukulaiset olisivat sovelluksesta ja sovelluksen toiminnallisuuksista. Tein kyselypohjan, jonka avulla keräsin seitsemältä eri haastateltavalta vastauksia sovellukseen liittyen. Kysymykset olivat luokkaa: ammattiryhmä, ikäryhmä, käyttäisikö haastateltava sovellusta ja selvitys siitä, miksi tai miksi ei. Selvitin myös mitä haastateltava olisi valmis maksamaan sovelluksen käytöstä, ja kirjasin haastateltavan mahdolliset uudet ideat ylös. Haastattelut toteutin kasvotusten: ensimmäiseksi esitin sovelluksen Figmasta ja kerroin lyhyesti, miten sovellus tulisi toimimaan. Tämän jälkeen kävin haastateltavan kanssa läpi edellä mainitut kysymykset, ja kirjoitin vastaukset ylös. Sainkin todella paljon positiivista palautetta sovellukseen liittyen ja jokainen olisi ainakin valmis testaamaan sovellusta. Hintojen arviointi olikin vaikeinta kyselyssä, mutta tähänkin sain jonkinlaista suuntaa ja niin kuin aina kyselyn avulla sain hyviä uusia ideoita sovellukseen ja sen toteutukseen.

Viikon 3 yhteenveto

Tämä viikko sujui todella vauhdilla ja sain hyvin toteutettua toiminnallisuuksia paketin kuljettajaan liittyen, en tosin ihan päässyt vielä suunnittelemaan minkälaisen tietojärjestelmän sovellus tulisi tarvitsemaan. Niin kuin aina, uusia ideoita kertyi taas ja sovellus jatkuu paisumistaan. Erittäin tyytyväinen olen kokonaismarkkinointikyselyyn, tämän toteutin huolella ja sain vastineeksi todella paljon tärkeää tietoa, miten sovellusta voidaan parantaa ja mitä haastateltavat olivat yleisesti mieltä sovellusideastani.

3.4 Seurantaviikko 4

Viikossa neljä mennään ja paljon on vielä tehtävää, jotta päästään itse sovelluksen toteutuksen pariin. Tämän viikon tavoitteena olisi saada käyttöliittymän suunnitelma loppuun Figmalla ja myös mahdollisesti päästä toteuttamaan tietokannan suunnittelua ja toteutusta. Ajattelin myös tutkia olemassa olevien kuljetussovelluksien käyttöliittymiä ja toimintoja.

Maanantai 15.04.2024

Uusi viikko alussa. Tänään päätin pitää vapaata toimintojen suunnittelusta ja rupesin selvittämään muiden vastaavien kuljetuspalveluiden sovelluksien toiminnallisuuksia. Aloitin Foodora-sovelluksesta, tutkin käyttöliittymää ja sen toimintoja: miten vaihdetaan käyttäjän asetuksia ja miten tilataan ruokaa. Toteutin tilauksen ja seurasin tilauksen kulkua, tilauksen kulusta sain tallennettua hyviä kuvia puhelimeeni. Seuraavaksi oli vuorossa Foodoran kilpailija Wolt. Woltin sovellusta en tutkinut ihan niin tarkkaan, mutta perustoiminnot kävin läpi ja myös tutkailin miltä Woltin ulkoasu näyttää. Kävin myös Postin ja DHL:n verkkosivuilla. DHL:n sivuilla huomasin, että toimitettavan paketin valitseminen on tehty todella helpoksi. Tarvitsi vain arvioida minkä kokoinen paketti oli kyseessä: kenkälaatikko, kaksi kenkälaatikkoa vaiko kirjekuoren kokoinen. Tämä olisi hyvä liittää myös omaan sovellukseeni, sillä ihmisillä harvoin on mittanauhoja lähettyvillä, joten paketin koon arviointi kävisikin todella kätevästi visualisoinnilla. Huomasin myös, että paketin lähettäminen on hyvin kallista Postilla ja DHL:llä. Samanlaista arviointia en tarkkaan voinut tehdä Foodoran ja Woltin kanssa, sillä he toimittavat pääasiallisesti ruokatoimituksia, mutta näistä pystyin arvioimaan matkan hintaa.

Tiistai 16.04.2024

Figmalla suunnittelu jatkuu, tänään sain tehtyä valmiiksi paketin kuljettajan toiminnot. Lisäsin kehysten, minkä avulla kuljettaja voi arvioida, miten kuljetus on toteutunut, onko lähettäjä tai vastaanottaja ollut ajallaan ja onko paketin koko pitänyt paikkaansa.

Keskiviikko 17.04.2024

Tänään tein viimeisiä muokkauksia käyttöliittymä-prototyyppeini Figmaan. Kävin läpi, miten kehukset on nimetty ja muutin näitä enemmän kuvaaviksi. Lisäsin uuden kehikon paketin toimittajaroolin ympärille, kuten olin tehnyt paketin lähettäjälle ja vastaanottajalle, jotta sovelluksen toiminnot ja kehukset olisivat omilla lokeroissaan. Näin sovellussuunnittelija näkee selkeämmin, miten sovelluksen tulisi toimia ja mitkä osat liittyvät toisiinsa. Kävin myös läpi painikkeiden ja kirjoituskenttien nimeämiset, koska kopioinnin yhteydessä oli jäänyt vanha nimi, joka ei kuvannut mitä tämä toiminto tulee tekemään. Mutta nyt on selkeä ja helposti ymmärrettävä käyttöliittymä, josta on hyvä jatkaa tietokannan suunnitteluun.

Torstai 18.04.2024

Tänään oli vuorossa tietokantasuunnittelua ja suunnittelun kertausta. Kävin läpi vanhoja tietokantojani mitä olin suunnitellut ja toteuttanut. Päädyinkin että toteutan tietokanta arkkitehtuurin MySQL Workbenchillä, koska tämä on tuttu alusta itselleni. Kävin läpi, miten rakennan toimivan tietokannan, ja missä järjestyksessä tämä kannattaisi toteuttaa. Kävin läpi Huotarisen, Hovin, lähdemäen tekemän materiaalin käsiteanalyysistä ja -mallinnuksesta. Nämä diat palauttivat mieleen, miten saadaan suunniteltua hyvä tietokanta, sekä myös tietokantamerkistöjä ja riippuvuuksia.

Perjantai 19.04.2024

Perjantaina päiväni alkoi tietokannan suunnittelulla. Aloitin suunnittelun miettimällä, mitä tauluja tullaan tarvitsemaan tietokantaan, ja aloin toteuttamaan tätä paperin ja kynän avulla. Tietokannan suunnitteluun kului todella paljon aikaa, sillä yritin ymmärtää mitkä kaikki taulut tullaan tarvitse-

maan toimivan tietokannan suunniteluun ja miten taulujen väliset riippuvuudet tulisivat toimimaan. Heti suunnittelun alussa oli vaikea ymmärtää miten toteuttaa, kun tilauksella on kolme eri käyttäjää, miten näiden riippuvuudet saataisiin toteutettua järkevästi. Lisäsin erillisen välitaulun tilauksen ja käyttäjän väliin, missä valitaan käyttäjän rooli tilauksessa, tämän avulla voin erottaa tilauksessa olevan roolin ja käyttäjän. Päivä meni nopeasti ja sain jo vähän suunniteltua mitä tauluja ja riippuvuuksia sovelluksen tietokanta tulisi tarvitsemaan.

Viikon 4 yhteenveto

Tuntui kuin en olisi saanut mitään tällä viikolla aikaan, vaikka joka päivä teinkin töitä, mutta semmoista se taitaa välillä olla ohjelmistosuunnittelun parissa. Mutta kaiken kaikkiaan hyvä viikko takana, sain hyviä ideoita vastaavien sovellusten käyttöliittymistä, sain myös käyttöliittymäsuunnittelun valmiiksi ja pääsin jo vähän alkuun tietokanta suunnittelussa.

3.5 Seurantaviikko 5

Viidennen viikon tavoite on selkeä: suunnitella ja toteuttaa toimiva tietokanta.

Maanantai 22.04.2024

Päivä alkoi MySQL Workbenchin ja XAMPPin asennuksella. XAMPP on avoimen lähdekoodin työkalu, jolla saa omalle koneelleen pyörimään Apache http-serveriä ja MariaDB-tietokantaa ilman isompia konfigurointeja, ja näin ollen helpottaa lokaalia kehitystä. Asennusten jälkeen otin yhteyden Workbenchin ja MariaDB:n välillä käyttäen XAMPPia, jotta sain lokaalitetokannan toimimaan ja pääsisin luomaan fyysistä tietokantaa. Tänään selvittelin myös mikä tulisi olemaan ohjelmistoni myynnin ja markkinoinnin suunnitelma ja perehdyin myös siihen, miten on mahdollista laskea yrityksen kassavirtaa. Myynnin ja markkinoinnin suunnittelussa määrittelin ketkä tulisivat olemaan sovelluksen pääkäyttäjiä, mitä sovellus tarjoaa ja miten mahdollinen markkinointi tulisi tapahtumaan. Sovelluksen mahdollisista käyttäjistä sain luotua hyvän listan siitä, keitä tulisi kuulumaan tähän ryhmään. Sovelluksen tarjoamaan palveluun en saanut mitään jo ennestään esille tullutta asiaa lisättyä, kirjotin vain lyhyen kuvauksen siitä mitkä ovat sovelluksen tarjoamat hyödyt ja miten sovellus toimii. Markkinointiin sain varmuuden, että se tulisi tapahtumaan sosiaalisen median

kautta, kuten Facebook, Instagram ja muiden vastaavien alustojen avulla. Mahdollisten positiivisten palautteiden luoma julkisuus toimisi myös hyvänä markkinointi välineenä. Kassavirtalaskelmaan sain myös hyvän Excel-pohjan, jossa on valmiiksi luodut kaavat ja kentät mitä laskelma tulee tarvitsemaan.

Tiistai 23.04.2024

Aloitin päivän luomalla ER-mallia paperille käyttäen avuksi Figmalla luotua käyttöliittymäprototyyppiäni. ER-malliin tein käyttäjän, vastaanottajan, lähettäjän ja kuljettajan taulut. Tauluille annoin sarakkeita ja jokaiselle sarakkeelle määriteltiin tietotyyppi mitä se voisi sisältää. Taulujen sarakkeiden tarvitsemat tiedot sain Figmalla tehdyllä suunnitelmastani. Sen avulla pystyin tekemään valmiit taulut mitä kaikkea tietoa kyseinen taulu tulisi sisältämään. Seuraavaksi oli vuorossa taulujen välisten relaatioiden luominen, minkälainen relaatio olisi kyseessä ja olisiko relaatio pakollinen, jotta saadaan toinen taulu luotua. Pitkän taistelun jälkeen sain luotua neljän taulun relaatiotietokantamallinnuksen. Seuraavana olikin tarkoitus kääntää tämä MySQL Workbenchin ”export”-työkalun avulla toimivaksi tietokannaksi, jotta pääsisin testaamaan toimivatko luodut relaatiot taulujen välillä. Sain kyseisellä työkalulla loppujen lopuksi luotua tietokannan luomisen SQL-lauseet, mutta en saanut tällä ikinä toimimaan taulujen relaatiota, enkä koskaan saanut selville miksi tämä sovellus ei osaa kääntää relaatioita oikein.

Keskiviikko 24.04.2024

Eilisen taistelun jälkeen takaisin tietokantojen pariin. Kysyin IT-alalla ohjelmoijana työskentelevältä tutultani neuvoa eiliseen ongelmaani, ja hän ehdotti, että kokeilisin PostgreSQL:ää tietokantana ja DBeaveriä hallintaohjelmana, sillä hän oli kokenut ne omassa käytössään toimivammiksi. Päätin kokeilla vaihtaa tietokannan hallintajärjestelmän MySQL workbenchistä DBeaveriin ja MariaDB:n PostgreSQL:iin. PostgreSQL:n asensin Windows Subsystem for Linuxin (WSL) kautta, sillä tuttuni pelotteli minua Windowsiin asentamisen vaikeudesta – WSL:n avulla pääsen myös eroon XAMPPista. DBeaverin asennuksen jälkeen loin DBeaverin ja PostgreSQL:n tietokannan välisen yhteyden, jotta sain lokaalin tietokannan toimimaan ja jotta pääsen testaamaan luotuja tietokantoja. Tämän jälkeen aloitin tietokantojen luomisen. Eilisen suunnittelun seurauksena päätin aloittaa fyysisten tietokantojen toteutuksen käyttäen suoraan SQL-lauseita. Nopean SQL-kertauksen jälkeen loin neljä

eri tietokantataulua: käyttäjä, lähettäjä, vastaanottaja ja kuljettaja. Taulujen luomisen jälkeen oli-kin vuorossa näiden väliset yhteydet, ja loin ensimmäisen yhteydet käyttäjän ja tilauksen välille. Huomasinkin tämän jälkeen, että DBeaver luo ER-mallin luoduista tauluista ja näiden yhteyksistä. ER-malli auttaa havainnollistamaan jo luotua tietokantaa, ja DBeaverin malli näyttikin selkeäm-mältä kuin MySQL Workbenchin. Sainkin nyt luotua yhteydet neljän eri taulun välille.

Viikon 5 yhteenveto

Viikolla viisi sain hyvän kuvan sovellukseni asiakasryhmästä ja siitä, miten tulisin markkinoimaan sovellustani. Pääsin myös hyvin alkuun tietokannan luomisessa, vaikka yksi päivä menikin hukkaan, kun yritin luoda tietokantaa MySQL Workbenchin avulla. Ensimmäinen työkalu, tai tapa, mitä käy-tän, ei välttämättä ole se oikea, ja neuvoa on hyvä kysyä pienellä kynnyksellä asiantuntijalta, jos siihen on mahdollisuus. Huomasin myös, miten paljon koulussa opittua on unohtunut, joten SQL-kertaus oli aiheellinen. Mutta onneksi vaihdoin tietokannan hallintajärjestelmää, nyt on taas posi-tiivinen mieli, että kyllä tämä vielä onnistuu.

3.6 Seurantaviikko 6

Tavoitteena olisi jatkaa tietokannan suunnittelua ja testausta. Ajattelin myös tarkentaa kassavirta-laskelman ja tutustua eri sovellusrajapintoihin ja näiden hintoihin.

Maanantai 29.04.2024

Tänään oli vuorossa kassavirtalaskelmaa. Kassavirtalaskelman avulla sain selville, että sovellus voisi mahdollisesti olla tuottava, jos vaan saan käyttäjiä tarpeeksi sovellukselleni. Laskelmaan keräsin yhdeksän eri tuottomahdollisuutta ja jokaiselle näille oli oman hintansa. Lisäsin kaavioon myös mahdollisten ulkopuolisten sovellusten käytön ja yhden sovellussuunnittelijan palkan, jotta las-kelma olisi mahdollisimman todenmukainen. Laskin, että jos käyttäjämäärät kasvavat 1.4 % kuu-kausivauhtia, sovellukseni tulee tuottamaan tuottoa ensimmäisen vuoden jälkeen. Näiden tietojen avulla voin ruveta suunnitelmaan mitä muutoksia tai lisätoiminnallisuuksia sovellus voisi sisältää, jotta saataisiin kävijämäärää ja tuottoa lisättyä.

Perjantai 03.05.2024

Muutaman päivän tauon jälkeen pääsin vihdoinkin takaisin tietokannan luomiseen ja testaukseen. Päätin rupeavani testaamaan tauluja luomalla tietoa näihin SQL-lauseiden avulla. Aloitin luomalla kolme riviä tietoa käyttäjätauluun. Seuraavaksi oli vuorossa kuljettaja-, ja vastaanottajataulujen tietojen syöttäminen käyttämällä jo luodun käyttäjätaulun tietoa. Näiden tietojen jälkeen pääsin-kin syöttämään tilaustietoja, eli kuka on käyttäjä, vastaanottaja ja kuljettaja. Huomasin, että tietokanta on rakennettu toimivaksi ainakin tähän mennessä. Aloin testaamaan mahdollisia muutoksia tauluihin. Poistin, lisäsin ja muutin tauluissa olevia tietoja varmistaakseni kaiken mahdollisen toiminnan mitä vaaditaan jo luotujen taulujen kohdalla, ja kaikki näyttikin toimivan miten pitää. Tutustujien suosituksesta rupesin myös tutustumaan ChatGPT:n käyttöön, sanoivat että olisi hyvä apu perusongelmien ratkomisessa, kunhan muistan, etten syötä sille arkaluontoista tietoa, sillä kyseinen tekoäly oppii jatkuvasti samalla, kun sitä käyttää, ja muistaa kaiken sille kertomani tiedon. Koikeiltuani hetken aikaa ChatGPT:tä, huomasin että tämä kyllä auttaa perusasioiden ymmärtämisessä, kuten termien ja miten kannattaisi luoda peruskäyttäjät ja yhteydet. Vaikka opettelinkin tekoälyä käyttämään, huomaan että etsin edelleen ensimmäisenä vastauksia netistä ja jos käytän tekoälyä, pyrin vielä tarkistamaan vastauksen netistä luotettavista lähteistä. ChatGPT antoi nimittäin myös kaikenlaisia vääriä vastauksia, ja pyydetessä korjaamaan ehdotti samaa, mutta kiertäen.

Viikon 6 yhteenveto

Tällä viikolla sain selville, että omalla ohjelmistollani voisi pärjätä markkinoilla ja pystyisin jopa mahdollisesti elättämään itseni. Sain myös varmuuden, että olen menossa oikeaan suuntaan tietokannan rakentamisessa. Lyhyeksi jäi vain tämä viikko sairauden takia, mutta eipä sille aina voi mitään.

3.7 Seurantaviikko 7

Viikolla seitsemän tavoitteena olisi saada tietokanta tehtyä ja osittain testattua. Mahdollisesti jos aikaa jää, aloitan käyttäjätarinoiden luomisen ja selvittämisen mitä mahdollisia ulkopuolisia sovelluksia tulen tarvitsemaan ja mitä näiden käyttö tulee maksamaan.

Maanantai 06.05.2024

Tänään kävin läpi mahdollisia rahoitusvaihtoehtoja, joita voisi saada sovelluskehitykseen ja yrityksen perustamiseen. Rahoitusvaihtoehtoja oli monia. Yritystehdas ja JAMK tarjoavat pieniä rahoitusvaihtoehtoja, joiden avulla voisin maksaa omassa sovelluksessa käytettäviä toisien sovelluksen maksuja, kuten karttapalvelua. Isomman rahoituksen voisi tarjota Finnvera: tämä mahdollistaisi yhden henkilön palkkaamisen sovellussuunnitteluun tai markkinointiin, ja sovelluksessa käytettävien sovellusten tai sovellusrajapintojen maksujen kattamiseen. Toki starttirahakin on yksi vaihtoehto, mutta tämän avulla en pelkästään pärjäisi ja tekisin itse kaiken. Henkilökohtainen laina olisi paras vaihtoehto, mutta tässä on esteenä, että pitäisi olla vakituiset tulot ja aloittavaa yritystä ei lasketa tähän. Tulin lopputulokseen, että starttiraha, yritystehdas ja JAMK voisivat olla yhdessä yksi vaihtoehto tai toinen vaihtoehto olisi yksi iso laina Finnveralta. Tein myös kannattavuuslaskelman, joka auttoi ymmärtämään minkä verran pitäisi olla tuloja päivässä, jotta yritys olisi kannattava. Selvitin myös, että Googlen karttapalvelua ei kannata käyttää, se on todella kallis verrattuna esimerkiksi Radariin.

Tiistai 07.05.2024

Tänään sain luotua neljä uutta taulua tietokantaan, en ainakaan keksinyt mitä muita tauluja tietokanta voisi tarvita. Relatioiden suunnittelu oli välillä vähän haastavampaa. Relatiot suunnittelin niin, että tilaus on käyttäjän alla ja tilaukseen kerätään kaikki tarvittava tieto, jotta tiedon hakeminen on tulevaisuudessa helppoa tilausta tai tilaajaa käyttäen. Annoin myös kaikille tauluille sarake-tiedot, minkälaista tietoa nämä pitäisivät sisällään. Sarakkeisiin annoin arvoja kuten INT (Integer, numeroarvo), jota käytän taulun ID:n ja pääavaimen luomiseen. Taulun ID on myös uniikki ja luodaan automaattisesti. VARCHAR(10) (variable-length characters, sarake, joka sisältää 10 merkkiä) sarakkeeseen loin etunimelle tallennuspaikan, TIME ja DATE sarakkeet ovat ajan ja päivämäärän tallentamista varten, BOOL (totuusarvo, true tai false) esiintyy parissa taulussa ja tämä toimii kyllä/ei periaatteella. Yhdestä taulusta löytyy myös BYTEA (bytearray, tavujoukko, PostgreSQL tietotyyppi), joka on kuvan tallentamista varten luotu sarake. Kuvat tosin tulen tallentamaan jatkossa erilliseen palveluun. Tänään sainkin tietokannan luotua melkein valmiiksi, yhden taulun relatio ja tiedot pitävät vielä tarkistaa.

Keskiviikko 08.05.2024

Tietokannan toteutus jatkui arviointitaulun sarakkeiden ja relaatioiden luomisella. Lisäsin arviointitauluun tarvittavat sarakkeet, kuten toimituksen arviointi. Relaatiot arviointitauluun loin siten, että käyttäjällä voi olla monta arviota, mutta arviolla vain yksi käyttäjä ja tilauksella voi olla monta arviointia, mutta arvioinnilla vain yksi tilaus. Tämä mahdollistaa jokaisen käyttäjän arvioinnin tilauksen toteutumisesta. Lisäsin myös kaikkiin tauluihin sarakkeet rivin luomisen ja muokkauksen seurantaan. Rivien luomista ja muokkausta seurataan ajan ja päivän perusteella TIMESTAMPTZ:in (timestamp with time zone) avulla, joka täyttää taulun luomis- ja muokauspäivän automaattisesti. TIMESTAMPTZ on aikaleima, joka ottaa huomioon myös aikavyöhykkeet, jolloin sarakkeen tietoa luodessa huomioidaan kesä- ja talviaika, sekä karkauspäivä tai -vuosi, serverin maa-asetuksen mukaan.

Perjantai 09.05.2024

Tänään loin tietokantaani tietoa SQL-lauseita käyttäen, jotta pääsen testaamaan tietoa fyysisessä tietokannassa. Tiedot tauluihin syötin Insert Into, Select, From ja Where pääkomentoja käyttäen. Insert Into määrittelee mihin tauluun ja mitä rivejä täytetään, Selectilla valitaan sarakkeet taulusta, mihin tietoa syötetään, From-komentoa käytetään, kun tarvitaan tietoa muista tauluista ja Where-komennolla voidaan asettaa rajoja tiedolle. Aloitin luomalla käyttäjiä tietokantaan, käyttäjiä loin seitsemänkymmentä kappaletta. Seitsemänkymmenen rivin luontiin käytin From -lauseen sisällä olevaa SQL-lausetta `generate_series(1, 70);`, joka luo seitsemänkymmentä riviä tietoa tietokantaan. Käytin myös Selectin sisällä olevaa SQL-lausetta: `LEFT(md5(random)::text), 10) as firstname`, joka arpoo tässä tapauksessa etunimelle arvalla 10-merkkiä pitkän kirjain- ja numeroyhdistelmän. Tämän avulla sain kaikki rivit täyteen nopeasti, vaikkakin merkkijonot eivät tarkoita mitään. Tämän jälkeen pääsin luomaan lähettäjien tietoja. Lähettäjiä loin 23 kappaletta ja näiden luontiin käytin jo edellä luotujen käyttäjiä. Tämän avulla sain luotua nopeasti joka kolmannesta käyttäjästä lähettäjän käyttäen SQL-lausetta: `where tUsers.user_id % 3 = 1;`. Samat toiminnot tein luodessani vastaanottajat, mutta viimeisen SQL-lauseen toteutin `tUsers.user_id % 3 = 2;` komennolla, jotta lähettäjä ja vastaanottaja eivät voi olla samoja. Seuraavaksi oli paketin tietojen täydentäminen. Paketin koko arvottiin käyttäen PostgreSQL:n `random()`-funktioita, joka generoi sattumanvaraisia lukuja. Loin myös `current_timestamp()`-funktiolla paketin noutoajan ja toimitusajan

`current_timestamp() + interval '2 hours'`, lauseella. `Current_timestamp()`-funktio palauttaa tiedon luontiajan, ja toimitusajan luontilause mahdollisti toimitusajan 2 tuntia myöhemmäksi. Päivän viimeinen homma oli täyttää tilaustaulun tiedot. Valitettavasti en saanut vielä tätä toimimaan, en vielä keksinyt miten saan jokaisesta taulusta tuotua tietoa tilaustauluun.

Viikon 7 yhteenveto

Tällä viikolla sain hyvin vietyä projektiani eteenpäin. Nyt tiedän, mitä mahdollisia rahoitusvaihtoehtoja voisin saada. Selvisi myös, että kannattaa valita tarkkaan mitä ulkopuolisia sovelluksia tulisin käyttämään omassa sovelluksessani, koska näillä on huomattavia hintaeroja. Sain vihdoin tietokannan valmiiksi ja vähän tietoa tietokanta tauluihin, nyt vaan seuraavaksi lisää tietoa tauluihin ja testaamaan.

3.8 Seurantaviikko 8

Viimeisen viikon tavoitteena olisi saada tietokanta testattua. Tulee myös selvittää mitä ulkoisia sovelluksia ja rajapintoja tulen tarvitsemaan omassa sovelluksessa, ja kenen tarjoamana näitä olisi järkevin käyttää, niin hinnan kuin toimivuuden puolesta.

Maanantai 13.05.2024

Tiedon luominen tietokantaan jatkui. Tänään oli vuorossa tilaustaulun tiedon luominen. Tämä vei yllättävän paljon aikaa, että keksin miten saan tuotua tarvittavat tiedot tähän tauluun. Tulin siihen lopputulokseen, että tuon samoja tietoja tilaustauluun mitä pakettitaulussa käytetään ja luon alkutiedot taulussa näiden avulla. Seuraavaksi tilaustauluun piti päivittää muitakin tietoja, kuten missä vaiheessa tilaus on. Ongelmaa oli myös siinä, kun poistin tiedot tauluista, niin indexit alkoivat siitä mihin edelliset poistetut tiedot olivat jääneet. Tämä johtui siitä, että tietokanta säilyttää taulujen indeksit omassa sekvenssitaulussansa. Sain tämän ratkaistua SQL-lauseella `ALTER sequence joutain.orders_order_id_seq restart with 1;`, joka nolaa indeksin tilaustauluun, jolloin saan indeksit alkamaan taas ensimmäisestä.

Tarkastelin myös kassavirtalaskelmia ja selvitin, mitä mahdollisten ulkopuolisten rajapintakutsujen hinta voisi olla: tulin siihen tulokseen, että yksi tilaus maksaa keskimäärin noin 0,1 euroa. Selvitin myös mistä saan apua, kun sovellus tulee olemaan melkein valmis tai jos tarvitsen apua sovelluksen rakentamiseen. Löysin parit mahdolliset yritysvalmennusyritykset, jotka auttavat aloittavaa yritystä kädestä pitäen tarvittaessa. Selvitin myös, miten minun tulisi valmistautua mahdollisiin haastatteluihin, kuten sovelluksen esitykseen ja talouslaskelmiin.

Tiistai 14.05.2024

Tänään muokkasin tietoja tauluissa ja testailin toimivatko taulujen väliset relaatiot oikein. Huomasin testausvaiheessa, että saan tuotua ei-toivottua tietoa toisesta taulusta toiseen. Tieto, jota toin toisesta taulusta, sisälsi aktiivisuussarakkeen, joka on boolean (kyllä tai ei). En halunnut mahdollistaa tiedon siirtoa toiseen tauluun, jos tämä sarake on ei, eli käyttäjä ei ole aktiivinen. Tämän estämiseksi käytin trigger-funktiota, joka käynnistyy aina, kun tauluun luodaan uutta tietoa, jos tieto on väärää. Tässä tapauksessa trigger-funktio antaa ilmoituksen, että boolean arvo on väärä, eikä tietoa voida luoda. Huomasin myös, että minulla on yksi ylimääräinen taulu, joka voidaan järkevämmiin toteuttaa toisessa taulussa. Jouduin poistamaan yhden taulun, jonka sarakkeet siirsin jo olemassa olevaan kuljettajatauluun, ja osan kuljettajataulun sarakkeista siirsin käyttäjätauluun. Näiden muutosten seurauksena pääsin muokkaamaan tietojen luomista tauluihin ja testaamaan taulujen toimivuutta.

Keskiviikko 15.05.2024

Tämä päivä alkoi siitä mihin edellinen jäi: uusiin tauluihin luodaan uutta tietoa ja muokataan toisesta tauluista periytyviä tietoja siten, että nämä toimivat keskenään. Tuli myös ongelmia siitä, miten luon arvalla bytea-tietoa sarakkeeseen, johon on tarkoitus tallentaa kuva. Sain tämän ratkaistua PostgreSQL:n lisäosalla pqcrypto avulla. Pqcryptoa käytetään myös salasanojen salaamiseen. Kaiken taistelun jälkeen olen saanut vihdoinkin luotua tauluihin tietoa automaattisesti, eikä minun tarvitse jokaista tietoa erikseen käsin syöttää. Seuraavaksi pääsin suunnittelemaan sitä, miten olisi järkevä hakea tarvittavaa tietoa tauluista, ettei sovelluksen käyttäjä saisi sille kuulumatonta tietoa käyttöönsä. Pääsin jo hyvin alkuun tässä, eli saan haettua tilaukseen kuuluvat tiedot eri tauluista, ja käyttäjälle mitä tilauksia käyttäjällä mahdollisesti on.

Torstai 16.05.2024

Tänään rupesin tekemään näkymiä. Näkymät luodaan tietokantaan, jotta voitaisiin hakea vain tarvittava tieto esimerkiksi käyttäjälle, eikä käyttäjä mahdollisesti saisi sille kuulumatonta tietoa käsiinsä. Näkymän luomiseksi pitää tehdä SQL-lause, joka tallennetaan tietokantaan näkymää kuvaavalla nimellä ja jota on jatkossa helppo käyttää yhdellä SQL-rivillä käyttämällä pelkästään näkymän nimeä. Sain jo jokusen näkymän luotua, kuten käyttäjän tiedot, tilauksen tiedot ja eriteltyinä vastaajan ja lähettäjän tiedot. Näkymiä tehdessäni huomasin myös, että tarvitsen osaan tauluista lisää sarakkeita, jotta tiedon hakemisesta ja taulujen käytöstä tulee entistä järjestelmällisempää. Muutokset tauluihin oli pääasiassa sellaisia, onko kyseisen taulun rivi aktiivinen vai ei. Muutokset olivat pakollisia vaikkakin työläitä. Piti tehdä muutos taulun sarakkeeseen ja mahdolliset relaatiot sarakkeesta mitä voidaan tarvita toisissa tauluissa. Tämän jälkeen syötetään taas tiedot, muokataan tietoja ja testataan tauluja. Päivä menikin melkein taulujen muutoksien ja testaamisen parissa, mutta tästä on hyvä jatkaa.

Perjantai 17.05.2024

Viimeinen päivä työn seurantaan. Järjestelin DBeaverissa SQL-lauseet omiin tiedostoihin, jotta lauseiden ajaminen on jatkossa selkeämpää. Tiedostot ovat: luodaan taulut, lisätään tietoja tauluihin, muokataan tietoja, triggerit ja luodaan näkymät. Kävin myös läpi, mitä olin saanut tehtyä tämän kahdeksan viikon aikana ja tein listan, miten jatkan sovelluksen suunnittelua ja toteutusta tulevaisuudessa. Sovelluksen suunnittelu ja toteutus jatkuu tietokannan loppuun suunnittelulla. Tämän jälkeen on vuorossa sovelluksen ohjelmointi, jonka jälkeen tehdään testausta ennen lopullista julkaisua.

Viikon 8 yhteenveto

Viimeinen viikko sujui hyvin, sain tehtyä tietokannan melkein valmiiksi. Kirjasinkin mitä muutoksia tietokantaan pitää tehdä ja mitä mahdollisia näkymiä pitää luoda. Huomasin myös, että tietokannan tekeminen ja suunnitteleminen onkin työläämpää kuin mitä olin luullut, mutta paljon opin tietokannan tekemisestä ja näkymien luonnista ja tästä onkin todella hyvä jatkaa projektia eteenpäin.

4 Pohdinta

4.1 Tavoitteet ja tulokset

Opinnäytetyön tavoitteena oli seurata sovellussuunnittelua oikeassa projektissa. Sovellus tulee toimimaan osana yritystä tulevaisuudessa. Seuranta toteutettiin päiväkirjaa kirjoittamalla kahdeksan viikon ajan: päiväkirjaan oli tavoitteena kirjoittaa mahdollisimman tarkka kuvaus siitä, mitä työpäivän aikana oli tehty, mahdolliset ongelmat ja niiden ratkaisut. Tulosten tavoitteena oli saada selkeä kuva siitä, mitä projektin suunnitteluun tarvitaan, niin suunnittelun, ohjelmoinnin ja yrityksen perustamisen näkökulmasta.

Sovellussuunnittelun seurannan avulla voitiin hahmottaa, mitä oikean projektin suunnittelu vaatii ja miten sen tekeminen voi olla joskus työlästä ja jopa tuloksetonta. Osa päivistä olikin semmoisia, että vaikka päivän päätteeksi tuntui, ettei ollut saanut mitään konkreettista aikaan, ajatustyötä tuli tehtyä sitäkin enemmän: tämä oli suunnitteluvaiheessa hyvä, sillä tätä kautta tulikin uusia ideoita ja ajatuksia. Asiat, joita ei saanut ratkottua päivän aikana jäivät myös pyörimään mieleen, ja monesti saattoikin käydä niin, että ylitsepääsemätön ongelma olikin aamulla päivänselvä. Seurannassa huomattiin myös, että itselläni sovellussuunnittelijana ei ollut paljon kokemusta sovellussuunnittelusta ja monesti aikaa meni käytettyjen teknologioiden opiskeluun ja kertaamiseen. Seuranta tehdessä huomasin, miten aliarvion tulevan viikon tavoitteiden työn määrän. Esimerkiksi tietokantojen suunnitteluun oli varattu viikko, mutta todellisuudessa suunnitteluun tarvittiinkin vielä lisäviikko. Seurannan aikana selvisi myös, mitä kaikkea uuden yrityksen perustamiseen vaaditaan: onko sovelluksella kysyntää, mistä saada rahoitusta yrityksen perustamiseen, mitä tarvitaan, jotta yritys voisi saada rahoitusta ja mitä mahdollisia dokumentteja ja laskelmia pitäisi tehdä, joilla voidaan varmistaa, että yritys olisi kannattava.

4.2 Sovellussuunnittelun onnistuminen ja kehityskohteet

Sovellussuunnittelu alkoi GitLab-ympäristöstä. GitLabiin onnistuttiin keräämään yleistä tietoa sovelluksen toiminnasta ja aikataulutusta sovellussuunnitteluun. Projektin edetessä GitLab-ympäristön päivittäminen ja seuraaminen unohtui, ja tähän pitäisikin panostaa jatkossa, kun projekti etenee ja kasvaa, muuten projektin seuraaminen ja aikatauluttaminen on hankalaa.

Seuraavana oli vuorossa sovelluksen käyttöliittymän suunnittelu Figman avulla. Figma osoittautui todella tehokkaaksi työkaluksi, jonka avulla voitiin suunnitella tarkka kuvaus sovelluksesta ja sen toiminnoista. Figman avulla sain suunniteltua sovelluksen käyttöliittymästä todella tarkan kuvauksen, miten se tulisi toimimaan ja mitä kaikkea mahdollinen sovellus voisi tulla tarvitsemaan. Sovelluksen käyttöliittymä suunnittelu vei todella paljon aikaa, mutta tämä tulee varmasti maksamaan itsensä takaisin, kun sovellusta ruvetaan toteuttamaan.

Tämän jälkeen oli vuorossa tietokantasuunnittelu. Tietokannan suunnittelun apuna käytin käyttöliittymä suunnitelmaa, jonka avulla sainkin jo hyvän kuvan miltä mahdollinen tietokanta tulisi näyttämään – tässä näkyi jo suoraan Figmalla suunnitellun prototyypin hyödyllisyys. Aloitin tietokantasuunnittelun loogisella mallinnuksella, kynää ja paperia käyttäen, mutta huomasin että en pääse suunnitelmassani eteenpäin tällä keinolla, joten päätin siirtyä suoraan fyysisen tietokannan toteutukseen. Tämä valinta osoittautui minulle sopivammaksi vaihtoehdoksi: tietokannan fyysistä mallia toteuttaessa pääsin suoraan näkemään tietokantajärjestelmän avulla myös loogisen ER-mallin. Tietokannan toteutus onnistuikin hyvin ja sain alustavan tietokannan suunniteltua. Tietokantaa suunnitellessani jouduin myös muuttamaan käyttöliittymäsuunnitelmaani Figmassa, jotta tämä pysyisi ajan tasalla eikä jatkossa tulisi ongelmia sovelluksen toteutuksen kanssa.

Mitä itse yrityksen perustamiseen tulee, sain sovellussuunnittelua tehdessä paljon uutta tietoa siitä, miten muiden vastaavat sovellukset toimivat ja minkälaisia maksuja yritykset pyytävät palveluista. Selvitin myös mahdollisesti sovelluksessa käytettävien ulkopuolisten sovelluksien ja rajapintojen hinnastoja, ja mitä näiden käyttäminen tulisi maksamaan. Onnistuin myös laskemaan sovellukseni mahdollisen kannattavuuslaskelman, josta jäikin positiivinen kuva, että sovelluksella voisi olla tulevaisuutta markkinoilla.

Yleisesti sovellussuunnittelu saatiin todella hyvälle mallille, ja tästä on mukava jatkaa kohti sovelluksen toteutusta. Seuraavana olisi vuorossa sovelluksen ensimmäisen prototyypin ohjelmointi, jonka jälkeen päästään testaamaan sovellusta ensin virtuaalisesti ja tämän jälkeen ihan oikeiden käyttäjien avulla.

4.3 Tulosten luotettavuus ja käytettävyys

Päiväkirjaa kirjoittaessa en valitettavasti voinut tuoda kaikkea tietoa esille mitä sovellukseen oli suunniteltu ja toteutettu. Sovelluksen on tarkoitus olla osana yrityksen toimintaa, joten sovelluksen osa toiminnoista on salattua ja näiden salattujen toimintojen avulla pyrin erottumaan kilpailijoistani. Tämä vaikeuttikin paikka paikoin päiväkirjan kirjoittamista. Toinen huono puoli salatuissa toiminnoissa on, että päiväkirja ei anna ihan oikeaa kuvaa, miten työtä on tehty ja miten aikataulutus on pitänyt paikkaansa, koska salattujen toimintojen suunnittelemiseen on mennyt jokin aika seurannasta.

Mitä hyötyä sitten tuloksista on? Seuranta antaa hyvän käsityksen, miten sovelluksen suunnittelu kannattaa aloittaa ja miten siihen kannattaa varata aikaa. Seuranta osoittikin, että sovellussuunnittelu seurantaympäristön pystyttämistä tietokantasuunnitteluun vie arviolta seitsemän viikkoa: tämä tietenkin johtuu sovelluksen laajuudesta, monta henkilöä projektissa on ja miten tarkkaan suunnitelma halutaan toteuttaa. Seurannan aikana huomattiin myös, että ensimmäinen suunnitelussa käytettävä sovellus ei aina välttämättä ole se oikea. Ja vaikka kuinka tarkka suunnitelma sovelluksesta on rakennettu, niin se tulee muuttumaan vielä seuraavaa vaihetta tehdessä. Esimerkiksi tässä tapauksessa käyttöliittymä oli suunniteltu todella huolella, mutta tietokantaa tehdessä käyttöliittymän suunnitelmaa piti välillä muuttaa. Tässä huomataan ketterän kehityksen etu verrattuna vesiputousmalliin: ketterää kehitystä käyttämällä voidaan palata edellä toteutettuun vaiheeseen helposti, kun taas vesiputousmallia käyttämällä tämä olisi paljon työläämpää.

4.4 Kehittyminen suunnittelijana

Sovelluksen suunnittelun ja toteutuksen aikana sain paljon uutta tietoa, miten sovellussuunnittelu tulisi tehdä ja mitä sovelluksia kannattaa käyttää sovellussuunnitteluun. Käyttöliittymä suunnittelussa käyttämäni Figma oli täysin uusi sovellus itselleni, ja mielestäni Figman käyttämisen oppiminen oli tärkeää, koska tulen tarvitsemaan Figmaa tulevaisuudessa, niin omassa sovelluksessa ja mahdollisesti myös työelämässä. Mitä tietokantoihin tulee, opin rakentamaan toimivan tietokannan sovellukselle. Tietokanta varmasti tulee vielä muuttumaan, kun sovellusta ruvetaan ohjelmoidaan ja tietokannasta haetaan tietoa. Ymmärsin myös sen, että itselleni on helpompi aloittaa fyysisen tietokannan toteutuksella, eikä loogisella mallilla; tähän varmasti vaikuttaa se että,

sovelluksen käyttöliittymä oli huolella suunniteltu ja tämän avulla tietokannan luominen oli selkeää. Huomasin myös, että opinnäytetyötä kirjoittaessa olen kehittynyt paljon itse kirjoittamisen tasolla, kuten pilkut, yhdyssanat ja lauserakenteet ovat alkaneet tulla jo luonnostaan, tästä olenkin todella tyytyväinen. Sovelluksen ja opinnäytetyön tekeminen käsikädessä osoittautui hyväksi valinnaksi, sain dokumentoitua mitä olen tehnyt sovellukseen ja voin tarvittaessa palata näihin, myös sovellus itsessään edistyi huomattavasti ja tästä onkin hyvä jatkaa sovelluksen toteutusta tulevaisuudessa.

Lähteet

Aaltonen, J. 2023. TIETOKANNOISTA YKSINKERTAISESTI JOHDOLLE. Viitattu 5.5.2024. <https://www.salescommunications.fi/blog/tietokannoista-yksinkertaisesti-johdolle>.

Agile / Ketterä kehitys - mitä ja miksi?. 2024. Viitattu 1.4.2024. <https://skillwell.fi/fi/blogi/kettera-kehitys-lyhyesti>.

Always start with an issue. 2016. Viitattu 10.4.2024. <https://about.gitlab.com/blog/2016/03/03/start-with-an-issue/>.

Basic writing and formatting syntax. 2024. Viitattu 17.4.2024. <https://docs.github.com/en/get-started/writing-on-github/getting-started-with-writing-and-formatting-on-github/basic-writing-and-formatting-syntax>.

Flowcharts - Basic Syntax. 2024. Viitattu 17.4.2024. <https://mermaid.js.org/syntax/flowchart.html>.

GitLab Flavored Markdown (GLFM). 2024. Viitattu 23.5.2024. <https://docs.gitlab.com/ee/user/markdown.html#diagrams-and-flowcharts>.

Gruber, J. 2004. Markdown. Viitattu 17.4.2024. <https://daringfireball.net/projects/markdown/>.

Ifloadus. Markdown. Viitattu 17.4.2024. <https://loadfocus.com/fi-fi/glossary/markdown>.

Issues. 2024. Viitattu 10.4.2024. <https://docs.gitlab.com/ee/user/project/issues/>.

Kaukojarju, A. 2017. Gitlab-järjestelmän versionhallintatyökalut tietojenkäsittelyn opetuksessa. Viitattu 10.4.2024. https://www.theseus.fi/bitstream/handle/10024/129339/Kaukojarju_Antti.pdf?sequence=1&isAllowed=y.

Kilpeläinen, A. Relaatiotietokanta. Viitattu 5.5.2024. <https://homes.jamk.fi/~kivni/http0140/material/relaatiotietokanta.html>.

Labels. 2024. Viitattu 18.4.2024. <https://docs.gitlab.com/ee/user/project/labels.html>.

Luotio, J. 2023. Ketterä kehittäminen – ohjelmistokehityksen moderni menetelmä. Viitattu 31.3.2024. <https://blog.innofactor.com/fi/kettera-kehittaminen>.

Luukkainen, M. 2022. Ohjelmistotuotanto, ketteruus, Scrum. Viitattu 1.4.2024. <https://ohjelmistotuotanto-hy-avoin.github.io/osa1/>.

Mitä tietokanta tarkoittaa?. Viitattu 5.5.2024. <https://www.oracle.com/fi/database/what-is-database/>.

Peterson, R. 2024. Tietokannan suunnittelu DBMS-opetusohjelmassa: Opi tietojen mallintamista. Viitattu 5.5.2024. <https://www.guru99.com/fi/database-design.html>.

Piispanen, T. 2021. Figma - Paras työkalu käyttöliittymien suunnitteluun. Viitattu 22.5.2024.
<https://www.webguru.fi/artikkelit/figma/>.

Virtanen, P. 2022. Käyttäjätarinoilla mitattavia tuloksia. Viitattu 10.4.2024.
<https://www.tieturi.fi/blogi/kayttajatarinoilla-mitattavia-tuloksia/>.