



Verkkosovelluksen kehitys Bubble-ohjelmistolla

Joonas Sipilä

OPINNÄYTETYÖ
Tammikuu 2024

Tietojenkäsittely
Ohjelmistotuotanto

TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Tietojenkäsittely
Ohjelmistotuotanto

SIPILÄ, JOONAS:
Verkkosovelluksen kehitys Bubble-ohjelmistolla

Opinnäytetyö 41 sivua, joista liitteitä 1 sivua
Tammikuu 2024

Tämän opinnäytetyön aiheena on mobiililaitteille tarkoitettun verkkosovelluksen suunnittelu ja kehitys Bubble.io:n koodittomalla, eli no-code pohjaisella alustalla. Työssä tutkittiin ja verrattiin Bubblien kehitysalustaa muihin no-code- ja full-code-pohjaisiin vastaaviin työkaluihin. Tavoitteena oli saada aikaan valmis verkkosovellus sekä perehtyä työn eri vaiheisiin aina suunnittelusta julkaisuun, mukaan lukien helppokäyttöisyys sekä grafiikka. Verkkosovelluksen nimi on Porkkanapankki, ja työn toimeksiantajana toimi Cirous Oy.

Opinnäytetyöprosessi alkoi Bubblien ohjelmiston opiskelulla. Opitun tiedon pohjalta rakennettiin mobiililaitteille optimoitu verkkosovellus. Kehityksessä käytettiin Bubble.io-kehitysalustan ja siihen integroitujen lisäosien lisäksi jonkin verran koodikieltä, sekä muun muassa Figma-työkalua sivun logiikan ja ulkoasun suunnitteluun. Käyttäjät testasivat sovellusta koko kehitysprosessin ajan helppokäyttöisyyden ja käyttömukavuuden varmistamiseksi.

Työn tuloksena valmistui Porkkanapankki-verkkosovellus, jota yritykset, seurat tai ryhmät voivat käyttää hyödyllisten, varsinaiseen työnkuvaan kuulumattomien tehtävien raportoimiseen ja ahkeruudesta palkitsemiseen. Työ täyttää sille asetetut oletukset, ja lähes kaikki työn tilaajan toivomat ominaisuudet toteutuivat. Käyttäjätestauksessa saatujen tulosten ja avoimen palautteen perusteella sovellus palvelee hyvin tarkoitustaan.

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Business Information Systems
Software Production

SIPILÄ, JOONAS:
Web Software Development Using Bubble-platform

Bachelor's thesis, 41 pages, appendices 1 pages
January 2024

The topic of this thesis is the design and development of a web application primarily designed for mobile devices. Bubble.io's no-code platform was used to create the web application. The study involved exploring and comparing Bubble's development platform with other no-code and full-code tools. The goal was to create a functional web application and to gain insights into various stages of the process, from planning to deployment, including considerations for user-friendliness and graphics. The thesis was commissioned by Cirous Oy and the application is called Carrot Bank.

The thesis was implemented by using the Bubble software to build a mobile-optimized web application. The development consisted of using the Bubble.io development platform, integrated add-ons, and some code language. During development, tools such as Figma were used for logic and interface design. Continuous user testing was conducted throughout development to ensure user-friendliness and usability.

As a result, Carrot Bank application was created for the need to report non-job-related tasks and reward diligence for businesses, associations, or groups. The work met the specified expectations and almost all the features requested by the client were successfully implemented. Based on the results of user testing and open feedback, the application effectively serves its intended purpose.

Key words: bubble.io, development platforms, web development

SISÄLLYS

1	JOHDANTO	6
2	TARKOITUS JA TAVOITE	8
	2.1 Työn tarkoitus ja tavoite	8
	2.2 Työnantajan näkökulma	9
	2.3 Työntekijän näkökulma	9
3	TEKNOLOGIAT	10
	3.1 Web-kehitys yleisesti.....	10
	3.2 No-code vs. Full-code	11
	3.3 Bubble.io	13
	3.3.1 UI näkymä	14
	3.3.2 Workflow sivu	16
	3.3.3 Tietokanta sivu	17
	3.3.4 Muut ominaisuudet	18
	3.4 Bublben lisäosat ja koodikieli	20
	3.5 Tekoälyn hyödyntäminen sovelluskehityksessä yleisesti	22
	3.6 Yhteenveto Bubblesta alustana.....	23
4	TYÖN TOTEUTUS.....	24
	4.1 Tuotokseen painottuva opinnäytetyö.....	24
	4.2 Suunnittelu	24
	4.3 Toteutus	25
	4.4 Grafiikka ja ulkoasu.....	26
	4.4.1 Värimaailma ja esteettömyys.....	26
	4.4.2 Elementtien asettelu ja intuitiivisuus	28
	4.5 Käyttäjätestaus ja julkaisu	31
	4.5.1 Käytetyt käyttäjätestauksen muodot.....	31
	4.5.2 Testauksen aikana käytetyt menetelmät	32
5	POHDINTA	35
	5.1 Tulokset	35
	5.2 Yhteenveto.....	36
	5.3 Mahdollisia tulevaisuuden toimia.....	37
	LÄHTEET	38
	LIITTEET	41
	Liite 1. Käyttäjätestauksen tehtävät	41

LYHENTEET JA TERMIT

responsiivinen	Eri kokoisille ja mallisille näytöille skaalautuva, käyttäjän toimintoihin reagoiva
API	Application Programming Interface. API:n kautta voi hakea tietoja kolmansien osapuolien sivustoilta
workflow	Työnkulku. Voi määrittellä sivun logiikkaa
state	Toiminto, jolla käyttäjän muuttuja voidaan tallentaa tietokantaan
Rest-rajapinta	Ohjelmointirajapintojen toteuttamiseen tarkoitettu malli
front end	Käyttäjälle näkyvä graafinen käyttöliittymä
back end	Palvelimella toimiva tietokantaohjelmisto
No-code	Kooditon
Low-code	Matalakoodinen
Full-code	Koodillinen
UX	User Experience

1 JOHDANTO

Verkkokehityksen maailma on jatkuvassa mullistuksessa uusien menetelmien ja mallien ansiosta. Verkkosivustojen luomisessa koodittomat alustat ovat antaneet uuden lähestymistavan perinteisten koodipohjaisten alustojen vaihtoehdoksi. Nämä alustat antavat niin yksilöille, kuin yrityksillekin mahdollisuuden luoda verkkosovelluksia ilman syvällistä koodausosaamista. (The Rise of No-Code Platforms: Will Web Development Soon Be a Thing of the Past, 2023) Yksi suosioita kasvattava alusta koodittomien alusten joukossa on Bubble.io. Bubble erottuu tehokkaana ja monimuotoisena työkaluna verkkosovellusten rakentamisessa sen helppokäyttöisyyden, skaalautuvuuden ja tehokkuuden takia perinteisempiin koodipohjaisiin malleihin verrattuna. Bubble alustan käyttö voi kehitettävästä sovelluksesta riippuen aiheuttaa joitakin haasteita, joista opinnäytetyössä kerrotaan myöhemmin.

Tässä opinnäytetyössä käsitellään Bubblea, sen ominaisuuksia sekä sen käyttöä alustana verkkosovelluksen luomisessa. Opinnäytetyössä käsitellään myös verkkosovelluskehityksen eri vaiheita omina kokonaisuuksinaan. Lisäksi tarkastellaan koodittomien alustojen eroja perinteisempiin koodipohjaisiin alustoihin verrattuna sekä sivutaan web kehitystä yleisesti.

Opinnäytetyön tuloksena valmistui Porkkanapankki niminen verkkosovellus, joka on pääasiassa tarkoitettu käytettäväksi puhelimella, mutta joka skaalautuu myös muille laitteille, kuten tietokoneille ja tableteille. Porkkanapankissa ylläpitäjä, esimerkiksi yrityksen esihenkilö, luo ryhmän ja kutsuu siihen työntekijät sähköpostitse. Ylläpitäjä luo ryhmään tehtäviä, kuten: siivoa toimisto, tyhjennä tiskikone tai turvallisuushavainto. Ylläpitäjä lisää samoin myös palkintoja, kuten: leffalippu, suklaalevy tai viinipullo. Näille tehtävillä ja palkinnoille päätetään ns. porkkanamäärät, eli kuinka monta pistettä kunkin tehtävän suorittamisesta saa ja kuinka monta pistettä palkintoon kuluu. Työntekijät keräävät sovelluksessa pisteitä eli porkkanoita tekemällä työnantajan lisäämiä tehtäviä. Keräämiään pisteitä he voivat vaihtaa itseään motivoiviin palkintoihin. Ylläpitäjä voi hyväksyä tai hylätä myös tämän pyynnön ja hyväksyessään toimittaa palkinnon valitsemallaan tavalla.

Sovellus toimii toistaiseksi suomen ja englannin kielillä. Sovelluksessa on kaksi väriteemaa: vaalea ja tumma. Ylläpitäjälle on luotu oma näkymä, jossa hän näkee ryhmän kaikki jäsenet, heidän porkkanamääränsä ja jossa hän voi tarkastella työntekijöitä ja heidän tehtävähistoriaansa yksittäin. Käyttäjänäkymässä käyttäjä pääsee käsiksi omaan profiiliinsa, ryhmäluetteloon, tehtävä- ja palkintoluetteloon sekä hän näkee oman porkkanamääränsä lisäksi ryhmän yhteenlasketun porkkanamäärän, mutta ei näe yksittäisten henkilöiden osuutta siihen.

Sovellusta ei lähdetty luomaan täysin tyhjästä. Työllä oli valmis runko ja muutamia ominaisuuksia, joiden pohjalta työtä lähdettiin jatkojalostamaan. Lopulliseen versioon on lisätty useita uusia ominaisuuksia, sivuista on tehty eri laitteille skaalautuvat ja sivujen määrää on pyritty vähentämään ja siirtymään mahdollisimman paljon single-pageksi, eli välttämään uusien sivujen lataamista sovellusta navigoidessa. Sovelluksen ulkoasu koki täyden muodonmuutoksen opinnäytetyöprojektin aikana ja sen käyttämään tietokantaankin tehtiin joitakin muutoksia ja lisäyksiä.

2 TARKOITUS JA TAVOITE

2.1 Työn tarkoitus ja tavoite

Porkkanapankki -sovelluksella pyritään parantamaan työntekijän mahdollisuuksia raportoida tekemiään hyödyllisiä, varsinaiseen työntekijän työnkuvaan kuulumattomia toimia työpaikalla, sekä antaa työnantajalle mahdollisuus tarkastella työpaikalla tehtyjä toimia ja palkita työntekijöitä ahkeruudesta ja oma-aloitteisuudesta. Sovelluksen käyttöönotolla voidaan pyrkiä parantamaan työmoraalia, lisäämään työntekijöiden oma-aloitteisuutta sekä helpottamaan havaittujen epäkohtien tai puutteiden raportointia. Käyttöönotolla pyritään myös lisäämään tyytyväisyyttä ottamalla työntekijöiden oma-aloitteisuus ja ahkeruus huomioon palkitsemalla heidät.

Vaikka sovellus onkin pääasiassa suunnattu yritysten käyttöön, on sovellus mahdollista ottaa käyttöön myös muissa ympäristöissä, kuten ryhmissä tai seuroissa esimerkiksi pitämään kirjaa kehityksestä ja saavutetuista tuloksista tai perheen sisäisesti esimerkiksi palkitsemaan perheen lapsia tehdyistä kotitöistä.

Opinnäytetyön tarkoituksena on kehittää verkkosovellusta Bubble.io nimisellä koodittomalla alustalla. Bubblien kooditon, eli no-code alusta eroaa perinteisemmistä koodipohjaisista full-code alustoista perustumalla visuaaliseen koodaamiseen perinteisten koodikielien sijaan. Sovellusta kehitettäessä on tarkoitus kiinnittää huomiota no-code alustan hyviin ja huonoihin puoliin perinteisempiin koodipohjaisiin alustoihin verrattuna, sekä tutkia mobiilikehityksen ja verkkosovelluskehityksen eri vaiheita.

Kehityksen aikana pyritään huomioimaan myös erilaisia graafisia sääntöjä ja ominaisuuksia kuten esteettömyyttä, värien yhteensopivuutta sekä elementtien asettelua. Kehityksen aikana suoritetaan myös käyttäjätestausta sovelluksen helppokäyttöisyyden varmistamiseksi. Näitä, sekä suunnittelua ja logiikan rakentamista tullaan opinnäytetyössä käsittelemään omina kokonaisuuksinaan.

Porkkanapankki -sovellusta on tarkoitus kehittää siten, että sitä voi käyttää luki- kuisilla eri laitteilla, kuten puhelimilla, tableteilla ja tietokoneilla. Sovellus kehitetään toimimaan sekä suomen, että englannin kielellä ja se pyritään kehittämään mahdollisimman kevyeksi, helpoksi ja nopeaksi käyttää.

2.2 Työnantajan näkökulma

Työnantajan tulisi hyötyä sovelluksen käyttöönotosta työpaikalla lisääntyneenä työntekijöiden oma-aloitteisuutena tehdä sivutoimia, kuten siivoamista tai vikojen korjausta. Näin työnantaja voi säästää muun muassa siivous- ja korjausku- luissa. Työnantaja pystyy sovelluksen avulla tarkastelemaan työpaikalla tehtyjä toimia isona kokonaisuutena, sekä yksittäisten työntekijöiden toimia erikseen. Näiden tulosten pohjalta työnantaja voi havaita mahdollisia puutteita ja/tai on- gelmia työpaikalla ja puuttua näihin hyvissä ajoin.

2.3 Työntekijän näkökulma

Työntekijän tulisi hyötyä ahkeruudestaan ja oma-aloitteisuudestaan ansaitse- malla työnantajan sovellukseen asettamia palkintoja. Työntekijä voi lunastaa haluamansa palkinnon milloin tahansa ja ansaita tämän heti työnantajan hyväk- syttyä tapahtuman. Sovelluksen käyttöönoton tulisi lisätä tyytyväisyyttä työtä kohtaan ja näin kasvattaa yrityksen sisäistä työhyvinvointia. Parhaassa tapauk- sessa sovelluksen avulla työntekijä voi nauttia enemmän työn ohessa tapahtu- vista sivutoimista tietäen hyötyvänsä niistä erilaisten palkintojen muodossa.

3 TEKNOLOGIAT

Sovelluksen teknisessä toteutuksessa käytettiin Bubble.io nimistä visuaalista ohjelmointisovellusta sekä pienissä määrin HTML-, CSS-, ja JavaScript-koodikieliä. Grafiikan ja väripaletin luontiin käytettiin apuna Figma ja Canva -sovelluksia. Sovellukseen on lisätty manuaalisesti useita erilaisia lisäosia Bublben omasta kirjastosta. Bubble on visuaalinen ohjelmointikieli, kehysalusta sekä palvelu, jonka avulla voidaan rakentaa ohjelmistoja ja tietokantoja ilman perinteistä koodikieltä. Sovellus toimii loistavana vaihtoehtona henkilöille, jotka eivät osaa koodata tai haluavat luoda nopeasti yksinkertaisen sivuston tai sovelluksen. Visuaalisella koodikielellä yksinkertaisten sovellusten luominen on nopeampaa kuin perinteisemmällä kielillä kuten JavaScriptillä, C++:lla tai Pythonilla. Kuitenkin sovelluksen koon kasvaessa Bubblea käytettäessä havaittiin joitakin ongelmia, joita perinteisemmällä kielillä tulee harvemmin vastaan. Näistä ongelmista mainitaan opinnäytetyössä myöhemmin

3.1 Web-kehitys yleisesti

Verkkokehityksen alkuvaiheessa 1980-luvulta 2000-luvun alkuun Internet oli suhteellisen uusi käsite. Verkkosivustot olivat yksinkertaisia ja painopiste oli ensisijaisesti tekstipohjaisessa sisällössä. Tänä aikana verkkokehitys oli kapeamuotoista ja kehittäjät tyytyivät manuaaliseen koodaukseen ja minimaalisiin työkaluihin. ”Siitä lähtien, kun internet tuli kaikkien saatavilla 1990-luvulla, siitä on tullut yhteiskunnan peruselementti. Verkkosivun omistaminen voi tehdä ihmeitä tavoitteesi saavuttamiseksi.” (College of Arts and Technology)

Web-kehityksellä tarkoitetaan verkkosivustojen tai -sovellusten kehittämistä internetiin julkaisutarkoituksessa. Web-kehitys voidaan jakaa kahteen pääkategoriaan. Front-End -kehitykseen, joka keskittyy käyttöliittymän rakentamiseen ja käyttökokemuksen luontiin sekä Back-End-kehitykseen, joka käsittelee palvelinpuolen logiikkaa ja tietokantoja. Bubblesa nämä ovat integroitu samaan sovellukseen, joka kommunikoi automaattisesti palvelimen tietokannan kanssa.

HTML, CSS ja JavaScript -ohjelmointikieliä pidetään käyttöliittymäkehityksen ydinkielenä, joista HTML:ää käytetään sisällön jäsentämiseen, CSS:ää tyylin luomiseen ja JavaScriptiä interaktiivisuuden lisäämiseen. ”CSS:n ja JavaScriptin ilmaantumisen ansiosta kehittäjät pystyvät luomaan monimutkaisempia ja interaktiivisempia verkkosivustoja.” (Mythili Raju, 2023 Lambdatest, 34 Web Development Trends in 2023: Embracing The Future) Uusien erikokoisten ja -mallisten laitteiden ilmaantuessa markkinoille verkkosivustojen luonnissa on entistäkin tärkeämpää ottaa huomioon responsiivisuus, eli sivuston mukautuminen erokokoisille näytöille ja laitteille.

3.2 No-code vs. Full-code

Full-codella koodatessa kehittäjät kirjoittavat koodia käyttämällä ohjelmointikieliä kuten Pythonia, Javaa tai JavaScriptiä. Tämä tapa antaa paremman hallinnan sovellukseen mutta vaatii asiantuntemusta koodauksesta. Full-code alustojen oppiminen voi olla haastavaa ja aikaa vievää sekä mahdollisesti vaatia koulutusta tai laajamuotoista itseopiskelua ja ohjelmointikielten opetteluun. No-code alustat, kuten Bubble, eivät vaadi koodaustaitoja perinteisessä mielessä. Koodittomilla alustoilla luodaankin sovelluksia rakentamalla logiikkaa visuaalisella koodaamisella. No-code alustat onkin suunniteltu siten, että niiden käyttö luonnistuu henkilöiltä, joilla on vähän tai ei ollenkaan ohjelmointikokemusta. Ne ovat usein helpoja oppia. Bubble tarjoaa lisäksi sovelluksen sisäisen dokumentaation teksti ja videomuodoissa kullekin toiminnallisuudelle. Tämä helpottaa oppimista entisestään. No-code alustat ovat huomattavasti aloittelijaystävällisempiä opetella verrattuna perinteisten koodikielten opetteluun sillä ne tarjoavat visuaalisen käyttöliittymän lisäksi valmiita komponentteja ohjelmiston kehitykseen.

No-code koodialustat tunnetaan nopeasta kehityksestään. Näillä sovelluksilla suunnittelu ja iterointi ovat huomattavasti perinteisempiä malleja nopeampaa. Bubble.io paikallistaa sovelluksessa esiintyvät virheet käyttäjän puolesta ja osoittaa niiden sijainnin. Tämä nopeuttaa virheidenkorjausta. No-code alustan käyttö saattaa olla kustannustehokkaampaa, varsinkin pienemmille yrityksille, koska se voi vähentää tai jopa eliminoida tarvetta palkata ulkopuolisia ammattilaisia sovel-

luksen kehittämiseen. Koodittomat alustat vähentävät huomattavasti verkkosovellusten rakentamiseen ja käyttöönottoon kuluvaan aikaan, jolloin yritykset voivat vastata nopeammin markkinoiden muuttuviin vaatimuksiin. Full-codella koodin kehittäminen vie huomattavasti paljon enemmän aikaa, varsinkin monimutkaisempien sovellusten kohdalla. Aikaa kuluu enemmän mm. koodaamiseen, virheenkorjaukseen ja testaamiseen.

No-code alustat sisältävät usein sisäänrakennetun tietokannan hallintajärjestelmän, mikä helpottaa sovelluksen ylläpitoa sekä päivitystä. Nämä järjestelmät eliminovat monimutkaisen palvelininfrastruktuurin tarpeen, johtaen kustannussäästöihin ja helpottaen ylläpitämistä. Full-code alustat vaativat manuaalista ylläpitoa ja päivittämistä sisältäen kirjastojen ja palvelimien hallinnan. Tämä vie usein enemmän aikaa mutta antaa ylläpitäjälle suuremman kontrollin. Toisaalta ylläpitäjällä on myös enemmän vastuuta.

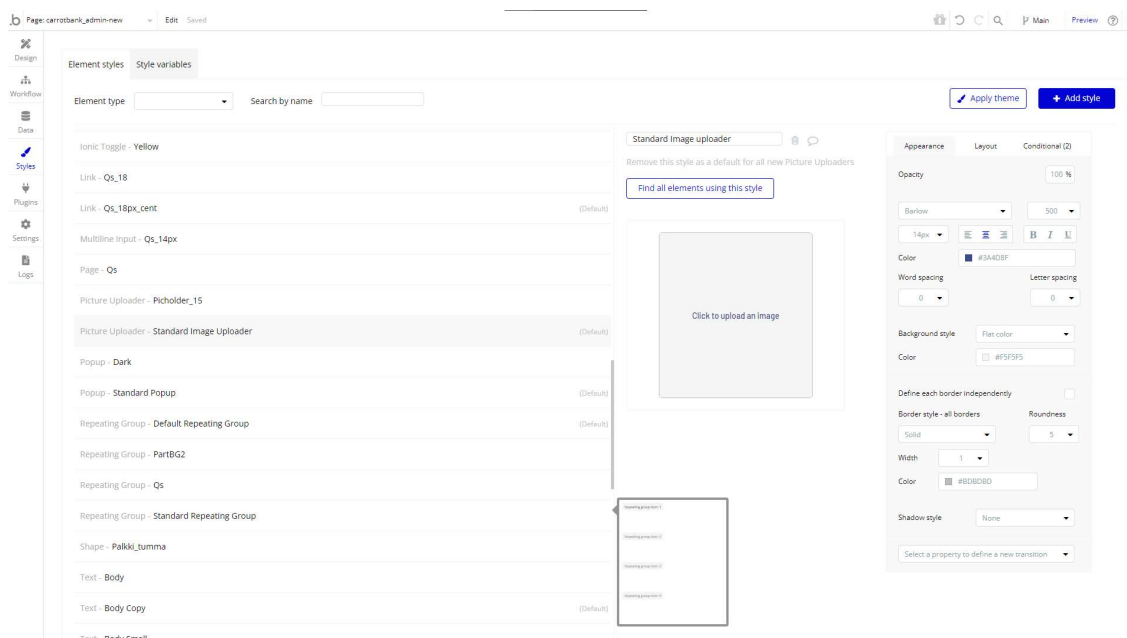
3.3 Bubble.io

Bubble.io on kooditon verkkokehitysalusta, jossa koodaamisen sijaan käyttäjät piirtävät käyttöliittymän rakentamalla erilaisia elementtejä ja asettelemalla niitä sivustolle tiettyyn järjestykseen määrittäen sivuston logiikan. Elementtien asetteleminen sivulle toimii vedä ja pudota -tavalla. Bubblessa on erilliset sivut grafiikan, logiikan ja tietokantojen rakentamiseen. Bubbleen on integroitu oma sisäänrakennettu tietokantojen hallintapaneeli, jossa tietokantaa voi tarkastella, hallita ja muokata reaaliajassa. Lisäksi sovellukseen voi rakentaa omia tyyliä (Kuva 1) ikään kuin CSS:ssä voi rakentaa koodipohjaisille sivuille ja sovelluksille tyyliä. Näitä tyyliä voi kehityksen aikana kierrättää useassa eri paikassa sovelluksessa ja samanlaisten elementtien muokkaaminen kaikkialla hoituu yhtä tyyliä muuttamalla. Bublleen natiivi sovellus tarjoaa myös API-toiminnon helpottamaan integraation luomista muihin ohjelmistoihin. Tämä sovellusliittymiin ja palveluihin integroituminen mahdollistaa yhteyden muodostamisen ja toimivuuden laajentamisen muiden työkalujen ja alustojen kanssa.

Bublleen tarjoamat visuaaliset kehitysympäristöt ja integroidut ominaisuudet yksinkertaistavat koko verkkokehitysprosessia, sillä käyttäjät voivat rakentaa ja muokata sovelluksiaan yhdistelemällä ja järjestelemällä elementtejä visuaalisesti. Nämä visuaaliset kehitysympäristöt pienentävät sovelluksen käyttöönoton kynnyksestä helpottamalla sovelluksen käyttöönottoa ja opettelua. Kuitenkin haastavampia, monimutkaista logiikkaa vaativien toiminnallisuuksien suunnittelussa visuaalisella koodaamisella voi olla perinteistä koodikieltä haastavampaa ja sekavampaa. Tämä johtuu pääasiallisesti rajallisista dokumentointimahdollisuuksista ja Bublleen visuaalisen koodikielen moniulotteisuuden aiheuttamasta koodin tulkitsemisen haastavuudesta.

Responsiivisten sovellusten rakentaminen on tehty Bubblessa helpoksi. Bubble mahdollistaa elementtien skaalautumisen sekä eri elementtiryhmien näkymisen tai piilottamisen sivuston koon vaihtuessa. Tämä responsiivinen rakennusmalli osoittautui opinnäytetyötä kehittäessä erittäin tärkeäksi, sillä Porkkanapankki on tehty ensisijaisesti käytettäväksi mobiililaitteilla ja mobiililaitteiden koot ja resoluutiot vaihtelevat merkittävästi eri laitteiden välillä. Elementtiryhmiä luomalla single-

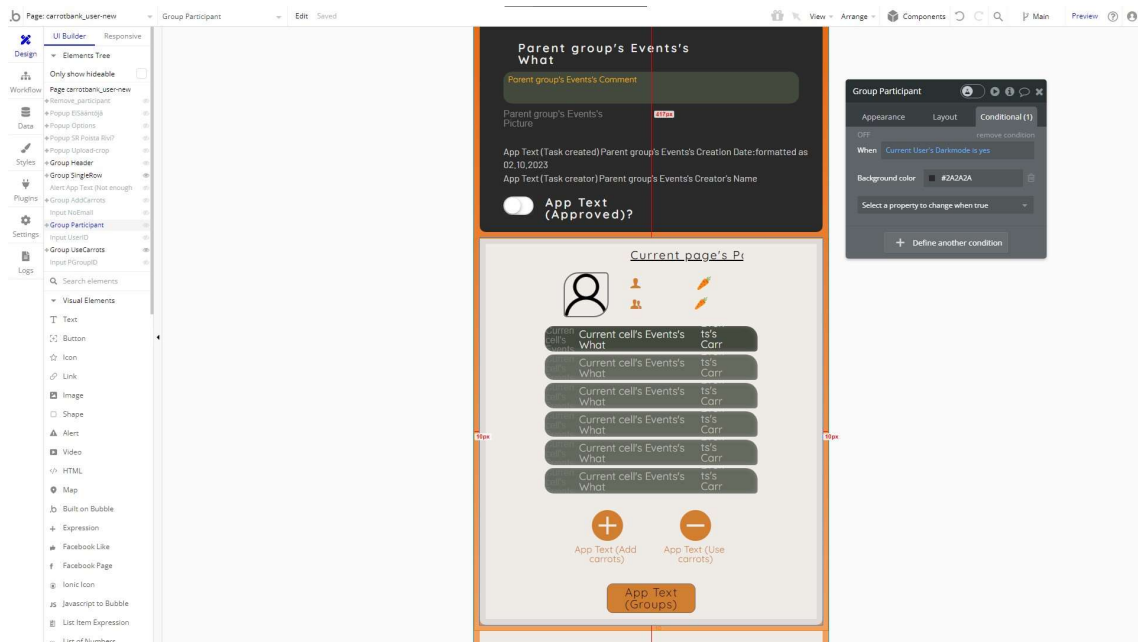
page sivustojen tekeminen on tehty mahdolliseksi. Toisaalta kun verkkosovelluksen sivumäärä kasvaa suureksi, on parametrien kuljetus sivujen välillä joissain tapauksissa käyttäjäystävällisempää, helpompaa ja kevyempää kun kaiken tiedon lataaminen yhdellä kertaa



KUVA 1. Bublenn tyylit sivu

3.3.1 UI näkymä

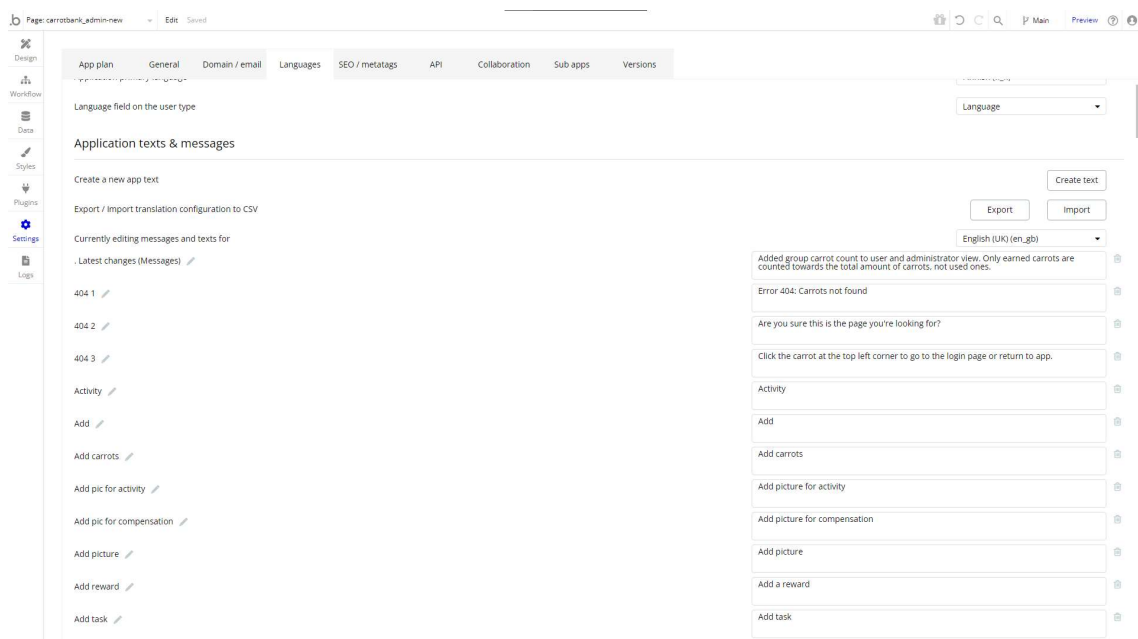
Bubble tarjoaa UI sivun käyttöliittymän ulkoasun ja toiminnallisuuden rakentamiseen. Sovellus tarjoaa valmiita komponentteja, joiden ulkoasua voi muokata appearance-valikosta sekä responsiivisuutta ja sijaintia layout-valikosta. Tämä responsiivinen rakentaminen on erityisen tärkeää aikakaudellamme, jossa sovelluksia voidaan käyttää lukuisilla erikokoisilla ja -mallisilla laitteilla. Elementteihin voi lisäksi lisätä ehtoja, kuten hover animaatioita conditional-valikosta (Kuva 2). Näitä animaatioita luodaan perinteisissä koodausmalleissa usein CSS koodikielellä.



KUVA 2. Bublenn UI sivu, jossa conditional valikko auki.

Bubble mahdollistaa myös kielten lisäyksen sovellukseen (Kuva 3). Kielet-valikosta voi luoda käännökset kaikkialle sivustolle ja uuden kielen lisättäessä käännökset toimivat kaikkialla sovelluksessa.

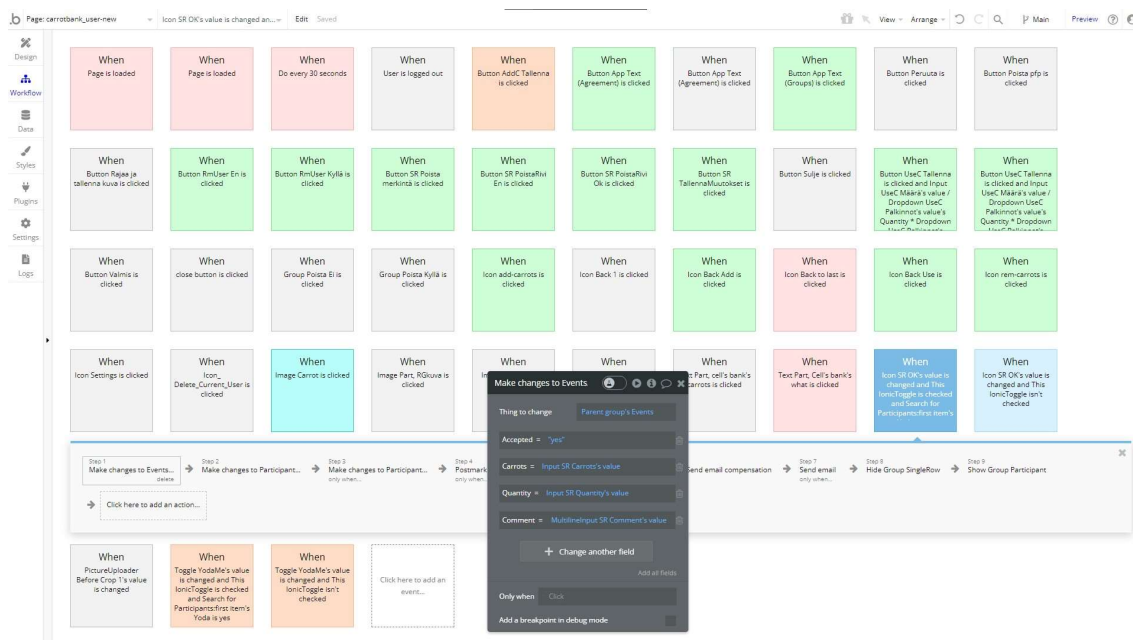
Sovellus päivittyy reaaliajassa muutoksia tehdessä. Tämä helpottaa ongelmien havaitsemista jo kehitysvaiheessa ja vähentää virheenkorjaukseen käytettävää aikaa.



KUVA 3. Bublenn Kielet sivun englanninkielinen osio

3.3.2 Workflow sivu

Bubblen workflow sivulla (Kuva 4) rakennetaan sivuston logiikkaa, kuten navigaatiota, elementtien ja elementtiryhmiä näyttämistä ja piilottamista, tietokannan taulun muokkaamista, kirjauksen lisäystä sekä animaatioita. Elementin voi lisätä osaksi sivuston workflowta ja tältä sivulta voi muokata mm. mitä tietyn napin painallus saa aikaiseksi, mitä sivun lataaminen saa aikaiseksi jne. Workflowiin voi lisätä myös ehtoja niin, että kyseinen workflow suoritetaan vain, jos ehdot toteutuvat. Näin visuaalisella koodaamisella voidaan helposti luoda if, else, for, while, ja case ehtoja. Workflowia rakentaessa kategorisointi on tärkeää ja sen voi tehdä lisäämällä workflowiin värejä ja järjestelemällä näitä alla näkyvässä kuvassa olevia niin sanottuja palikoita. Suurien, useita logiikkaelementtejä sisältävien, sivujen kohdalla Bubblen workflow sivu saattaa täytyä kymmenistä elementteistä, jolloin workflowin navigointi saattaa olla haastavaa. Bubble tarjoaa ratkaisuna tähän ongelmaan mahdollisuuden siirtää elementin workflowiin UI sivulta käsin. Tämä ei kuitenkaan ratkaise ongelmaa, mikäli workflow sivulta pitäisi esimerkiksi etsiä ja poistaa turha elementti.



KUVA 4. Bubblen Workflow sivu

3.3.3 Tietokanta sivu

Yksi Bubblen erottuvista ominaisuuksista on sen visuaalinen ja helppokäyttöinen tietokannan hallintajärjestelmä. Tämä sisäänrakennettu järjestelmä mahdollistaa helpon tietokannan muokkaamisen sovelluksen workflowista käsin. Tietokanta-sivu tarjoaa tehokkaan ja skaalautuvan tietokantaratkaisun, jonka avulla käyttäjät voivat määritellä tietokantarakenteita, määrittää tietotyyppien välisiä suhteita sekä luoda dynaamista sisältöä sovellukseen (Kuva 5). Tietokanta päivittyy sovelluksen kanssa reaaliajassa ja tietokannasta käsin rivin muokkaaminen muuttaa sen myös sovellukseen. Tämä nopeuttaa ja helpottaa suurienkin käyttäjämäärien hallitsemisen. Tietokanta sivu mahdollistaa myös rivien manuaalisen muokkaamisen sivulta käsin.

The screenshot shows the Bubble database management interface. At the top, there are tabs for 'Data types', 'Privacy', 'App data', 'Option sets', and 'File manager'. The 'App data' tab is selected, showing 'Application data - All Users - Development version'. Below this, there are search and filter options, including 'Data entries' (416 entries) and buttons for 'New entry', 'Delete (0)', 'Upload', 'Modify', 'Export', and 'Bulk'. The main area displays a table with columns: 'Email', 'Darkmode', 'Language', 'Last Logged In', 'Name', 'Created Date', 'Modified Date', and 'Slug'. The table contains multiple rows of user data, including names like Emma, Sari, Seppo, Juhana, and others, along with their creation and modification dates. A 'Refresh data' button is located at the bottom left of the table area.

	Email	Darkmode	Language	Last Logged In	Name	Created Date	Modified Date	2 additional fields
Run as...				Oct 4, 2023 8:41 am	Emma G.	Oct 4, 2023 11:11 am	Oct 4, 2023 11:11 am	
Run as...				Oct 4, 2023 8:23 am	Info DM	Oct 4, 2023 8:41 am	Oct 4, 2023 8:41 am	
Run as...				Oct 4, 2023 8:57 am	Info DM	Oct 3, 2023 8:45 pm	Oct 4, 2023 8:57 am	
Run as...				Oct 4, 2023 9:49 am	Tia H.	Oct 3, 2023 8:38 pm	Oct 4, 2023 9:49 am	
Run as...					Erno	Oct 2, 2023 3:24 pm	Oct 4, 2023 5:07 pm	
Run as...		fi_fi		Oct 3, 2023 3:49 am	Essi	Oct 2, 2023 3:24 pm	Oct 4, 2023 5:10 pm	
Run as...		fi_fi		Oct 3, 2023 3:20 am	Juhani	Oct 2, 2023 3:24 pm	Oct 4, 2023 5:11 pm	
Run as...		en_gb		Oct 2, 2023 8:50 am	Maailda	Oct 2, 2023 3:22 pm	Oct 4, 2023 5:11 pm	
Run as...		en_gb		Oct 2, 2023 8:50 am	James	Oct 2, 2023 3:20 pm	Oct 4, 2023 5:11 pm	
Run as...		en_gb		Oct 2, 2023 8:23 am	Erik	Oct 2, 2023 3:20 pm	Oct 4, 2023 5:11 pm	
Run as...		en_gb		Oct 2, 2023 8:09 am	Tuula R.	Oct 2, 2023 3:09 pm	Oct 2, 2023 3:09 pm	
Run as...		en_gb		Oct 2, 2023 8:10 am	Anneli	Oct 2, 2023 3:09 pm	Oct 4, 2023 5:11 pm	
Run as...		fi_fi		Oct 2, 2023 5:53 am	Matti V.	Oct 2, 2023 3:09 pm	Oct 4, 2023 5:11 pm	
Run as...		fi_fi			Emma	Oct 2, 2023 3:09 pm	Oct 2, 2023 3:09 pm	
Run as...		fi_fi			Sari	Oct 2, 2023 3:09 pm	Oct 4, 2023 5:08 pm	
Run as...		fi_fi			Seppo	Oct 2, 2023 3:08 pm	Oct 4, 2023 5:09 pm	
Run as...		fi_fi			Juha	Oct 2, 2023 3:08 pm	Oct 4, 2023 5:09 pm	
Run as...		fi_fi			Anne	Oct 2, 2023 3:08 pm	Oct 4, 2023 5:09 pm	
Run as...		fi_fi			Tuula S.	Oct 2, 2023 3:08 pm	Oct 2, 2023 3:08 pm	
Run as...		fi_fi			Matti H.	Oct 2, 2023 3:08 pm	Oct 4, 2023 5:09 pm	
Run as...		fi_fi		Sep 28, 2023 4:42 pm	Aleksi	Sep 28, 2023 4:31 pm	Sep 28, 2023 4:42 pm	
Run as...		fi_fi		Sep 27, 2023 2:37 pm	Tarja V.	Sep 27, 2023 9:22 am	Sep 27, 2023 2:37 pm	
Run as...		fi_fi			Sanna V.	Sep 27, 2023 9:21 am	Sep 27, 2023 9:21 am	
Run as...				Oct 2, 2023 7:44 pm	Emmi HT	Sep 26, 2023 3:26 pm	Oct 2, 2023 7:44 pm	
Run as...					Jouuri T.	Sep 26, 2023 3:22 pm	Sep 26, 2023 3:22 pm	
Run as...		fi_fi		Sep 26, 2023 5:16 pm	Amos G.	Sep 26, 2023 3:20 pm	Sep 27, 2023 9:14 am	
Run as...					Info DM	Sep 26, 2023 3:18 pm	Sep 26, 2023 3:18 pm	
Run as...		fi_fi		Sep 27, 2023 7:14 am	Mia L.	Sep 26, 2023 2:17 pm	Sep 27, 2023 7:14 am	
Run as...					Test 3	Sep 22, 2023 1:54 pm	Sep 22, 2023 1:54 pm	
Run as...					Test 2	Sep 22, 2023 1:54 pm	Sep 22, 2023 1:54 pm	
Run as...					Test 1	Sep 22, 2023 1:54 pm	Sep 22, 2023 1:54 pm	

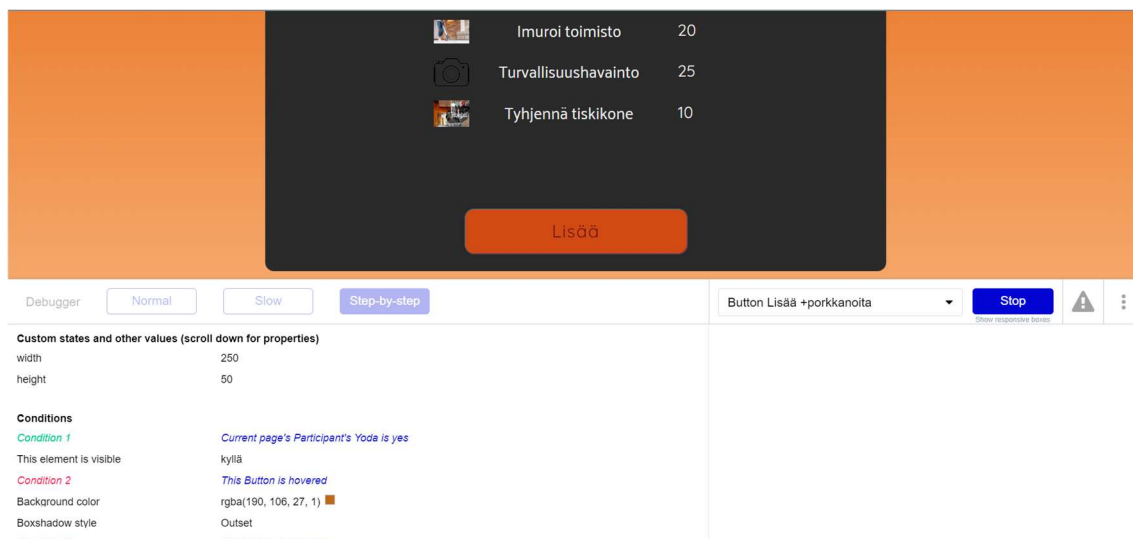
KUVA 5. Bubblen Tietokanta sivu

3.3.4 Muut ominaisuudet

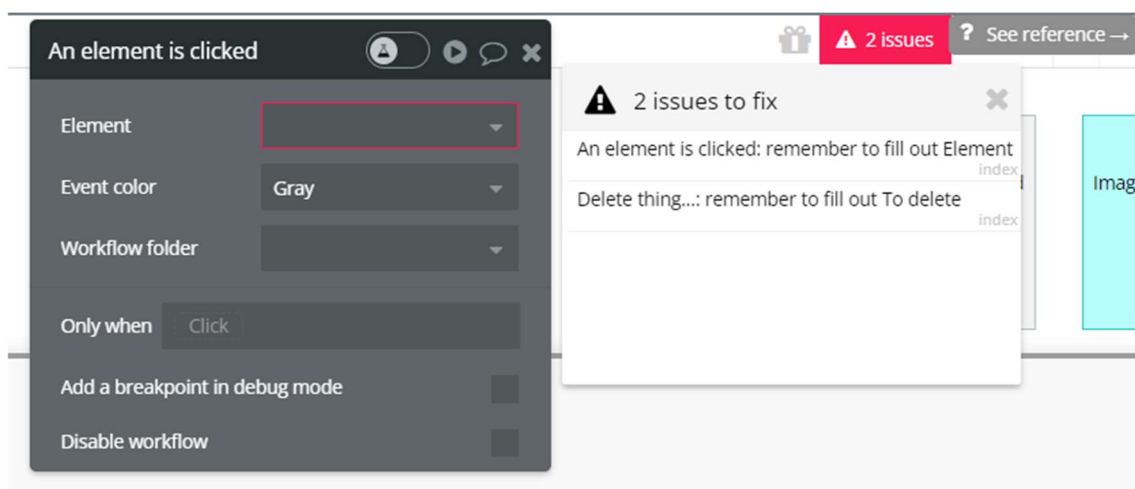
Bubblesissa on myös state-ominaisuus, joka mahdollistaa mm. useamman eri näkymän vaihtamisen yhtä statea vaihtamalla. Yksi esimerkki statesta voi olla sivuston teema. Käyttäjän kääntäessä teema vivun vaaleasta tummaan teemaan, käyttäjän state vaihtuu tietokannassa vaaleasta tummaan ja sivun vaalea tausta muuttuu tummaksi. Porkkanapankissa stateilla on luotu teeman lisäksi myös mahdollisuus näyttää ja piilottaa salasana mm. kirjautuessa tai salasanaa vaihdettaessa. Bubblesin sisäänrakennettu state-ominaisuus on hyvin samankaltainen JavaScript kielestä löytyvään state ominaisuuteen.

Porkkanapankki sovellus lähettää Bubblesin natiivilla sähköpostiominaisuudella käyttäjälle sähköpostitse tunnukset, kun hänelle luodaan tili. Sovellus ilmoittaa myös, kun käyttäjän tekemä tehtävä on hyväksytty. Sähköpostin lähetyksen voi vaihtoehtoisesti toteuttaa integroimalla kolmannen osapuolen sähköpostinhallintasovellus kuten Brevo tai Mailgun.

Virheenjäljitys, eli debuggaus on prosessi, jossa tunnistetaan koodissa olevia virheitä. Debuggauksessa analysoidaan koodia sekä jäljitetään ongelman lähdettä, kuten sovelluksen kaatumista tai odottamatonta toimintaa. Virheiden jäljitys ja korjaus ovat tärkeä osata sovelluksen luotettavuuden ja toimivuuden varmistamiseksi. Bubble tarjoaa oman Debuggaus-työkalun, (Kuva 8) jolla sovelluksen toimintaa on helppo analysoida vaihe vaiheelta. Bubble ei myöskään anna julkaista sovelluksen uutta versiota pakottamatta julkaisua, mikäli virheitä on korjaamattomina. (Kuva 9)



KUVA 8. Bubblen oma debuggaus-työkalu

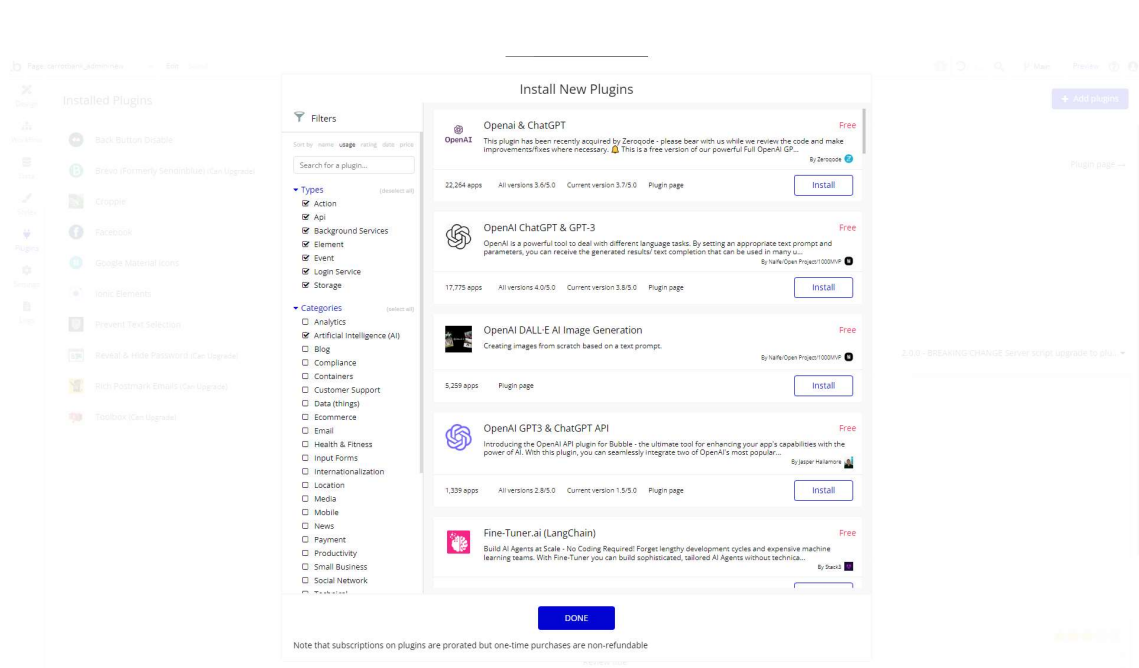


KUVA 9. Bubblen oma virheistä varoittava näkymä

Hostaus eli palveluntarjonta on välttämätöntä verkkosivustoille ja -sovelluksille, jotka ovat yhteydessä internetiin. Myös versionhallinta on tärkeää sovellusta kehitettäessä. Monet koodaajat käyttävät versionhallintajärjestelmiä, kuten Git:iä, koodimuutosten ja versioiden hallintaan. Bubble tarjoaa oman hostauspalvelun sovelluksilleen, sekä oman versionhallintajärjestelmänsä, jolla sovelluksen versioita voi päivittää, palauttaa ja hallita.

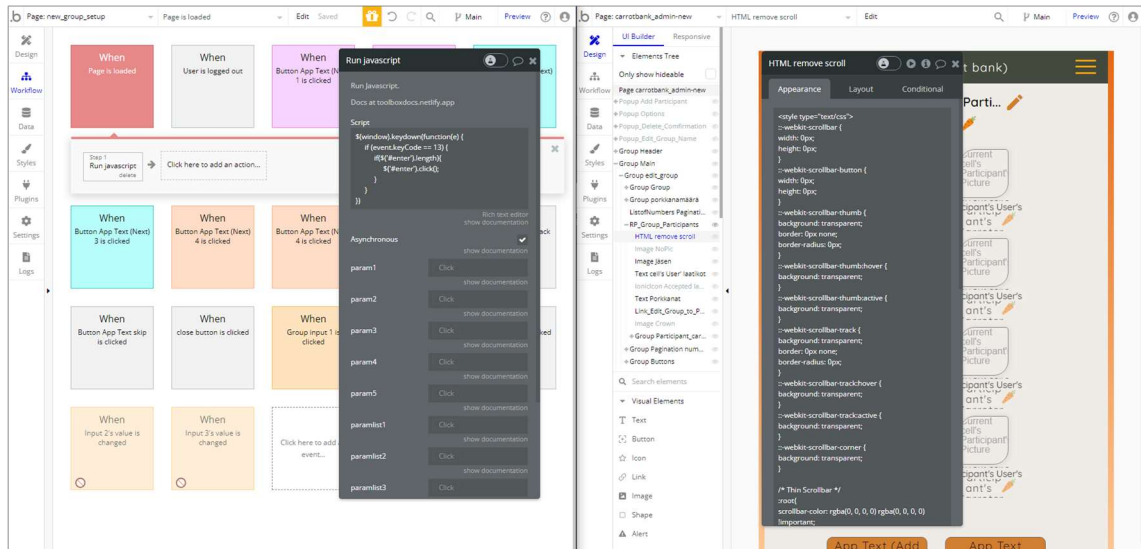
3.4 Bubblen lisäosat ja koodikieli

Sovellusta kehitettäessä Bubblen laajasta lisäosakirjastosta oli merkittävästi hyötyä (Kuva 6). Lisäosien avulla sovellukseen voitiin lisätä Bubblen omia tai kolmannen osapuolen luomia ominaisuuksia ja toiminnallisuuksia, joita Bubblen natiivi sovellus ei itsessään tue. Lisäosat voivat myös tarjota helpomman, nopeamman ja/tai kevyemmän tavan toteuttaa jokin toiminnallisuus. Kaikki Bubblen tukevat lisäosat ovat ladattavissa sisäänrakennetusta lisäosakirjastosta.



KUVA 6. Bubblen lisäosakirjasto

Bubblen Toolbox-niminen JavaScript laajennus mahdollisti JavaScript elementtien lisäämisen sovellukseen. (Kuva 7) Tämän avulla sovellukseen pystyttiin lisäämään mm. ylläpitäjänäkymän sivunumerointi sekä mahdollistamaan näppäin komennot sivustolla. Toinen sivulla laajasti käytetty lisäosa on Ionic Elements. Tämä lisäosa tarjoaa lukuisia ikoneita Bubblen oman valikoiman lisäksi. JavaScriptin lisäksi sivustolla on käytetty HTML elementtiä. Kyseinen elementti löytyy valmiiksi Bubblen natiivista sovelluksesta eikä sitä tarvitse asentaa erikseen. Sovelluksessa tätä elementtiä on käytetty mm. poistamaan vierityspalkki oikeasta reunasta.



KUVA 7. JavaScript elementti workflowissa (vasen) sekä HTML Elementti UI sivulla (oikea)

Bubblen tarjoamat mahdollisuudet sivustojen rakentamiseen ovat käytännössä rajattomat. Bubblen markkinoimiseksi käyttäjille, sovelluksella onkin kopioitu useita suosittuja sovelluksia, kuten Airbnb, Twitter ja Uber. Bubblella on luotu lukuisia verkkokauppoja, sosiaalisen median kanavia, rahoitussovelluksia sekä blogeja. Bubblella on myös mahdollista tehdä mobiilisovelluksia käärimällä verkkosovellus BDK-laajennuksella ja julkaista se Apple- ja/tai Play kauppaan. Natiivimallien kehitys Bubblella ei kuitenkaan ole mahdollista, kuitenkin kiitos Bubblen responsiivisen rakennusmallin, sovellukset on mahdollista rakentaa skaalautumaan kaikenkokoisille ja -mallisille laitteille.

Porkkanapankki sovelluksessa on testattu ja toimiva Rest-rajapinta. Rajapinnan avulla sovellus voi käyttää HTTP pyyntöjä ja menetelmiä. Bubblella on oma versionhallintatyökalu, jolla sovelluksen versioita voi hallita.

3.5 Tekoälyn hyödyntäminen sovelluskehityksessä yleisesti

Elämme parhaillaan tekoälyn mullistuksessa ja tekoälyn integraatiosta on tulossa jatkuvasti yhä suositumpaa myös sovelluskehityksessä. Tekoälyintegraatiolla voidaan helposti räätälöidä sovellus käyttäjälle hyödyntämällä esimerkiksi erilaisia algoritmeja. Tämä näkyy jo muun muassa sosiaalisen median tarjoamassa personalisoidussa syötteessä. Tekoälyä hyödynnetään myös kyberturvallisuuden vahvistamisessa. Tekoälyn avulla voidaan tunnistaa ja vähentää uhkia ja sitä käytetään myös laajalti esimerkiksi suodatamaan roskasisältöä, sekä avustamaan sisällön luonnissa. Tekoälyintegraatio mullistaa ohjelmistokehitystä jatkuvasti.

Bubblen oma kirjasto tarjoaa lukuisia tekoälyä hyödyntäviä lisäosia, kuten OpenAIn kehittämät GhatGPT ja Dall-E laajennukset sekä mm. opetettavia Chatbotteja ja Puhe tekstiksi tai Teksti puheeksi -ohjelmia. Nämä laajennukset helpottavat tekoälyn integrointia sovelluksiin huomattavasti ja tekoälyn kehittyessä niistä tulee entistäkin tärkeämpiä työkaluja käyttäjäystävällisten ja tehokkaiden sovellusten luomisessa.

Tekoälyn integrointi Bubblen sovelluksiin voi parantaa niiden toimivuutta ja tehdä niistä entistäkin reaktiivisempia ja käyttäjäystävällisempiä. Esimerkiksi teksti puheeksi -ohjelmilla voidaan parantaa sovelluksen esteettömyyttä mahdollistamalla sen käyttö näkörajoitteesta riippumatta. Toisena esimerkkinä tekoälyn integraatiomahdollisuuksista Bubbleen mainittakoon tekoälypohjainen automaatio. Automaatisoimalla joitain toisteisia tehtäviä sovelluksen käyttöä voidaan sujuvoittaa. Automaation avulla voidaan esimerkiksi luoda sivustolle Chatbot, joka vastaa käyttäjien ongelmiin ja ohjaa vasta epäonnistuessaan ratkaisemaan käyttäjän ongelman hänet oikealle asiakaspalvelijalle. Tämä säästää yrityksen henkilöstökuiluissa ja mahdollisesti nopeuttaa ratkaisun tarjoamisessa asiakkaan ongelmantilanteeseen. ” Väittäisin, että ohjelmistokehityksen tulevaisuus on kooditon kehitys + tekoäly” (Joshua Haas 2023, Why AI + No-Code Is the Future.)

3.6 Yhteenveto Bubblesta alustana

Bubble.io on muiden koodittomien alustojen ohella mahdollistanut helpon kehityksen laajemmalle yleisölle. Bubble mahdollistaa yksinkertaisten sivustojen ja sovellusten luomisen riippumatta siitä onko kehittäjällä aikaisempaa koodaus-taustaa. Visuaaliset koodausalustat eivät ole ainakaan toistaiseksi syrjäyttä-mässä perinteisiä malleja, mutta niiden helppokäyttöisyys tarjoaa hyvän vaihto-ehdon esimerkiksi yrityksen kotisivujen luontiin. Bubblen visuaalinen kehitysmalli, kestävät ominaisuudet, sekä lukuisat integraatio-ominaisuudet tekevät siitä hou-kuttelevan valinnan monille. Koodittomien alustojen kehittyminen muuttaa jatku-vasti tapaamme lähestyä verkkokehitystä ja tekee siitä entistäkin helpompaa, te-hokkaampaa ja halvempaa. Koodittomien alustojen kehittyessä voidaan odottaa uusia innovaatioita verkkokehityksen maailmassa. Bubble toimi loistavana työka-luna Porkkanapankki-sovellusta luodessa, sillä ominaisuuksien prototypointi oli nopeaa ja samasta ominaisuudesta pystyi luomaan useita versioita, joista paras toteutettiin sovellukseen.

Huonona puolena Bubblesa havaittiin sen rajalliset dokumentaatiomahdollisuu-det. Bubble ei tarjoa mahdollisuutta kommentoida elementtejä erikseen, joka saattaa aiheuttaa haasteita logiikan ymmärtämisessä koodaajan vaihtuessa. Haasteita aiheutti myös Bubblen visuaalisen koodikielen moniulotteisuuden ai-heuttama koodin tulkitsemisen vaikeus, etenkin monimutkaisia logiikkarakenteita luodessa. Alusta ei tarjoa mahdollisuutta etsiä sanahaulilla tiettyä elementtiä ja tämä aiheutti aika ajoin haasteita elementtien löytämisessä.

Voidaan todeta Bubblen toimivan hyvänä vaihtoehtona pieniä sovelluksia tai si-vustoja rakennettaessa, varsinkin jos kehittäjä ei omaa koodaustaitoja perinteis-sessä mielessä. Suuria ja monimutkaisia logiikkarakenteita vaativia sovelluksia kehitettäessä perinteisemmät koodiratkaisut voivat olla helpompia hallita.

4 TYÖN TOTEUTUS

Tässä kappaleessa käsitellään Porkkanapankin kehitystä osa-alue kerrallaan. Työllä oli valmis runko, jossa oli muutamia elementtejä ja toiminnallisuuksia valmiina. Toimenkuvana oli parantaa näitä olemassa olevia muutamia ominaisuuksia sekä lisätä työn tilaajan esittämiä muutoksia ja uusia ominaisuuksia. Sovelluksen ulkoasu koki kehityksen aikana täyden muodonmuutoksen.

4.1 Tuotokseen painottuva opinnäytetyö

Tämä opinnäytetyö on tuotokseen painottuva opinnäytetyö. Opinnäytetyön kirjallisen tuotoksen lisäksi on kehitetty verkkosovellus yritykselle. Tämä mobiililaitteille suunniteltu verkkosovellus on julkaistu ja se on otettu käyttöön useassa yrityksessä. Työssä merkittävässä roolissa oli käytännönläheisyys sekä työelämälähtöisyys. Kehityksen aikana pyrittiin käyttämään sellaisia menetelmiä, jotka vastaavat työelämässä käytettäviä menetelmiä. Opinnäytetyö pohjautui yritykseltä saatuun toimeksiantoon. Opinnäytetyön aihe pohjautuu oman alan opintoihin. Työn alkuvaiheessa suunnitelluista ominaisuuksista toteutuivat lähes kaikki.

4.2 Suunnittelu

Opinnäytetyön ensimmäinen vaihe oli perehtyä varsinaiseen toimeksiantoon. Toimeksiantona oli kehittää Porkkanapankki-nimistä sovellusta. Sovellukseen oli tarkoitus lisätä joitakin ominaisuuksia, sivuston ulkoasu piti päivittää ja sivustosta tuli tehdä responsiivinen sekä kevyt käyttää. Lisäksi sovelluksesta piti muuttaa joitakin sen valmiista ominaisuuksista toisenlaisiksi.

Tilattua työtä suunniteltiin toimeksiantajan kanssa koko projektin keston ajan viikoittaisten tapaamisten muodossa. Tapaamisissa käytiin läpi työn sen hetkinen tila ja työn tilaaja pääsi kommentoimaan tehtyä työtä ja määrittämään seuraavia askeleita. Sovelluksen ulkoasua suunniteltiin prototyyppiä käyttäen sitä Bubblesa,

Figmassa ja Canvassa. Prototyypointiin kuului mm. värimaailman suunnittelu, sivun asemointi, elementtien suunnittelu sekä logiikan rakentaminen.

4.3 Toteutus

Porkkanapankki-sovellusta lähdettiin kehittämään ketterällä kehitysmallilla. Tarkoituksena oli kehittää sovellukseen yhtä näkymää kerrallaan ja julkaista uusi versio aina yhden näkymän valmistuessa. Alussa sovellusta päivitettiin ketterällä mallilla ja yhteen päivitykseen kuului ainoastaan yhden tai kahden näkymän päivittäminen tai esimerkiksi koko sivun taustan värien tai painikkeiden tyylin päivittäminen. Myöhemmässä vaiheessa projektia tästä mallista täytyi kuitenkin luopua sovelluksessa esiintyvien ongelmien takia. Sivustoa pyrittiin kehittämään nopealla tahdilla ja ongelmien ilmentyessä ne sivuutettiin hetkellisesti. Tästä syystä loppuvaiheessa sivustoa kehitettiin niin monesta paikasta samanaikaisesti, ettei sen julkaiseminen ollut mahdollista ennen kaikkien ongelmien ratkaisemista ja kattavan käyttäjätestauksen suorittamista.

Kehityksen alussa pyrittiin saamaan sovelluksen logiikasta toimiva ja virheetön. Ominaisuuksia pyrittiin lisäämään ja parantelemaan ja graafinen ulkoasu sekä helppokäyttöisyys jäi toissijaiseksi asiaksi. Alkuvaiheessa projektin edistyminen oli hitaampaa uuden alustan opetteluun kuluneesta ajasta johtuen. Tämä kuitenkin nopeutui jatkuvasti projektin edistyessä ja alustan käydessä jatkuvasti tutummaksi.

Kehityksen puolivälin tienoilla painopiste oli sovelluksen logiikan ja grafiikan yhdenmukaistamisessa sekä tietokannan muovaamisessa vastaamaan sovelluksen sen hetkistä tilaa. Sovelluksesta alettiin hioa hiljalleen versiota, joka vastaa sovelluksen nykytilaa. Tässä vaiheessa Bubbles alustan käyttäminen oli jo tuttua ja aikaa ei kulunut ominaisuuksien opetteluun.

Kehityksen loppuvaiheessa painopistettä siirrettiin ominaisuuksien lisäämisen sijaan käyttäjystävällisyyden, kuten helppokäyttöisyyden ja esteettömyyden, parantamiseen. Myös grafiikan parantaminen oli oleellista. Sovelluksen väripaletti piti suunnitella porkkanamaiseksi mutta kuitenkin kontrastitestiä läpäiseväksi ja

helppokäyttöiseksi. Tämä vaati paljon suunnittelua ja osoittautui odotettua hankalammaksi. Nämä ongelmat saatiin lopulta selätettyä ja sovelluksen lopullinen värimaailma vastasi tilaajan toiveita helppokäyttöisyydestä tinkimättä.

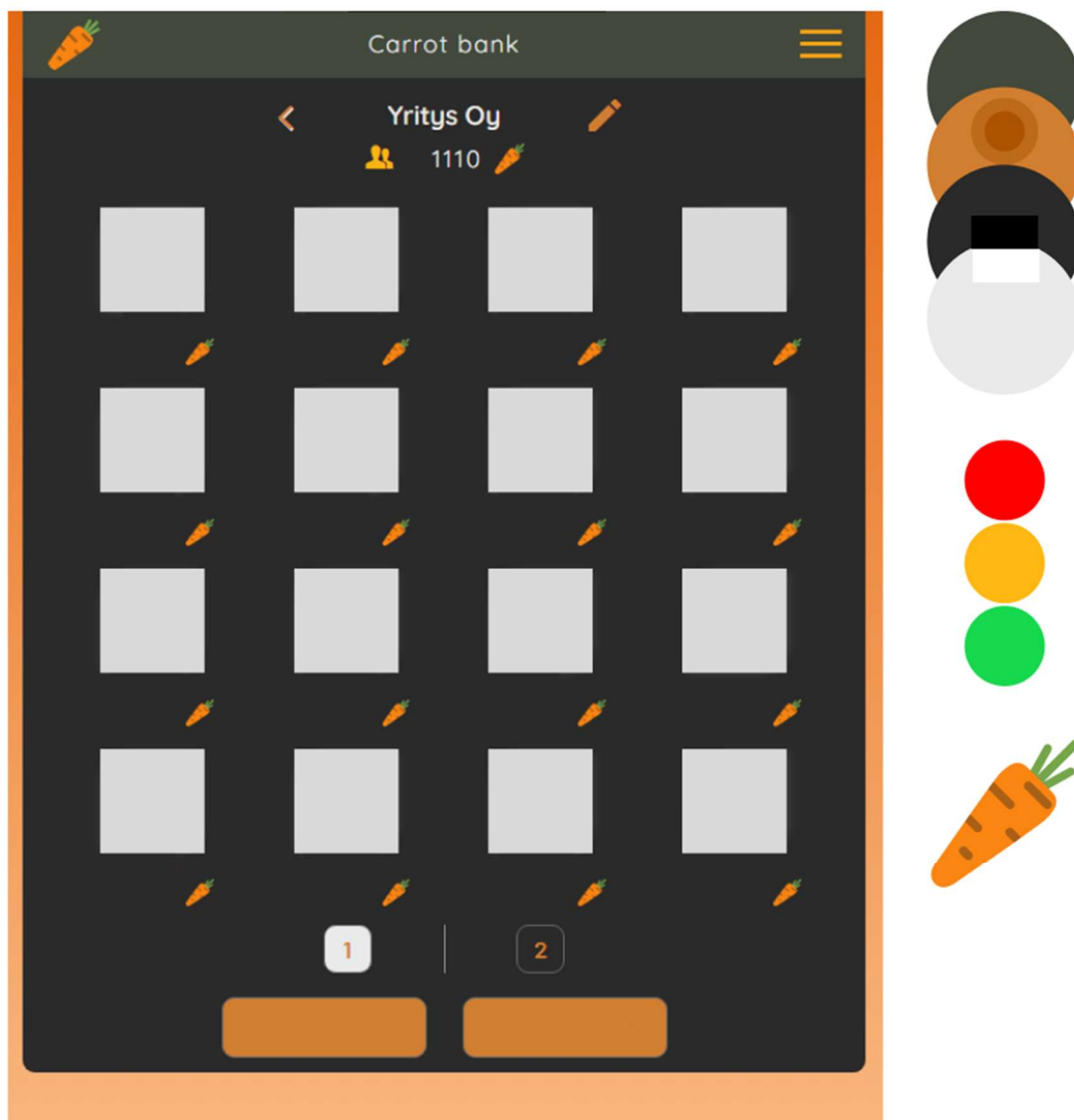
4.4 Grafiikka ja ulkoasu

Grafiikalla on tärkeä rooli verkkosovelluksissa. Graafinen ulkoasu vaikuttaa merkittävästi sivuston helppokäyttöisyyteen ja käyttömukavuuteen. Ulkoasu antaa sivustosta ensivaikutelman. Ensivaikutelma saattaa olla käänteentekevässä roolissa siinä ostaako joku sovelluksen tai palvelun.

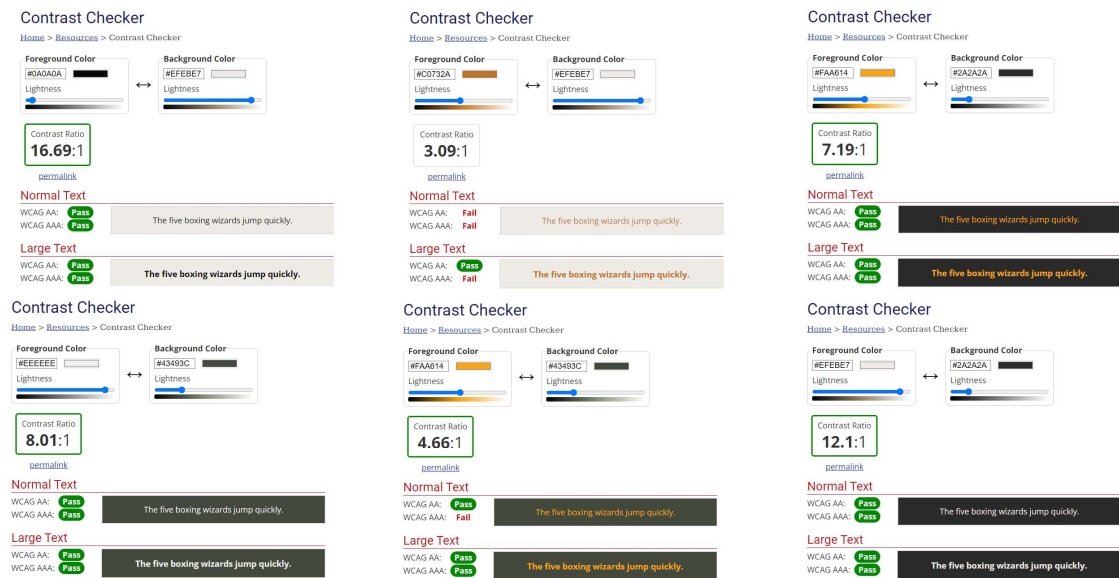
4.4.1 Värimaailma ja esteettömyys

Porkkanapankin ulkoasua suunniteltaessa värimaailmasta koitettiin saada mahdollisimman porkkanamainen. Sivuston pääväreinä toimii mustan ja valkoisen lisäksi vihreä ja oranssi. (Kuva 10) Sovelluksessa käytetyille pääväreille suoritettiin kontrastitestit esteettömyyden parantamiseksi. (Kuva 11) Värit valittiin siten, että niiden välillä olisi riittävä kontrasti tekstin lukemisen helpottamiseksi. Fonteista ja fonttiko'oista pyrittiin valitsemaan selkeät ja helppolukuiset fontit esteettömyyden turvaamiseksi. Tällä varmistettiin, että sovelluksen käyttö lievästä näkörajoitteesta huolimatta olisi mahdollista. Selaimen sivukokoa kasvattamalla sovelluksen teksti- ja painike-elementit kasvavat responsiivisesti. Tämä osaltaan parantaa heikompinäköisten mahdollisuuksia käyttää sivua.

Sovelluksen värimaailmasta, elementtien muodoista sekä animaatioista pyrittiin rakentamaan yhdenmukaiset ja helppokäyttöiset, mutta myös esteettisesti miellyttävät. Bubble tarjoaa mahdollisuuden luoda tyylejä sovelluksen sisäisesti, samoin kuin CSS:ää voidaan käyttää HTML:n lisänä. Näitä tyylejä käyttämällä elementeistä saatiin yhdenmukaiset. Sovelluksen varoitusväriksi valittiin kirkkaan keltainen tai punainen riippuen varoituksen vakavuudesta. Kirkkaan vihreä väri taas kertoo toiminnon onnistumisesta ongelmitta.



KUVA 10. Sovelluksen värimaailma sekä Figma prototyyppi sovelluksen admin näkymästä



KUVA 11. Sivun pääväreille Webaim:illa tehtyt kontrastitestit.

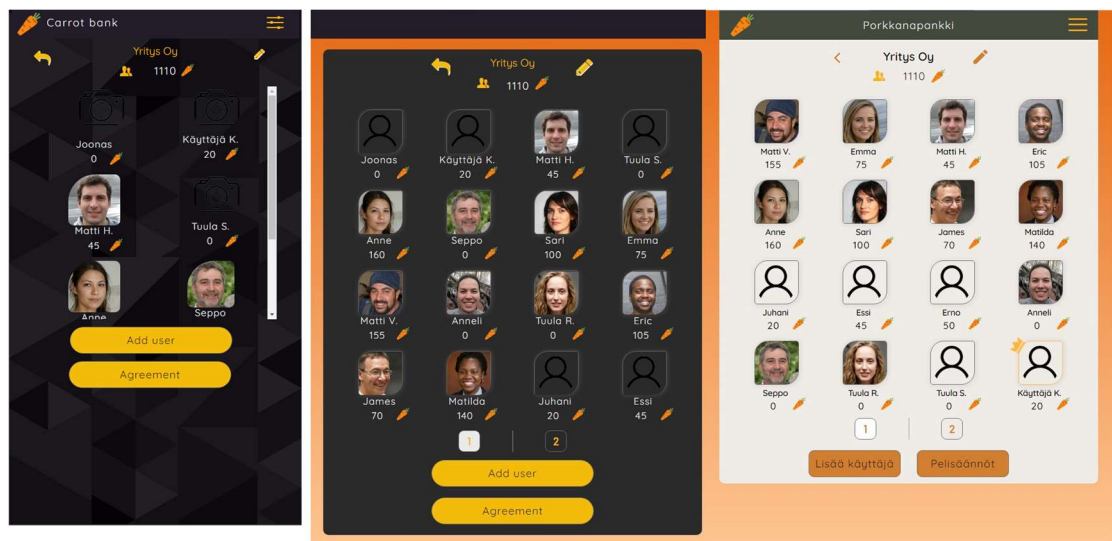
4.4.2 Elementtien asettelu ja intuitiivisuus

Elementtien asettelussa otettiin huomioon niiden sijainnin intuitiivisuus. Esimerkiksi peruuta ja eteenpäin painikkeet pyrittiin asettamaan siten, että niiden sijainti olisi siellä missä käyttäjä ensisijaisesti olettaisi niiden olevan. Tämä toteutettiin käyttäjätestauksella, tutkimalla markkinoilta löytyviä suosittuja sovelluksia sekä kyselemällä joukolta ihmisiä mistä he lähtisivät ensimmäisenä edellä mainittuja painikkeita etsimään. Elementtien asettelussa otettiin huomioon myös yleiset standardit, kuten se että takaisin painike löytyy ruudun vasemmasta yläreunasta ja niin sanottu hampurilaisvalikko asetuksille löytyy toisesta yläkulmasta. Tärkeimpien ja useimmiten käytettyjen elementtien asettelu sellaisiin paikkoihin, mihin käyttäjän sormet ylettyvät puhelinta kädessä pidettäessä, otettiin huomioon sivuston asettelussa. Sivustoon lisättiin myös mahdollisuus vaihtaa väriteema tummaan tilaan. Tämä muuttaa taustan värin tummaksi ja näin säästää silmiä sekä sopii paremmin käytettäväksi hämärissäkin tiloissa. Väriteeman vaihtaminen voi joillekin käyttäjille olla myös preferenssikysymys.

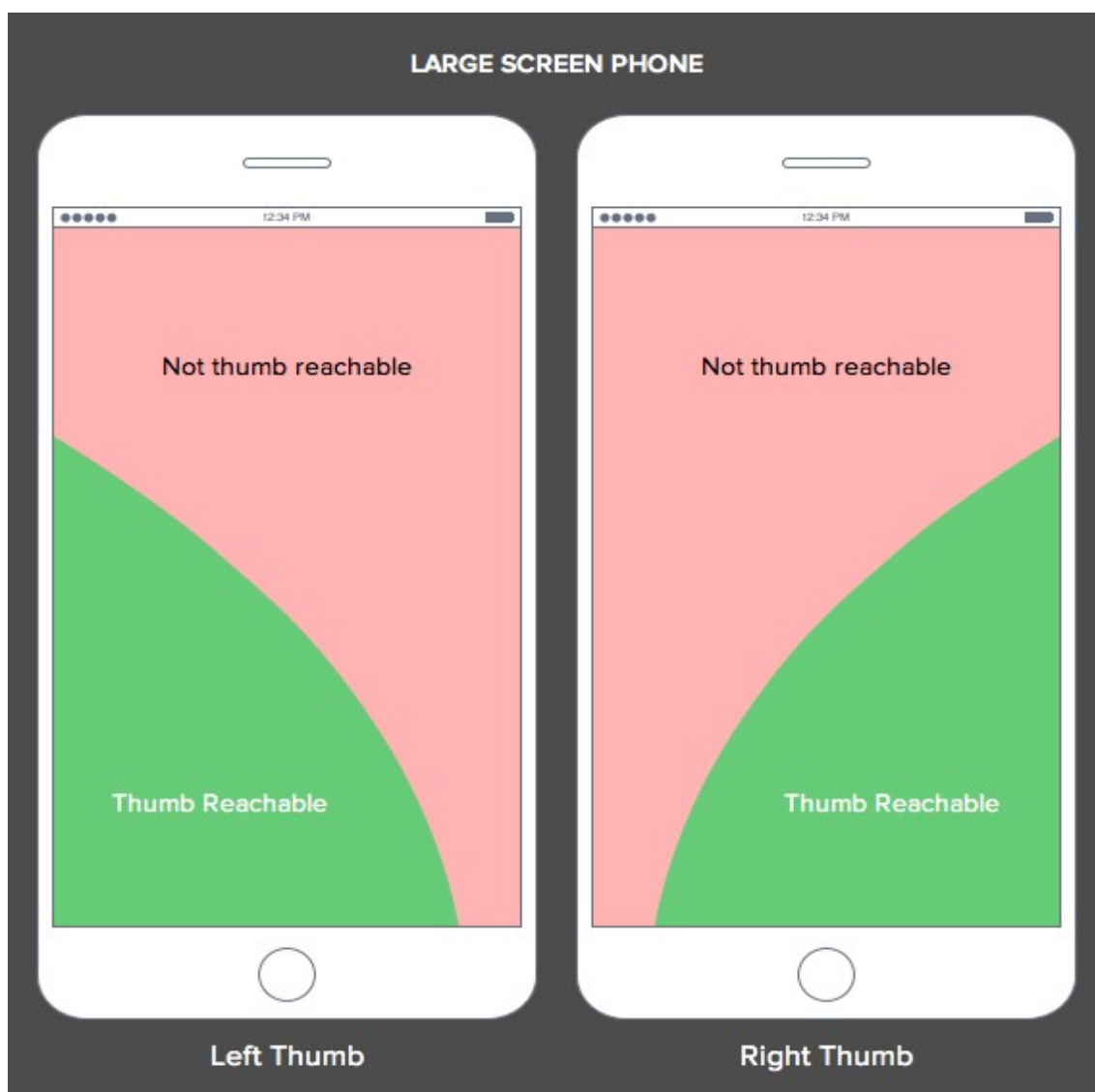
Sovellusta iteroitiin sen kehityksen aikana useaan eri otteeseen niin ominaisuuksiltaan kuin grafiikaltaan. Sovelluksesta onkin olemassa 3 varsinaista isompaa kokonaisuutta, (Kuva 12) joista jokaisesta pyrittiin kehittämään nopeampi, helpokäyttöisempi, ominaisuuksiltaan rikkaampi sekä grafiikaltaan miellyttävämpi.

Sovelluksen ensimmäinen versio oli ulkoasultaan karkea koska siinä keskityttiin puhtaasti sen toimivuuteen. Toisessa versiossa ulkoasu alkoi saada nykyistä muotoansa ja sovelluksen logiikka oli lähes valmis. Tämän version kehittämisessä oli eniten haasteita. Kolmas versio vastaa ulkoasullisesti ja ominaisuuksiltaan sovelluksen nykytilaa pieniä julkaisin jälkeisiä korjauksia lukuun ottamatta. Kolmanteen versioon pyrittiin korjaamaan kaikki esille tulleet ongelmat.

Sovellus rakennettiin mobile-first periaatteella, eli se suunniteltiin ensisijaisesti käytettäväksi mobiililaitteilla ja sitten skaalautumaan myös muille laitteille. Sen sijaan että luodaan työpöytä sivusto, ja pakotetaan se sopimaan matkapuhelimeen, aloitetaan pienestä näytöstä. Sen skaalaaminen suuremmille näytöille on huomattavasti helpompaa kuin toisinpäin. Porkkanapankissa mm. elementtien koko ja sijainti, tekstin fonttikoko ja kirjainten välitys sekä käyttäjäruudukon muoto vaihtuu näytön koon vaihtuessa. Porkkanapankki on suunniteltu pääasiallisesti käytettäväksi matkapuhelimelta käsin mutta se toimii myös tableteilla sekä työpöydältä. (Kuva 13)



KUVA 12. Kolme julkaistua pääversiota sovelluksesta, järjestyksessä vanhimmasta (vasen) uusimpaan (oikea)



KUVA 13. Mobiililaitteen ruudun ulottuvuus (UXMovement, anthony 2017)

4.5 Käyttäjätestaus ja julkaisu

Sovelluksen valmistuttua sen viimeisimpään muotoonsa oli aika suorittaa käyttäjätestaus. Tällä pyrittiin paikallistamaan sovelluksessa piilevät virheet ja korjaamaan ne. Käyttäjätestaus suoritettiin sovelluksen valmistuttua nykyiseen muotoonsa.

4.5.1 Käytetyt käyttäjätestauksen muodot

Käyttäjätestaus suoritettiin käytettävyydestestauksen ja UX-testauksen muodossa. Käytettävyydestestauksella tarkoitetaan keskeneräisen työn esittelemistä käyttäjille. Tämän suorittamisella voidaan arvioida asiakaskokemuksen laatua. Sen avulla voidaan arvioida mm. toimiiko sovellus toivotulla tavalla ja kuinka intuitiivinen sovellus on käyttää. Se on myös loistava tapa kerätä palautetta käyttäjiltä. UX-testauksella tarkoitetaan käytettävyydestestausta lähinnä graafisesta näkökulmasta. Tässä testauksen osa-alueessa huomioidaan mm. elementtien asettelua sekä värimaailmaa. Tämä alue tarkastelee sovellusta myös esteettömyyden näkökulmasta. ”Suomeksi puhutaan käyttäjäkokemuksesta tai käyttökokemuksesta (user experience, UX).” ”Käyttäjäkokemus muodostuu tunteista, joita syntyy käytön aikana. Positiivisia tunteita herättää esimerkiksi helppokäyttöinen käyttöliittymä.” (UX/UI design -sanasto tutuksi, 2021)

”On tärkeä pitää mielessä, että käytettävyydestaus ei ole vain virstanpylväs”. ”Ryhmällä tulisi olla tavoite, miksi he testaavat, ja sitten implementoida tulokset”. (Usability.gov Usability testing, 2023) Tehokkaan käytettävyydestin suorittamiseksi laadittiin testisuunnitelma ja etsittiin joukko sopivia osallistujia. Ennen varsinaisen käyttäjätestauksen suorittamista määriteltiin testille tavoite. Tavoitteena oli saada käyttäjiltä palautetta sivuston helppokäyttöisyydestä, värimaailmasta, käyttömukavuudesta sekä ominaisuuksista. Näitä havaintoja ja palautteita oli myöhemmin tarkoitus hyödyntää sovelluksen jatkokehityksessä ja näin myös toimittiin. Kuvassa 14 näkyy kuinka sovelluksen suunnittelija usein arvioi loppukäyttäjän toiminnot väärin.

Testikäyttäjiksi valitaan useimmiten joukko ihmisiä, jotka vastaavat ominaisuuksiltaan, iältään ja taidoiltaan sovelluksen loppukäyttäjiä. Tässä tapauksessa sovelluksen käyttäjän ikä sekä tietotekninen osaaminen voi olla mikä tahansa. Tämän takia testikäyttäjiksi valittiin joukko eri ikäisiä ja tietoteknisiltä taidoiltaan eri tasoisia ihmisiä. Näin voitiin varmistaa sovelluksen helppokäyttöisyys tietoteknisestä taitotasosta ja iästä riippumatta.



KUVA 14. Käytettävyystestaus pähkinäkuoressa.

4.5.2 Testauksen aikana käytetyt menetelmät

Testikäyttäjille annettiin lyhyt selostus sovelluksen tarkoituksesta ja sen mahdollisista kohderyhmistä. Tämän jälkeen testikäyttäjät lähtivät suorittamaan yksi kerrallaan annettuja tehtäviä (Liite 1). Käyttäjille lueteltiin tehtävä, kuten ”vaihda sovelluksen kieli englanniksi” tai ”vaihda väriteema tummaksi”. Testikäyttäjän toimia tarkasteltiin tehtävän suorittamisen aikana. Testin valvoja kirjoitti ylös mistä käyttäjä ensimmäiseksi lähti toimintoa sovelluksesta etsimään. Tämän pohjalta myöhemmin toiminnon sijaintia voidaan arvioida paremmin ja sille voidaan määrittää intuitiivinen ja helppokäyttöinen paikka.

Ääneen ajattelu ja käytetyn ajan mittaaminen

Käyttäjää kehoitettiin ajattelemaan ääneen mahdollisimman paljon tehtävän suorittamisen aikana. Ääneen ajattelu vähentää väärinymmärrysten riskiä testaajan ja testin valvojan välillä sekä auttaa ymmärtämään peruskäyttäjän ajatusproses-

sia sovellusta käyttäessään. Tästä oli merkittävästi hyötyä intuitiivisen ja helppo-käyttöisen käyttöliittymän rakentamisessa. Käyttäjien tehtäviin käytetty aika kellotettiin ja kirjattiin ylös (Kuva 15). Tämän testin avulla saatiin selkeitä tuloksia mitä sovelluksessa täytyy kehittää nopeammaksi ja mitkä asiat ovat nykytilaansa jo tarpeeksi nopeasti tehtävissä. Ajan ottaminen auttoi myös ymmärtämään millä sivuilla käyttäjä viettää eniten aikaa sovellusta navigoidessaan. Käyttäjätestaukseen kului aikaa testattavasta henkilöstä riippuen 30–60 minuuttia jokaista testattavaa henkilöä kohden.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	
96	TESTIKÄYTTÄJÄ 5																			
97	Ikä	47																		
98	Tietotekninen osaaminen (oma arvio)	3 tai 4																		
99	Käytetty laite	Iphone 8																		
100									KUINKA HELPOSTI LÖYTYI	OLIKO INTUITIIVINEN	AIKA	1. OIKEIN ?	AVOIN PALAUTE							
101	1. Kirjautu sisään sähköpostiin tulleilla tunnuksilla	5	5	~ 50	X															
102	2. Seuraa opastettu ryhmän luonti loppuun asti	5	4	~ 120	X	1. ja 2. elementin transitio epäselvä ja														
103	3. Vaihda kieli englanniksi ja värimaailma tummaksi (vaihda takaisin jos haluat)	5	5	> 10	X															
104	4. Vaihda salasana	5	5	~ 30	X															
105	5. Lisää ryhmään uusi käyttäjä ilman sähköpostia	3	4	~ 30		Epäselvää että näin voi tehdä jos ei ti														
106	6. Tee uudesta käyttäjästä ylläpitäjä	5	5	> 10	X															
107	7. Lisää käyttäjälle tehdyksi yksi tehtävä x50	5	5	~ 40	X															
108	8. Lisää käyttäjälle lunastetuksi yksi palkinto	5	2	~ 30	X	Ohje mitä porkkanan lisääminen teht														
109	9. Lisää ryhmään uusi käyttäjä (Minä)																			
110	10. Hyväksy lisäämä tehtäväni	5	5	~ 20	X															
111	11. Poista tehty tehtävä käyttäjältä	3	5	> 10		Käyttäjän poistaminen vahingossa liia														
112	12. Poista käyttäjä	5	5	> 10	X															
113	13. Lisää itsellesi profiilikuva	4	5	~ 50	X	pfp placeholderiksi esim. Kamera tms														
114	14. Poista profiilikuvasi	5	5	> 10	X															
115	15. Lisää uusi tehtävä ja/tai palkinto ryhmään	5	5	> 10	X	Agreement sivun lisää tehtävä tai palk														
116	16. Muokkaa yhtä tehtävää tai palkintoa	5	5	~ 30	X															
117	17. Tee uusi ryhmä	5	5	> 10*	löytö	X														
118	18. Muokkaa ryhmän nimeä	4	4	~ 20		Kynä iconi näyttää liikaa samalta kuin														
119	19. Kirjautu ulos	5	5	> 10	X															
120																				
121																				

KUVA 15. Yhden testikäyttäjän käyttäjätestauksen tuloksia Excel taulukossa.

Avoim palaute

Käyttäjältä kysyttiin jokaisen tehtävän suorittamisen jälkeen avoin palaute tehtävästä. Käyttäjältä kysyttiin, kuinka helppo tehtävä oli suorittaa ja kuinka intuitiiviseksi hän arvioisi tehtävän suorittamiseen vaadittujen elementtien sijainnin sovelluksessa. Käyttäjät arvioivat nämä asiat asteikolla 1–5. 1 on heikoin ja 5 paras arvosana. Käyttäjää haastateltiin lopuksi sovelluksen käytöstä kokonaisuutena ja näitä havaintoja sekä palautteita otettiin huomioon sovelluksen jatkokehityksessä. Käyttäjätestausta helpotti sovelluksesta jo työn alkumetreillä julkaistusta versiosta oikeilta asiakkailta kerääntynyt palaute, jota hyödynnettiin työn kehityksen aikana.

Käyttäjätestauksella saatiin sovelluksesta paljon sekä positiivista sekä negatiivista palautetta. Näitä palautteita käytettiin hyödyksi sovelluksen jatkokehityksessä. Ilman käyttäjätestausta moni testauksen aikana esiin tulleista ongelmista olisi varmasti jäänyt huomiotta mikä olisi selvästi heikentänyt lopullisen sovelluksen laatua.

5 POHDINTA

Seuraavissa kappaleissa on käyty läpi opinnäytetyön lopullinen tuotos, työn kriiteereitä ja kehittämisehdotuksia. Lisäksi pohdin työn aikana tapahtunutta oppimista sekä yhteenvetoa tärkeimmistä asioista. Viimeiseksi muutamia asioita, joita sovellukseen voisi tulevaisuudessa lisätä tai joita voisi muuttaa.

5.1 Tulokset

Opinnäytetyön tarkoituksena oli tuottaa yritykselle henkilökunnan motivoimiseen ja tehtyjen toimien kirjaamiseen tarkoitettu sovellus nimeltään Porkkanapankki. Tuloksena syntyi mobiililaitteille suunniteltu verkkosovellus, joka täyttää asetetut oletukset ja palvelee hyvin tarkoitustaan. Opinnäytetyön aihe oli melko laaja. Se käsitteli verkkosovelluksen kehityksen eri vaiheita, sovelluskehitystä Bubble.io -ohjelmistolla, Web-kehitystä yleisesti, sekä koodittomien alustojen ja perinteisten alustojen eroja.

Opinnäytetyön kirjoittaminen ja Porkkanapankin kehittäminen olivat hyvin opettavaisia. Ne kehittivät tietokantojen parissa tarvittavia taitoja, graafisia taitoja sekä ongelmanratkaisukykyä. Sovelluksen kehitys auttoi ymmärtämään paremmin koodittomien alustojen toimintaperiaatetta. Opinnäytetyön kirjoittaminen auttoi kehittämään kirjoitus- ja tiedonhakutaitoja. Oppimispäiväkirjan kirjoittaminen työn aikana helpotti aikatauluttamista, tehtyjen muutosten esittämistä, sekä työn kehityksen seuraamista.

Sovelluksen tilaajan kanssa keskusteleminen sekä palaverit opettivat paljon tärkeitä työelämätaitoja. Aikataulutukset sekä tuotettujen tuloksien esittäminen tehokkaasti oli tärkeää. Varsinkin työn alkuaikoina ketterä kehitysmalli tuli tutuksi. Sovelluskehityksen aikana kestävien ja uudelleenkäytettävien ratkaisujen käytön merkitys korostui. Sovelluksen rakentaminen ja dokumentointi niin että toinen ihminen voi tarvittaessa jatkaa sovelluksen kehitystä ja käyttää hyödykseen valmiita rakennuspalikoita oli tärkeää ja tätä pyrittiin ylläpitämään koko kehitysprosessin ajan.

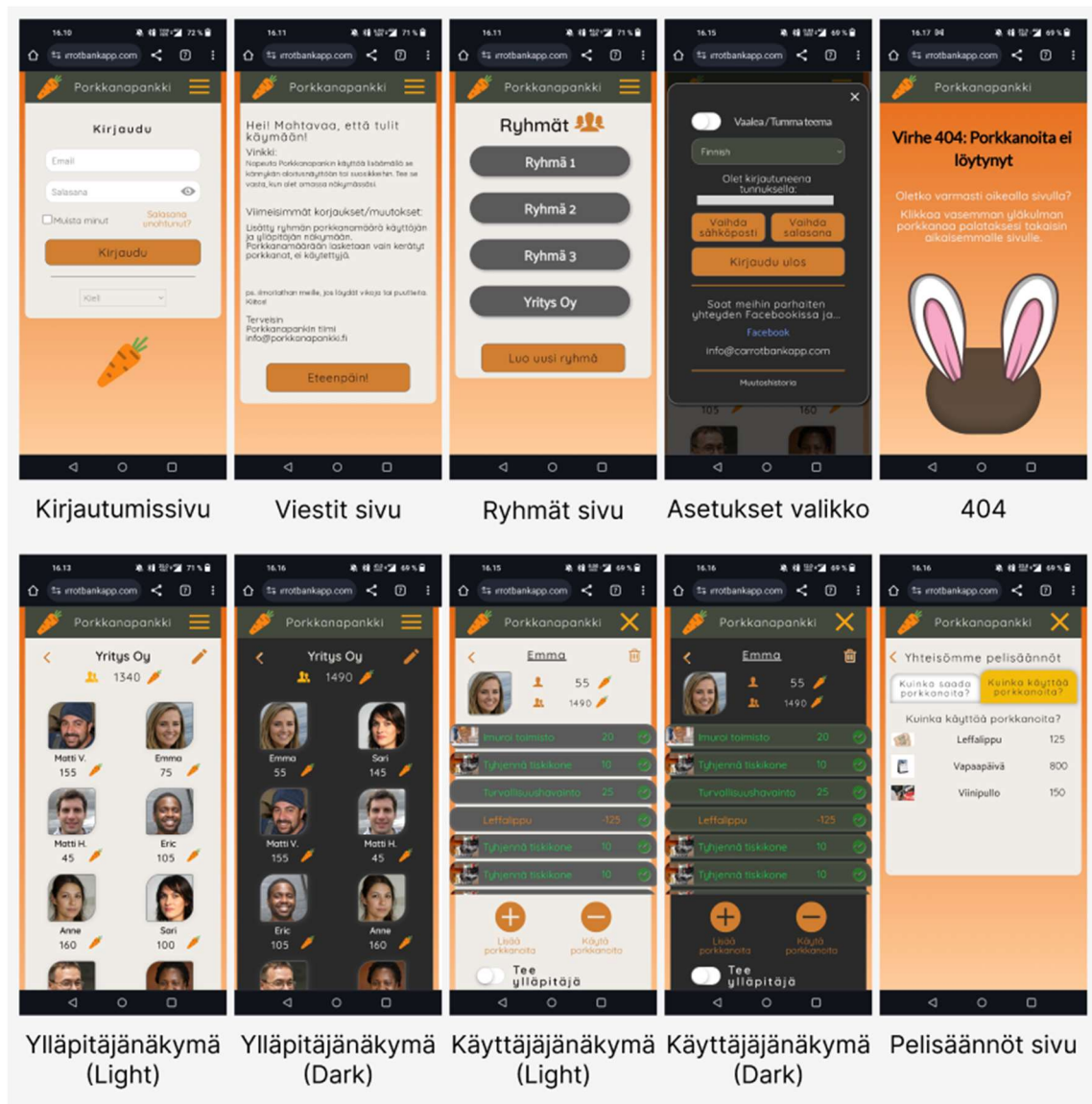
Kehityksen loppupuolella käyttäjätestauksen merkityksellisyys korostui virheiden havaitsemisessa ja palautteen saamisessa. Monet sovelluksessa piilevät ongelmat olisivat jääneet varmasti havaitsematta kehityksen aikana ilman käyttäjätestausta. Käyttäjätestaus myös auttoi ymmärtämään kuinka keskivertokäyttäjä käyttää kehitettyä sovellusta. Sovelluksen kääntäminen englanniksi auttoi kehittämään kielitaitoja ja kasvatti ammattisanastoa. Lisää tietoa Porkkanapankista löytyy osoitteesta www.porkkanapankki.fi. Porkkanapankki on opinnäytetyön kirjoitushetkellä käytössä useammalla eri yrityksellä.

5.2 Yhteenveto

Bubble on tehokas alusta verkkosovellusten kehittämiseen ja sen käyttöliittymä on helppo oppia. Bubble virtaviivaistaa kehitysprosessia ja säästää siten aikaa. Bubblen sisäänrakennetut ominaisuudet kuten tietokannanhallintapaneeli, responsiivinen suunnittelu ja laaja lisäosakirjasto tekevät verkkosovellusten rakentamisesta vaivatonta. Porkkanapankkia kehitettäessä ei tullut vastaan sellaista ongelmaa, jota ei Bubblen sisäänrakennetuilla ominaisuuksilla tai lisäosilla olisi voinut ratkaista. Bubblen tekoälyintegraatio vaikuttaa erittäin hyödylliseltä ja tulevaisuudessa laajalti käytetyltä ominaisuudelta. Koodittomien alustojen helppokäyttöisyys ja laajat mahdollisuudet olivat positiivinen yllätys. Projektia aloittaessa vallitsi kuvitelma koodittomien alustojen olevan hyvinkin rajoittuneita ominaisuuksiltaan, kuitenkin kehityksen aikana tämä kuvitelma osoittautui vääräksi. Sovellusta kehitettäessä havaittiin kuitenkin Bubblen haastavuuksien kasvavan ja kehityksen hidastuvan sovelluksen koon ja logiikan kasvaessa.

5.3 Mahdollisia tulevaisuuden toimia

Lähes kaikki projektin alkuvaiheessa suunnitellut ominaisuudet saatiin rakennettua Porkkanapankkiin. Sovelluksen grafiikka kehittyi huomattavasti parempaan suuntaan. (Kuva 16) Jos sovellusta lähdetään jatkokehittämään voisi painopiste keskittyä sovelluksen pelillistämiseen eli siihen, kuinka sovelluksesta saadaan houkuttelevampi käyttää. Tämä voidaan toteuttaa mm. erilaisten animaatioiden ja kannustimien avulla.



KUVA 16. Porkkanapankin lopullinen verkkosovellus puhelimella käytettynä

LÄHTEET

UX Academy. 2021. UX/UI design -sanasto tutuksi. Julkaistu 21.12.2023. Viitattu 10.10.2023

<https://www.uxacademy.fi/ux-ui-design-sanasto-tutuksi/>

Usability.gov. 2023. Usability testing. Viitattu 10.10.2023

<https://www.usability.gov/how-to-and-tools/methods/usability-testing.html>

Bubble blog 2023. Why AI + No-Code is the Future. Julkaistu 27.3.2023. Viitattu 25.10.2023.

<https://bubble.io/blog/bubble-ai-no-code/>

College of Arts and Technology. Web Development in a Nutshell and Why You Should Study it. Viitattu 25.3.2023

<https://www.ciit.edu.ph/web-development-in-a-nutshell-and-why-you-should-study-it/>

Lambdatest 2023. 34 Web Development Trends In 2023: Embracing The Future. Julkaistu 9.5.2023. Viitattu 20.1.2023

<https://www.lambdatest.com/blog/web-development-trends/>

Medium 2023. The Rise of No-Code Platforms: Will Web Development Soon Be a Thing of the Past? Julkaistu 19.6.2023. Viitattu 8.12.2023

<https://medium.com/@prashanth.colourmoon/the-rise-of-no-code-platforms-will-web-development-soon-be-a-thing-of-the-past-a281fc393e9d>

Bubble manual 2023. How to use bubble manual effectively. Julkaistu 10 / 2023. Luettu 24.10.2023

<https://manual.bubble.io/>

Bubble videos 2020. The Bubble Fundamentals. Julkaistu 23.9.2020. Luettu 16.9.2023

<https://bubble.io/video/what-you-can-build>

Ameliesolutions 2023. What is Bubble? Luettu 26.9.2023

<https://www.amliesolutions.com/what-is-bubble-io/>

Computerworld 2015. Bubble, Bubble, ending the developer struggle? Julkaistu 22.12.2015. Luettu 13.10.2023

<https://www.computerworld.com/article/2982331/bubble-bubble-ending-the-developer-struggle.html>

Medium 2021. The Good and Bad of Bubble.io. Julkaistu 16.5.2021. Luettu 13.10.2023

<https://medium.com/nerd-for-tech/the-good-and-bad-of-bubble-io-37f88a700019>

Airdev 2023. Full code vs. no-code: When should you use no-code over traditional development? Luettu 2.10.2023

<https://www.airdev.co/post/full-code-vs-no-code-when-to-build-bubble-app>

Medium 2023. 15 Best No-Code Platforms to Build Amazing Apps in 2023. Julkaistu 3 / 2023. Luettu 4.11.2023

<https://medium.com/geekculture/15-best-no-code-platforms-to-build-amazing-apps-in-2023-874fa969822b>

Coaching No-Code Apps. 2022. Using Plugins on Bubble (Avoid a Poorly Performing App). Julkaistu 27.4.2022. Luettu 20.9.2023

<https://coachingnocodeapps.com/using-plugins-in-bubble>

Lambdatest. 2023. A complete Guide To Mobile First Design. Julkaistu 23.3.2023. Luettu 16.11.2023

<https://www.lambdatest.com/blog/mobile-first-design/>

Zdnet. 2021. Survey: Low-code and no-code platform usage increases. Julkaistu 1.7.2021. Luettu 6.11.2023

<https://www.zdnet.com/article/survey-low-code-and-no-code-platform-usage-increases/>

Bubble blog. 2023. An Introductory Guide to Visual Programming (2023). Julkaistu 2023. Luettu 15.9.2023

<https://bubble.io/blog/visual-programming/>

W3.org. 2023. Web Content Accessibility Guidelines (WCAG) 2.1. Julkaistu 21.9.2023. Luettu 5.10.2023

<https://www.w3.org/TR/WCAG21/>

HAMK 2020. Paras ohjelmointikieli ohjelmoinnin opiskelun aloittamiseen. Julkaistu 17.4.2020. Luettu 11.12.2023

<https://unlimited.hamk.fi/yrittajyys-ja-liiketoiminta/paras-ohjelmointikieli-ohjelmoinnin-opiskelun-aloittamiseen/>

Porkkanapankki-verkkosovelluksen kotisivut

<https://www.porkkanapankki.fi/>

KUVA 11. Kontrastitestaaja

<https://webaim.org/resources/contrastchecker/>

KUVA 12.

<https://uxmovement.com/mobile/why-mobile-menus-belong-at-the-bottom-of-the-screen/>

LIITTEET

Liite 1. Käyttäjätestauksen tehtävät

1. Kirjaudu sisään sähköpostiin tulleilla tunnuksilla
2. Seuraa opastettu ryhmän luonti loppuun asti
3. Vaihda kieli englanniksi ja värimaailma tummaksi (vaihda [takaisin](#) jos haluat)
4. Vaihda salasana
5. Lisää ryhmään uusi jäsen ILMAN sähköpostia
6. Tee uudesta käyttäjästä ylläpitäjä
7. Lisää käyttäjälle tehdyksi yksi tehtävä x50
8. Lisää käyttäjälle lunastetuksi yksi palkinto
9. Lisää ryhmään uusi käyttäjä (Testin valvoja)
10. Hyväksy lisätty tehtävä
11. Poista tehty tehtävä käyttäjältä
12. Poista käyttäjä
13. Lisää itsellesi profiilikuva
14. Poista profiilikuvasi
15. Lisää uusi tehtävä ja/tai palkinto ryhmään
16. Muokkaa yhtä tehtävää tai palkintoa
17. Tee uusi ryhmä
18. Muokkaa ryhmän nimeä
19. Kirjaudu ulos