



Verkkosovelluksen kehittäminen Keski-Uudenmaan pelastuslaitokselle

Valtteri Saariaro

2020 Laurea



Laurea-ammattikorkeakoulu

Verkkosovelluksen kehittäminen Keski- Uudenmaan pelastuslaitokselle

Valtteri Saariaro
Tietojenkäsittely
Opinnäytetyö
Toukokuu, 2020

Valtteri Saariaro

Verkkosovelluksen kehittäminen Keski-Uudenmaan pelastuslaitokselle

Vuosi

2020

Sivumäärä

34

Opinnäytetyön tavoitteena oli kehittää Keski-Uudenmaan pelastuslaitokselle arjen turvallisuuden liittyvä verkkosovellus. Sen kehityksessä hyödynnettiin pelillistämisen keinoja, koska kohderyhmänä olivat 10-vuotiaat lapset. Tärkeimmät vaatimukset sovellukselle olivat käytettävyys ja saavutettavuus.

Kehittämistyössä perehdyttiin selainpohjaisen sovelluksen kehittämisen eri vaiheisiin vaatimusmäärittelystä valmiin sovelluksen julkaisuun. Sovellus toteutettiin JavaScript- ja HTML-ohjelmointikielillä. Teoriaosuudessa käsitellään sovelluskehityksen eri vaiheet sekä pelillistämisen ja saavutettavuuden periaatteet. Laadullisina tutkimusmenetelminä käytettiin haastatteluja sekä kyselyä.

Kehittämistyön tuloksena syntyi valmis selainpohjainen sovellus, jossa käyttäjät voivat testata tietouttaan arjen turvallisuudesta. Sovelluksen toiminnallisuuksia ovat aihealueen valinta ja siihen liittyviin kysymyksiin vastaaminen. Sovellus antaa käyttäjälle jatkuvaa palautetta edistymisestä ja lopullisen palautteen sen mukaan, kuinka moneen kysymykseen käyttäjä on vastannut oikein. Opinnäytetyön lopussa esitellään konkreettiset ehdotukset, joilla sovellusta voisi kehittää edelleen.

Valtteri Saariaro

Web application for Keski-Uusimaa Fire Department

Year 2020

Pages

34

The purpose of this thesis was to develop a web application on everyday safety for the Keski-Uusimaa Fire Department. Since ten-year-old children were the target group of the application, it also needed to use elements of gamification to motivate the user. The main requirements for the application were usability and accessibility.

The development took a practical approach on creating a browser-based web application all the way from the initial requirement analysis to the finished product. HTML and JavaScript were used as programming languages. The theoretical framework of the thesis concentrates on the process of application development and explains the concepts of gamification and accessibility. Interviews and inquiries were chosen as the qualitative methods for this thesis.

In conclusion, a functioning browser-based application was delivered to the Keski-Uusimaa Fire Department. In the app, users can test their knowledge on everyday safety by answering questions. The functionalities include topic selection and answering the questions related to the selected topic. The user also receives feedback from the app along the way and a final grade when finishing the questions. Finally, the thesis lists concrete examples for the client on how to further develop and improve the app.

Keywords: Web application, gamification, accessibility, development

Sisällys

1	Johdanto.....	6
2	Työn lähtökohdat.....	6
2.1	Kehittämiskohteen kuvaus ja kehittämistavoitteet	7
2.2	Toimeksiantajan esittely	7
2.3	Tutkimuskysymykset ja aihealueen rajaus	7
2.4	Keskeiset käsitteet.....	7
3	Verkkosovelluksen kehittämisprosessi.....	8
3.1	Scrum kehitysmenetelmänä	10
3.2	Ohjelmistotestaus	10
3.3	Responsiivisuus	11
3.4	Saavutettavuus	11
4	Pelillistäminen	12
4.1	Pelaajatyypit.....	13
4.2	Pelillistämisen keinoja	14
5	Tutkimusmenetelmät	15
5.1	Haastattelu ja kysely	15
5.2	Reliabiliteetti ja validiteetti	16
5.3	Vertailuanalyysi	17
6	Kehittämistyön toteutus.....	17
6.1	Sovelluskehitysprosessin tiedonkeruu	18
6.2	Vaatusmäärittely	19
6.3	Suunnitteluvaihe vaatimusten mukaisesti	20
6.4	Sovelluksen näkymät.....	20
6.5	Sovelluksen toiminnallisuudet.....	23
6.6	Pelillistämisen keinot.....	24
6.7	Sovelluksen testaus	25
7	Kehittämistyön tulokset	25
8	Yhteenveto ja johtopäätökset.....	26
9	Jatkokehitysehdotukset	26
10	Oman oppimisen arviointi	27
	Lähteet.....	29
	Kuviot	31
	Liitteet	32

1 Johdanto

Keski-Uudenmaan pelastuslaitos suorittaa pelastuslain (379/2011) sekä valvontasuunnitelman mukaisia valvontakäyntejä eri kiinteistöissä. Kouluihin ja päiväkoteihin suoritetaan valvontakäyntejä, joiden yhteydessä pelastuslaitos valistaa eri ikäluokkia arjen turvallisuudesta. Taus-talla on elinkaariajattelu, jossa lapset ja nuoret pyritään tavoittamaan mahdollisimman monta kertaa, jotta turvallisuusajattelua ja -osaamista saadaan iskostettua lapsille sekä nuorille varhaisesta vaiheesta lähes aikuisuuteen saakka. Valistuksen kohderyhmänä ovat eskari-laiset, 4-luokkalaiset, 8-luokkalaiset sekä lukiolaiset. Pelastuslaitoksen suorittaman valistuk-sen tarkoituksena on saada muun muassa lapset ja nuoren kiinnostumaan turvallisuudesta, li-sätä heidän osaamistaan sekä esitellä pelastuslaitoksen toimintaa. Eri ikäryhmiä pyritään mo-tivoimaan eri tavoin, joista hyvinä esimerkkeinä toimivat paloautoihin tutustumiset sekä 8-luokkalaisille suunnattu NouHätä! -kampanja. Tuossa kampanjassa nuoret voivat osallistua turvallisuuskoulutukseen ja halutessaan osallistua valtakunnalliseen kilpailuun koulutuksen ai-heista.

Tämän opinnäytetyön tarkoituksena on kehittää verkkosovellus arjen turvallisuudesta, jonka kohderyhmänä ovat 4-luokkalaiset. Sovellusta on mahdollista hyödyntää pelastuslaitoksen suo-rittamassa valistuksessa. Tämä opinnäytetyö käsittelee verkkosovelluksen kehittämistä sen eri vaiheissa. Lisäksi käsitellään sitä, kuinka sovelluksessa voidaan hyödyntää pelillistämisen kei-noja käyttäjien motivoimiseen ja sen kautta oppimisprosessin kehittämiseen.

Teoriaosuudessa käsitellään verkkosovelluksen kehittämistä, pelillistämistä ja pelillistämisen hyödyntämistä sovelluksen käytössä. Kehittämistyössä sovelluksen kehittäminen toteutetaan JavaScript- sekä HTML-ohjelmointikieliä hyödyntäen. Kehittämistyön laadullisina tutkimus-menetelminä käytetään haastatteluja sekä kyselyjä.

2 Työn lähtökohdat

Opinnäytetyön lähtökohtana oli luoda toimeksiantajalle verkkopohjainen sovellus arjen tur-vallisuudesta, jossa käyttäjät voivat testata omaa osaamistaan aihealueeseen liittyen. Kehi-tettävän sovelluksen kohderyhmänä olivat 4-luokkalaiset oppilaat, jotka olivat osallistuneet toimeksiantajan järjestämään valistukseen. Toimeksiantajalla ei ollut ennestään käytössä vas-taavaan tarkoitukseen sovellusta, joten kaikki suunnittelu piti tehdä puhtaalta pöydältä.

2.1 Kehittämiskohteen kuvaus ja kehittämistavoitteet

Tarkoituksena oli kehittää verkkopohjainen sovellus, jota on tarkoitus hyödyntää Keski-Uudenmaan pelastuslaitoksen järjestämän valistuksen tukena. Sovelluksessa tarkoituksena oli vastata arjen turvallisuuteen liittyviin kysymyksiin eri aihealueista, joita on käyty lävitse valistuksen aikana. Sovelluksen kohderyhmän oli tarkasti määritelty tiettyyn ikäryhmään, joten sovelluksen kehityksessä otetaan huomioon kohderyhmän saavutettavuus.

2.2 Toimeksiantajan esittely

Keski-Uudenmaan pelastuslaitos on yksi Suomen 22:sta alueellisesta pelastuslaitoksesta. Keski-Uudenmaan pelastuslaitos on hallinnollisesti Vantaan kaupungin alainen kunnallinen liikelaitos. Keski-Uudenmaan pelastuslaitoksen toimii kahdeksan kunnan alueella, jotka ovat Vantaa, Kerava, Järvenpää, Tuusula, Hyvinkää, Nurmijärvi, Mäntsälä sekä Pornainen. (Keski-Uudenmaan pelastuslaitos 2020.)

Keski-Uudenmaan pelastuslaitos palvelee kyseisiä kuntia sekä niiden asukkaita pelastuslaitokselle määriteltyjen tehtävien mukaisesti. Pelastuslaitoksen tehtäviin kuuluu muun muassa huolehtia pelastusviranomaisille kuuluvasta onnettomuuksien ehkäisystä sekä palotarkastuksista, huolehtia pelastustoimen alaan kuuluvasta valistuksesta ja neuvonnasta sekä huolehtia pelastustoimintaan kuuluvista pelastus- ja onnettomuustehtävistä. (Keski-Uudenmaan pelastuslaitos 2020.)

2.3 Tutkimuskysymykset ja aihealueen rajaus

Kehittämistyön tulisi vastata kysymyksiin: Miten verkkopohjainen sovellus voidaan kehittää? Miten kohderyhmä ja saavutettavuus huomioidaan sovelluksen toteutuksessa? Miten pelillistämistä voidaan hyödyntää sovelluksen käytössä?

Tässä kehitystyössä aihealue on rajattu verkkopohjaisen sovelluksen kehitysprojektiin. Rajauksen ulkopuolelle on jätetty verkkosovelluksen julkaisu sekä valmiin sovelluksen ylläpitoon liittyvät asiat.

2.4 Keskeiset käsitteet

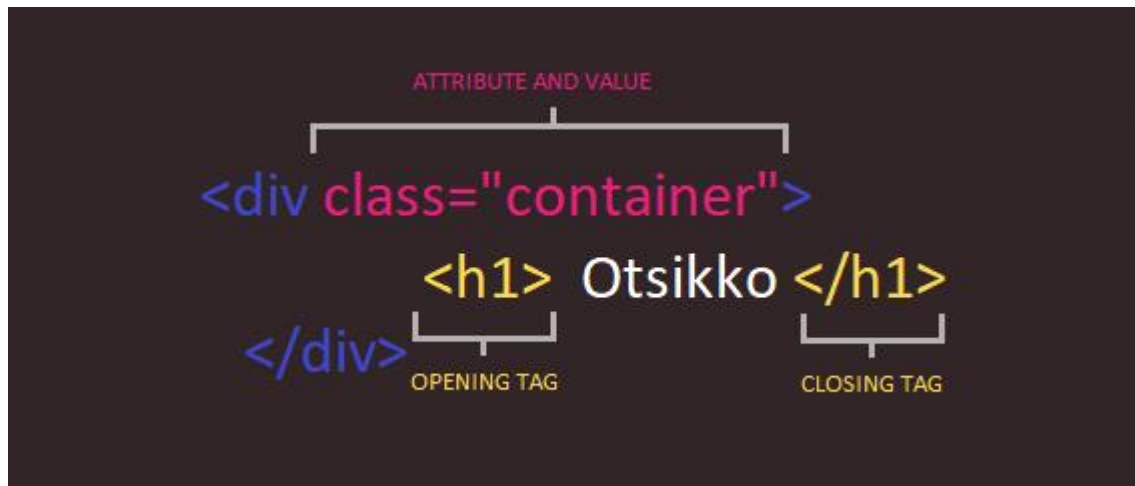
Notepad++	Avoimeen lähdekoodiin pohjautuva tekstieditori ja lähdekoodieditori Windowsille ja se perustuu vapaaseen Scintilla-komponenttiin
Gimp	Avoimeen lähdekoodiin perustuva monipuolinen kuvankäsittelyohjelma, joka on tarkoitettu rasterikuvien muokkaamiseen.

HTML	Kieli, jolla useimmat verkkosivut ovat tehty. Sitä käytetään luomaan sivuja ja niille staattista sisältöä, kuten tekstiä ja linkkejä, sekä liittämään niihin kuvia.
CSS	Kieli, jolla määritellään miten HTML-elementit esitetään. Kielen lyhenne tulee sanoista Cascading Style Sheets.
JavaScript	Ohjelmointikieli, jota käytetään verkko-ohjelmoinnissa. Sen avulla voidaan manipuloida HTML- ja CSS -koodia, laskea, luoda ehtolauseita ja funktiota. Javascriptin avulla voidaan luoda sivuille toiminnallista ja dynaamista sisältöä.
GoogleDrive	Googlen tarjoama pilvipalvelu, jota voi käyttää tietokoneelta, selaimelta tai mobiililaitteilta.
Microsoft Azure	Microsoftin julkinen pilvipalvelu. Azurea voidaan käyttää sekä virtuaalipalvelinten alustana (Infrastructure as a Service, IaaS) että kehittäjille tarkoitettuna kehitysalustana (Platform as a Service, PaaS)

3 Verkkosovelluksen kehittämisprosessi

Ensimmäinen vaihe verkkosivujen tai sovellusten toteutuksessa on miettiä, mikä on sen kohderyhmä. Kohderyhmä määrittää sivun tai sovelluksen tarpeen. Jokaisen eri kohderyhmän tarpeet tulee ottaa huomioon suunnitteluvaiheessa. Kun kohderyhmä on tiedossa, voidaan suunnitella, mitä sivun tai sovelluksen on tarkoitus tehdä. Jos sivulla on tarkoitus vain esitellä staattisia elementtejä, kuten kuvia ja tekstejä, riittää toteutustavaksi yksinkertaisimmillaan HTML ja visuaaliseen määrittelyyn CSS. Näillä ohjelmointikielillä pystyy toteuttamaan staattiset sivut, jota pystyy selaamaan verkkoselaimen kautta. (Beighley & Morrison 2009, 2.)

HTML määrittelee sivun rakenteen ja lisäksi sillä voidaan määritellä tekstien tyyliä, kuten esimerkiksi mikä on otsikkoa ja mikä leipätekstiä. Näitä erilaisia rakenteita käytetään käyttämällä erilaisia tageja, eli tunnisteita, jotka ovat sijoitettu hakasulkumerkkien väliin. Avaava ja sulkeva tunniste ovat hyvin samankaltaisia ja ne eroavat toisistaan vain kauttaviivalla sulkevan tunnisteiden kohdalla. HTML-tunnisteilla voidaan myös määritellä attribuutteja, jotka sijoitetaan avaavaan tunnisteeseen. Attribuuttien on tarkoitus antaa lisätietoa tunnisteiden sisällyttämisestä elementistä. Tätä voidaan käyttää esimerkiksi määrittämään luokan nimen. HTML-tunnisteiden rakennetta on kuvattu kuviossa 1. (Duckett 2011, 20-26.)



Kuvio 1: HTML-tunnisteen rakenne

CSS:ää käytetään määrittelemään, miten HTML:n avulla kirjoitetut elementit esitetään. CSS:n avulla voidaan luoda määritelmiä, jotka kertovat miltä eri elementit näyttävät. Sen avulla voidaan esimerkiksi määritellä sivun elementin näkyvyyttä, tekstien kokoa, väriä tai fonttia sekä minkä värinen sivun tausta on. CSS:ään voidaan kirjoittaa suoraan HTML-tiedoston tyyli-tunnisteiden sisälle, mutta suositeltavampaa on kuitenkin käyttää erillistä omaa tiedostoaan. Tätä tiedostoa kutsutaan HTML-tiedostosta. CSS on rakenteeltaan yksinkertainen, sillä se sisältää valitsijan (selector) ja sille määritelmän (declaration). Määritelmiä voi kuitenkin olla useampia ja ne sijoitetaan aaltosulkujen sisäpuolelle. Määritelmä koostuu kahdesta osasta, ominaisuudesta (property) ja sen arvosta (value). Ominaisuus kuvaa mitä asiaa halutaan määritellä, kuten väriä, fonttia, leveyttä tai korkeutta, kun taas arvo määrittelee tämän. CSS-säännön rakennetta on avattu kuviossa 2. (Duckett 2011, 227-232).



Kuvio 2: CSS-säännön rakenne

JavaScriptin avulla sivustolle voidaan luoda dynaamisia toiminnollisuuksia. JavaScriptin avulla voidaan esimerkiksi määritellä erilaisia suoritettavia funktioita tai manipuloida CSS tyylimäärittelyksiä. CSS:n tavoin JavaScriptiä voidaan kirjoittaa suoraan HTML-tiedostoon, mutta suositeltavaa olisi tuoda JavaScript ulkopolisesta tiedostosta. Javascriptin oliomallin perusyksiköjä ovat objekti- ja funktio-oliot. (Diaconescu & Wagner 2017, 14-17).

3.1 Scrum kehitysmenetelmä

Scrum on projektinhallinnan viitekehys, jota käytetään yleisesti ketterässä ohjelmistokehityksessä. Vaikka Scrum on kehitetty erityisesti ohjelmistoprojektien hallintaan, sitä voidaan soveltaa myös yleisesti projektinhallinnassa. Scrum-mallissa ohjelmiston tai sovelluksen toteutus on jaettu sprintteihin. Nämä sprintit ovat yhdestä viikosta kuukauteen kestäviä työskentely jaksoja, joiden aikana kehitetään kullekin sprintille määrätyt vaiheet. Sprintteihin kuuluu myös kehitettyjen ominaisuuksien testaaminen. Scrum-mallin yksi tärkeimmistä asioista, on tiimin sisäinen kommunikointi. Scrum-mallin yksi keskeinen osa on lyhyet päivittäiset tapaamiset, jolloin kaikki ovat kartalla, mitä kukin on tehnyt. (Kasurinen 2013.)

Scrum-mallissa tiimit ovat itseohjautuvia ja ketteriä, eikä tiimissä ole turhia hierarkkisia esteitä. Tiimin sisällä on määritelty ainoastaan scrum master sekä tuoteomistaja. Scrum masterin tarkoituksena on olla niin sanottu mahdollistaja, eli hänen tärkein tehtävänsä on poistaa kehitystiimiltä kehittämistä vaikeuttavia esteitä. Hän siis pyrkii varmistamaan, että kehitystiimi työskentelee mahdollisimman tehokkaasti menetelmää noudattaen. Tuoteomistaja on taas vastuussa tuotteen back logista, eli kehitysjonosta. Hänen tärkein tehtävänsä on siis varmistaa, että kehitystiimi tietää mitä heidän tulee seuraavaksi tehdä. (Schwaber & Sutherland 2017, 15.)

Scrum-projektit pyöriivät kehitysjonon ympärillä, jossa on listattuna kaikki ominaisuudet, toiminnot, vaatimukset sekä korjaukset, jotka tulee tehdä. Kehitysjono ei kuitenkaan ole lukkoon lyöty, vaan se voi muuttua ja mukautua projektin edetessä. Tuotteen kehitysjonosta muodostetaan jokaiselle sprintille oma kehitysjono, eli sprint backlog, joka sisältää kaikki kohdat, jotka tulee kehittää kyseisen sprintin aikana. (Schwaber & Sutherland 2017, 15.)

3.2 Ohjelmistotestaus

Testaamalla kehitettyä sovellusta tai ohjelmistoa pyritään varmistamaan se, että kaikki niiden eri osa-alueet ja toiminnot toimivat oikein. Lisäksi testaamalla pyritään varmistamaan se, että kehitetty palvelu on tavoitellun mukainen. Toisin sanoen, testaamalla varmistetaan, että kehitetty tuote toimii ja että se on tehty oikein. (Ohjelmistotestauksen käsikirja 2013.)

Ohjelmistokehitysprosessin testaaminen voidaan suorittaa usealla eri tavalla. Jos kehityksessä käytetään vesiputousmallia, jossa tarkoituksena on edetä aina työvaiheesta seuraavaan, niin kaikki testaus suoritetaan, kun ohjelmisto on valmis. Kaikki testaus tehdään juuri ennen kuin siirrytään ylläpitovaiheeseen. Vesiputousmallissa siis kaikki testaus tapahtuu yhdessä vaiheessa, kun kaikki kehitysvaiheet ovat suoritettu. Tämän takia testausprosessi olisi parempi suorittaa noudattamalla esimerkiksi V-mallia, jossa testaus ja ohjelmiston kehittäminen hoidetaan yhdessä koko ohjelmiston kehitysprosessin ajan. Tähän malliin kuuluvat hyväksymistestaus, järjestelmätestaus, integraatiotestaus sekä yksikkötestaus. Näillä kaikilla eri testausprosesseilla on omat käyttötarkoituksensa. Esimerkiksi ohjelmistolle määritettyjä vaatimuksia testataan hyväksymistestauksen avulla ja ohjelmointia testataan yksikkötestauksen avulla. (Kasurinen 2013.)

3.3 Responsiivisuus

Koska verkkosivujen selaaminen mobiililaitteilla on lisääntynyt, myös verkkosivujen suunnittelussa tulee ottaa huomioon eri kokoiset laitteet, joilla sivua käytetään. Tietokoneen näytölle suunniteltujen sivustojen selaaminen mobiililaitteilla ei ole kovin mielekästä ja pahimmassa tapauksessa niiden käyttö voi olla jopa turhauttavaa. Tästä syystä mobiiliversioiden suunnittelu verkkosivujen suunnittelun yhteydessä on yleistynyt ja lähes jokaiselta nykyaikaiselta verkkosivulta löytyy mobiiliversio. (Hong 2018, 11-13.)

Ensimmäiset ratkaisut mobiililaitteiden tuomaan ongelmaan oli tehdä kaksi eri versiota verkkosivusta. Tietokoneelle tehtiin omansa ja mobiililaitteille omansa. Tämä tietenkin tarkoitti kahden erillisen sivuston ylläpitoa, joka on työlästä. CSS3 -version tuomat Media Queryt, eli ehdolliset CSS-määreet, toivat helpotuksen tähän ongelmaan. Näitä hyödyntämällä, pystyttiin kehittämään samasta sivustosta eri näkymiä sen mukaan, minkä kokoisella laitteella sivustoa käytetään. Ehdollisten CSS-määreiden avulla sivustosta pystytään määrittelemään erilaisia variaatioita esimerkiksi ruudun resoluution ja koon mukaan. (Hong 2018, 16-18.)

Toinen vaihtoehtoinen tapa huomioida mobiililaitteet sivustojen suunnittelussa on Mobile first ajattelutapa. Tätä noudattamalla sivustosta suunnitellaan ensiksi vain välttämättömimmät ominaisuudet ja nämä ominaisuudet suunnitellaan mobiililaitteille. Kun sivusto on suunniteltu valmiiksi mobiililaitteille, vasta sen jälkeen se viedään työpöytäversioksi. Näin pystytään varmistamaan, että sivusto on suunniteltu toimimaan kaiken kokoisilla laitteilla. (Hong 2018, 16-18.)

3.4 Saavutettavuus

Käytettävyydellä kuvataan sitä, miten tietty käyttäjäjoukko pystyy käyttämään tuotetta sille ennalta määrätyn tavoitteen saavuttamiseksi. Lisäksi käyttäjän tulisi pystyä saavuttamaan

tuotteen käytölle asetettu tavoite tehokkaasti, tuloksellisesti ja tyytyväisesti. (Saavutettavuus 2020.)

Saavutettavuudella tarkoitetaan sitä, että kohde on helposti lähestyttävissä kaikille käyttäjille. Verkkoympäristössä tämä tarkoittaa käytännössä sitä, että verkkopalvelu on kaikkien käytettävissä mahdollisimman vaivattomasti. Jotta verkkopalvelu olisi kaikkien saatavilla, sen tulee olla esitettynä sellaisessa muodossa, että käyttäjä pääsee siihen käsiksi rajoitteista riippumatta. Sisältöä voidaan esittää eri muodoissa, kuten esimerkiksi tekstinä tai äänenä, jotta se saavuttaa mahdollisimman laajan käyttäjäjoukon. Lisäksi saavutettavuuteen liittyy myös ymmärrettävyys. Tällä tarkoitetaan tiedon esittämistä niin, että se on selkeässä ja helposti ymmärrettävässä muodossa. (Saavutettavuus 2020.)

Verkkosovelluksen saavutettavuuteen liittyy sekä tekninen että sisällöllinen saavutettavuus. Näiden lisäksi verkkosovelluksen käytettävyys on tärkeää saavutettavuuden kannalta. Teknisellä saavutettavuudella tarkoitetaan käytettävyyttä erilaisten teknisten apuvälineiden avulla. Toisin sanoen verkkosovelluksen tulisi olla kehitetty niin, että se on koodattu virheettömästi ja standardeja noudattaen. Sisällön saavutettavuudella pyritään siihen, että verkkosovelluksen sisältö on helposti ymmärrettävissä, omaksuttavissa sekä käytettävissä. Sisällön saavutettavuutta voidaan parantaa muun muassa käyttämällä yleisesti selkeää kieltä, välttämällä erikoisia ja vaikeasti ymmärrettäviä termejä ja jaksottamalla tekstisisältöä helposti ymmärrettäviin kokonaisuuksiin. (Saavutettavuus 2020.)

4 Pelillistäminen

Pohjimmiltaan pelillistämisen idea kumpuaa oletuksesta, että koska pelit ovat hauskoja, mikä tahansa muu järjestelmä, joka tehdään pelinkaltaiseksi, muuttuu myös hauskemaksi ja mielekkäämmäksi. Näin ollen pelillistäminen on palvelun- tai järjestelmän suunnittelua, jonka pyrkimyksenä on herättää samoja psykologisia vaikutuksia, joita pelaaja kokee pelatessaan perinteisiä pelejä. Tätä hyödyntäen pelillistämässä pyritään kannustamaan tietynlaiseen toimintaan tai käyttäytymiseen. Pelillistämisen avulla voidaan siis ohjata pelaajaa kehittäjän toivomaan suuntaan, kuten oppimaan uusia asioita tai liikkumaan enemmän. (Peliviikko 2013.)

Pelillisessä oppimisprosessissa yksi keskeisimmistä asioista on se, että pelaaja pääsee itse vaikuttamaan pelin etenemiseen omien valintojensa kautta. Pelaaja päättää itse miten hän toimii sen pohjalta, mitä hän on ymmärtänyt pelin tarinan kehityksessä. Näiden valintojen kautta pelaaja edistyy tavoitteissaan. Pelimotivaation kannalta pelin suunnittelijan on tärkeää suunnitella palautejärjestelmä, jonka kautta pelaaja saa tietoa omasta edistymisestään. (Peliviikko 2013.)

Pelillistäminen on prosessina samankaltainen, kuin perinteisen pelin suunnittelu. Pelillistämisen keinoina hyödynnetään pääosin samoja elementtejä, kuin perinteisissä peleissä. Näitä elementtejä ovat esimerkiksi tavoitteet, pisteet, ansiomerkit tai tarinallinen kerronta. Kuitenkin perinteisten pelien suunnittelussa sekä pelillistämässä on isoja eroja. Perinteisen pelin suunnittelussa yksi tärkeimmistä tarkoituksista on tehdä pelistä mahdollisimman hyvä, kun taas pelillistämisen yksi tärkeimmistä ominaisuuksista on jonkin tietyn järjestelmän tai palvelun arvon lisääminen sekä lisätä kohderyhmän motivaatiota sitä kohtaan. (Peliviikko 2013.)

Koska pelillistämisen tarkoituksena on saada käyttäjä motivoitumaan, on tärkeä miettiä, mikä tekee niistä mukaansatempaavia sekä mielekkäitä. Erään pelitutkimuksen mukaan, pelien vetovoimaisuus perustuu niiden synnyttämään osaamisen, onnistumiseen ja hallinnan kokemukseen, sekä jännitykseen ja sosiaaliseen kanssakäymiseen. Pelillistämisen tavoitteena on näiden samojen kokemusten herättäminen muissakin konteksteissa kuin perinteisissä peleissä. (Peliviikko 2013.)

4.1 Pelaajatyypit

Andrzej Marczewskin mukaan pelaajat voidaan jakaa karkeasti kuuteen erilaiseen tyyppiin. Marczewski on jakanut nämä kuusi erilaista pelaajatyyppejä sen mukaan, mitkä asiat saavat heidät motivoitumaan pelatessaan. Hänen mukaansa neljä näistä ovat perustasoltaan luontaisia pelaajatyyppejä. Tässä mallissa heidät on luokiteltu seuraavasti: sosiaalinen (socialisers), oman tiensä kulkija (free spirits), saavuttaja (achievers) sekä hyväntekijä (philanthropists). Nämä pelaajatyypit motivoituvat samaistuttavuudesta (relatedness), itsenäisyydestä (autonomy), hyväksi tulemisesta (mastery) sekä tarkoituksesta (purpose) ja hän käyttää näistä lyhennettä RAMP. Näille kaikille on yhteistä, että motivaation lähde tulee pelaajista itseltään, eli sisäisestä motivaatiosta (intrinsic motivation). Kaksi muuta pelaajatyyppejä ovat häiritsejä (disruptor) sekä pelaaja (player). Näille kahdelle pelaajatyypille on ominaista ulkoiset motivaation lähteet. Marczewskin mukaan eri pelaajatyypit motivoituvat erilaisista asioista alla olevan kuvion 3. mukaisesti. (Marczewski 2015, 65-80.)

- **Sosiaalinen** pelaaja motivoituu **Samaistuttavuudesta**. He haluavat olla toisten pelaajien kanssa tekemisissä ja luoda sosiaalisia kontakteja.
- **Oman tiensä kulkija** motivoituu **Itsenäisyydestä**. He haluavat luoda ja tutkia oman tahtonsa mukaan.
- **Saavuttaja** motivoituu **Hyväksi tulemisesta**. He motivoituvat oppiessaan uusia asioita ja kehittyessä paremmaksi. He haluavat vaikeusasteen kasvavan kehityksen myötä.
- **Hyväntekijä** motivoituu **Tarkoituksesta**. He haluavat antaa muille pelaajille ja rikastuttaa toisten elämää jollain tavalla ilman, että he odottavat saavansa siitä itse palkintoa.
- **Pelaaja** motivoituu **Palkinnosta**. He tekevät kaiken mitä tarvitsee saavuttaakseen seuraavan palkinnon.
- **Häiritsijä** motivoituu **Muutoksesta**. Yleisesti ottaen he haluavat häiritä systeemiä. Joko suoraan tai välillisesti muiden pelaajien kautta.

Kuvio 3: Pelaajatyyppien motivoiminen

Yksinkertaistettuna pelaajat mielellään pelaavat jotain peliä, jossa pisteet ja palkinnot ovat saavutettavissa. Häiritsijät taas ovat tämän vastakohta, jotka eivät halua kumpaakaan näistä. Loput neljä pelaajatyyppiä tarvitsevat hieman enemmän, jotta heidän mielenkiintonsa peliä kohtaan säilyy. (Marczewski 2015, 65-80.)

4.2 Pelillistämisen keinoja

Pelimekaniikat ovat perustoimintoja, prosesseja, visuaalisia ja ohjausmekanismeja, joita käytetään toimintojen pelillistämiseen. Pelimekaniikat, jotka ovat pelillistämisen perusta, ovat säännöt ja palkinnot, jotka muodostavat pelin ja luovat kiehtovan kokemuksen. Pelidynamiikka viittaa joukkoon tunteita, käyttäytymismalleja ja toiveita, jotka löytyvät pelimekaniikasta ja jotka resonoivat ihmisten kanssa. Esimerkkejä voivat olla kilpailu erilaisten pisteiden vertailujen kautta, erilaisten yhteisöllisten tehtävien suorittaminen, muiden kanssa kommunikointi, erilaisten uniikkien saavutusten kerääminen ja uusien tehtävien tuoma yllätyksellisyys. Pelidynamiikkaa käytetään pelimekaniikan kanssa sitoutumisen edistämiseen ja osallistujien motivointiin. (Marczewski 2015, 65-80.)

Pisteiden kerääminen on yksi pelimekaniikoista, joita voidaan hyödyntää pelillistämässä. Pisteitä voidaan käyttää myös tilan ilmoittamiseen, korkean prioriteetin toimintojen tunnistamiseen tai joissain tapauksissa asioiden ostamiseen. Pisteihin liittyvä pelidynamiikka sisältää saavutuksen ja etenemisen tunteen. Käyttäjät haluavat tuntea olevansa palkittuja, ja pisteet

voivat auttaa käyttäjiä saavuttamaan tämän. Pisteisiin voidaan myös yhdistää sosiaalinen elementti tulostaulukoiden kautta. Tulostaulukon avulla käyttäjät voivat verrata omaa edistymistään tai tasoaan toisiin käyttäjiin nähden. Tulostaulun avulla voidaan vaikuttaa käyttäjän motivaatioon suorittaa haluttu toiminto entistä paremmin. (Marczewski 2015, 65-80.)

Haluttuja toimintoja voidaan pelillistää myös erilaisten tehtävien avulla. Tehtävät tarjoavat käyttäjille tavoitteita joko ryhmänä tai yksilöinä. Kun tehtävä on suoritettu, käyttäjät saattavat tuntea onnistumisen tunteen. Tehtävien tekemiseen voidaan yhdistää myös niiden lukitukset, jolla pyritään motivoimaan käyttäjää pitkäaikaiseen suorittamiseen halutun toiminnon parissa. Esimerkiksi käyttäjän on suoritettava tehtävä A, jonka suorittaminen avaa seuraavan tehtävän B. (Marczewski 2015, 65-80.)

Tietokilpailua voidaan hyödyntää yhtenä pelimekaniikan osana pelillistämässä. Tietokilpailua hyödynnetään yleensä osana muita pelillistämisen keinoja. Tietokilpailun avulla käyttäjät voivat testata tietämystään halutun aihepiirin parissa ja tällä tavalla antaa käyttäjälle palautetta hänen edistymisestään. Käyttäjän edistymisen visualisointi on jo itsessään yksi pelillistämisen keino. (Marczewski 2015, 65-80.)

5 Tutkimusmenetelmät

Laadullisessa tutkimuksessa tutkimuksen tuloksia tarkastellaan tutkimuksen sisällön mukaan. Kvantitatiivisessa, eli määrällisessä tutkimuksessa, taas keskitytään enemmän lukuihin ja määriin. Näiden menetelmien tiedonkeruu eroaa toisistaan hieman. Laadullisen tutkimuksen prosessi on usein joustavampi ja sen aineisto on kerätty todellisissa tilanteissa. Laadullisen tutkimuksen tulosten keräämiseen käytetään usein erilaisia haastatteluja, havainnoiteja tai esimerkiksi kuvien, videoiden, pöytäkirjojen tai vuosikertomusten tulkitsemista. Haastattelut voivat olla esimerkiksi ryhmä- tai teemahaastatteluja. Laadullisen tutkimuksen aineistoa usein analysoidaan sen hankkimisen kanssa samanaikaisesti. Tällä tavalla saadaan jäsenettyä kerättyä aineistoa, jotta aineiston tulkinta olisi mahdollisimman selkeää. Analysointi aloitetaan useimmiten tutustumalla analysoitavaan aineistoon ja käymällä se lävitse. Tämän jälkeen aineisto luokitellaan tyyppin, teeman tai sisällön mukaan. Viimeisenä vaiheena aineistosta löydetyt havainnot liitetään oikeaan asiayhteyteen. (Järvenpää 2006.)

5.1 Haastattelu ja kysely

Haastattelu ja kysely eroavat toisistaan niiden suoritus tavan perusteella. Haastattelu suoritetaan reaaliaikaisessa vuorovaikutuksessa haastattelijan ja haastateltavan välillä. Haastattelijan ja haastateltavan ei tarvitse kuitenkaan kohdata toisiaan kasvotusten vaan haastattelu voidaan järjestää myös esimerkiksi puhelimitse tai internetin välityksellä. Kyselyssä taas kyselyyn vastaava vastaa ennalta määrättyihin kysymyksiin esimerkiksi lomakkeella tai kirjoittaen.

Kyselyssä ei ole vuorovaikutusta kyselyn laatija ja kyselyyn vastaavan henkilön välillä. Kyselyissä käytetään kahden tyyppisiä kysymyksiä, suljettuja sekä avoimia. Avoimessa kysymyksessä vastaaja päättää itse, miten hän siihen vastaa, kun taas suljetussa kysymyksessä on ennalta määrätty vaihtoehdot, joista vastaaja valitsee osuvimman. Avoimia ja suljettuja kysymyksiä voidaan käyttää myös yhtäaikaaisesti kyselyn aikana. (Routio 2007.)

Kyselyn etuna haastatteluun verrattuna on se, että haastattelu vaatii kummaltakin osapuolelta enemmän aikaa. Kuitenkin haastattelun etuna on se, että se joustaa ja mukautuu tilanteen mukaan kyselyä enemmän. Haastattelussa tärkeää on aineiston dokumentointi ja sen voi suorittaa esimerkiksi nauhoittamalla haastattelu tai tekemällä haastattelusta yksityiskohtaiset muistiinpanot. Haastattelu mahdollistaa tilanteeseen mukautumisen ja sen aikana voidaan esittää tarkentavia kysymyksiä. Haastatteluja voidaan järjestää esimerkiksi teema- tai ryhmähaastatteluina. Haastattelu sopii sellaisiin tilanteisiin, jossa aineistoa keräävä ei pysty määrittelemään etukäteen kysyttäviä kysymyksiä. (Routio 2007.)

Kyselyn haasteena on se, että mikäli ennalta määrättyjä kysymyksiä tarvitsee muokata, mahdollisesti koko kysely joudutaan tekemään uusiksi. Tämän takia on tärkeää, että kysymykset on suunniteltu hyvin. Lisäksi kysymysten tulee olla tarpeeksi selkeitä ja ymmärrettäviä, sillä kyselyssä ei ole haastattelun tapaan vuorovaikutuksessa kyselyn tekijän kanssa. Kysely sopii hyvin sellaisissa tilanteissa, jossa tutkimuksen resurssit ovat pienet tai tutkimukseen osallistuu paljon vastaajia, jolloin kaikkien heidän haastattelemisensa olisi todella työlästä. (Routio 2007.)

Aineiston analyysillä on tarkoitus tiivistää ja jaotella saatua aineistoa niin, että aineistosta saadaan looginen ja selkeä kokonaisuus, ilman että tärkeää tietoa katoaa. Analyysivaiheessa selvitetään, millaisia vastauksia tutkimuskysymyksiin saadaan sekä miten ne olisi voinut mahdollisesti asetella paremmin. Kerätty aineisto litteroidaan, eli se kirjoitetaan kokonaisuudessaan puhtaaksi. (Hirsjärvi, Remes & Sajavaara 2004, 209- 210.)

5.2 Reliabiliteetti ja validiteetti

Tutkimuksen luotettavuutta voidaan verrata tutkimukseen käytetyn mittarin luotettavuuteen. Perinteisesti luotettavuutta voidaan kuvata kahdella termillä: reliabiliteetilla ja validiteetilla. Vaikka molemmat termit viittaavat luotettavuuteen, niin reliabiliteetin sisältö tarkoittaa tutkimuksen toistettavuutta. Tämä tarkoittaa sitä, että jos samaa ilmiötä tutkittaisiin useita kertoja ja tutkimuksessa käytettäisiin samoja mittareita, niin kuinka samankaltaisia tai eroavaisia vastauksia tutkimuksella saataisiin. Mikäli mittaustulos on tutkimusten välillä yhteneväinen, voidaan todeta tutkimuksen olevan reliaabeli. Toisin sanoen, jos tutkimuksen tulos ei ole reliaabeli, niin saatu tulos on silloin sattumaa eikä ole näin ollen toistettavissa. Validiteetilla tarkoitetaan sitä, onko tutkimuksen kohderyhmä kysymykset juuri kyseiseen tutkimukseen.

Validiteettia voidaan pohtia siltä kannalta, kuinka hyvin tutkimusote ja siinä käytetyt menetelmät vastaavat tutkittavaa ilmiötä. (Metsämuuronen 2001, 50.)

5.3 Vertailuanalyysi

Benchmarkingilla eli vertailuanalyysilla tarkoitetaan oman toiminnan vertaamista toiseen vastaavaan toimintaan. Vertailuanalyysin perusideana on oppia toisten toiminnan kautta sekä kyseenalaistaa oma toiminta. Tarkoituksena on jatkuvasti verrata toimintojen tuottavuutta, laatua, työprosessien ja työtapojen tehokkuutta vastaaviin toimintoihin, jotka ovat kyseisillä alueilla parhaimpia. Vertailua voi tehdä joko saman alan tai muiden alojen toimijoihin nähden. (Itä-Suomen Yliopisto 2020.)

Usein vertailuanalyysi toteutetaan vierailemalla organisaatiossa, jonka kanssa oman organisaation toimintaa halutaan vertailla. Vertailun voi suorittaa muillakin tavoilla, kuten esimerkiksi etsimällä tietoa parhaista käytänteistä erilaisista julkaisuista, kuten artikkeleista, kirjoista tai Internet-sivuilta. Vertailua voidaan tehdä esimerkiksi vertailemalla eri mittareita organisaatioiden välillä ja tällä tavalla löytää ne toiminnot, joita haluaa tutkia tarkemmin. Vertailuanalyysia voidaan tehdä myös yhteistyössä eri organisaatioiden välillä, jolloin vertailuanalyysiin osallistuvat organisaatiot vertailevat toimintaansa säännöllisesti yhteistyössä ja kehittävät toimintatapojaan yhdessä. Lisäksi vertailuanalyysia voidaan suorittaa organisaation sisäisesti, jolloin eri yhtiöt vertaavat toimintojaan sekä prosesseja keskenään ja jakavat hyväksi huomaaviaan käytänteitä koko organisaation sisällä. (Itä-Suomen Yliopisto 2020.)

6 Kehittämistyön toteutus

Kehittämistyön aluksi kartoitettiin, onko vastaavanlaisia tuotteita jo olemassa. Tällä haluttiin varmistaa, ettei täysin samanlaista tuotetta ole jo kehitetty, jota asiakas olisi voinut hyödyntää. Samalla oli tarkoituksena kartoittaa, mikäli samankaltaisia tuotteita oli kehitetty, että, mitä niistä voisi oppia. Mitä asioita kannattaisi hyödyntää tai vastaavasti minkälaisia asioita kannattaisi välttää.

Kartoitus, eli vertailuanalyysi, toteutettiin etsimällä internetistä vastaavanlaisia tuotteita niin Suomesta kuin ulkomailtakin. Täysin samasta aiheesta tai vastaavanlaista tuotetta ei löytynyt. Tästä syystä tutkittiin samalle kohderyhmälle tarkoitettuja sivustoja ja niiden sisältöä. Yhtäläisyytenä suurimmalle osalle lapsille tarkoiteuista sovelluksista oli niiden yksinkertainen toteutus ja helppokäyttöisyys. Näitä samoja ominaisuuksia haluttiin myös kehitettävään sovellukseen.

Vaativuusmäärittely suoritettiin sen jälkeen, kun muut vastaavanlaiset tuotteet tai sivustot oli kartoitettu. Asiakkaan näkökulmasta toteutukselle ei annettu rajoitteita, vaan opinnäytetyön

tekijä sai vapaat kädet teknisen toteutuksen suhteen. Sovellukselle määriteltiin halutut toiminnallisuudet, joita siinä pystyy tekemään sekä joitakin määriytyksiä visuaalisuuteen liittyen. Pelillistämisen keinoista ei ollut tarkkoja määritelmiä, vaan ne sai toteuttaa sen mukaan, mikä opinnäytetyöntekijän mielestä onnistuu parhaiten.

Kun kaikki sovelluksen toiminnallisuudet olivat määritelty, seuraava vaihe oli niiden toteutus. Tässä vaiheessa ei vielä kiinnitetty huomiota visuaalisuuteen vaan siihen, että toiminnot toimivat halutulla tavalla. Koska sovelluksen toiminnallisuudet olivat yksinkertaisia, ne pystyttiin toteuttamaan samaan aikaan, kun vielä visuaalista ilmettä suunniteltiin.

Sovelluksen toimivuutta testattiin koko sen kehityksen ajan. Aina kun uusi toiminto oli ohjelmoitu, sen toimivuus tarkistettiin. Uusien toiminnallisuuksien lisäämisen jälkeen kaikki olemassa olevat toiminnallisuudet käytiin lävitse, jotta voitiin varmistua siitä, että mikään ei ole mennyt rikki. Testauksessa käytettiin eri selaimia sekä laitteita, jotta pystyttiin varmistamaan toimivuus eri alustoilla. Lisäksi valmiin sovelluksen kohdalla testattiin sovelluksen toimivuutta oikealla kohderyhmällä ennen julkaisua.

Kehittämistyössä ei varsinaisesti ollut käytössä mitään tiettyä kehitysmenetelmää, sillä työskentely tapahtui pääasiassa yksin. Kuitenkin kehitystyö noudatti pääsääntöisesti yleisiä ketterien kehitysmenetelmien malleja, varsinkin Scrum-kehitysmenetelmää. Tämä työskentelytapa toimi, koska työskentely tapahtui yhden henkilön toimesta, eikä erityisiä aikataulu rajoituksia toteutukselle ollut. Tiedon ja materiaalin jakamiseen toimeksiantajalle käytettiin Googlen tarjoamaa Google Drive -pilvipalvelua. Valmista työtä varten sovellus ladattiin opinnäytetyöntekijän omaan Microsoft Azure -pilvipalveluun, jossa toimeksiantaja pystyi testata sovellusta.

6.1 Sovelluskehitysprosessin tiedonkeruu

Kehittämistyön tiedonhankintamenetelmänä käytettiin vapaamuotoista haastattelua sovelluksen visuaalisuuteen liittyen. Vapaamuotoisessa haastattelussa selvitettiin värien käyttöä ja niiden vaikutusta käyttäjään, joka on värisokea. Haastattelu toteutettiin näyttämällä haastateltavalle eri versioita erilaisista väri- ja niiden sävy-yhdistelmistä ja haastattelussa kerättiin tietoa siitä, kuinka hyvin haastateltava pystyi hahmottamaan kyseistä sisältöä. Haastattelujen avulla värien käytöstä saatiin konkreettista tietoa sen saavutettavuudesta, joka oli tärkeässä osa kehitystyötä.

Koska haastattelut olivat yksinkertaisia, tulosten taltiointiin käytettiin vain muistiinpanoja. Tällä varmistettiin, että tuloksiin voidaan tarvittaessa palata myöhemminkin. Haastatteluissa saatujen vastausten lukumäärä oli vähäinen, joten se saattaa vaikuttaa niiden analysointiin sekä reliabiliteettiin. Haastattelujen tulokset jaettiin eri kategorioihin tulosten analysoimiseksi. Nämä kategoriat olivat värien erottaminen toisistaan, värien selkeys sekä värien vai-

kutus ulkoasun selkeyteen. Kategorisoinnin avulla tuloksia analysoitiin ja sen perusteella pystyttiin valita sellaiset värit, jotka olivat mahdollisimman selkeät käyttäjien mahdolliset haasteet huomioon ottaen.

Kun sovelluksesta oli ensimmäinen versio valmis, kerättiin siitä palautetta ja kehitysehdotuksia järjestämällä kysely kahdelle kohderyhmään kuuluvalle luokalle. Kyselyssä kerättiin tietoa sovelluksen ulkoasuun ja sisältöön liittyen. Lisäksi kyselyssä oli lopuksi avoin kysymys, mikäli kyselyyn osallistuva halusi antaa muuta palautetta avoimesti. Kyselyyn osallistuneiden oppilaiden luokanopettaja antoi omia havaintojaan oppilaiden mielipiteistä. Kyselyn tuloksia analysoitiin teemojen mukaisesti, jotka olivat visuaalinen ilme sekä sisältö. Näiden teemojen sisällä vastaukset kategorisoitiin vielä kahteen kategoriaan sen perusteella, tulisiko vastauksen aihe muuttaa vai toimisiko se sellaisenaan.

Sovelluksen kehitysprosessin alkuvaiheessa kerättiin tietoa vastaavanlaisista sovelluksista vertailuanalyysiä hyödyntämällä. Vertailuanalyysillä kerättiin tietoa, onko vastaavanlaisia sovelluksia jo olemassa ja minkälaisia ominaisuuksia niistä kannattaisi hyödyntää myös tässä projektissa. Vertailuanalyysin tuloksena ei kuitenkaan löytynyt vastaavan kaltaisia sovelluksia, joten vertailua suoritettiin sovelluksiin ja verkkopalveluihin, jotka olivat suunniteltu lapsille.

6.2 Vaatimusmäärittely

Asiakkaan kanssa käytiin yhdessä lävitse mitä vaatimuksia opinnäytetyöprojektille oli. Paloturvallisuuteen liittyvän pelillistämisen toteutukseen annettiin erittäin vapaat kädet. Vaatimuksia määriteltäessä neljä pääkohtaa nousi esiin, jotka olivat saavutettavuus, värimaailma, käytettävyyden ja tekninen toteutus.

Saavutettavuus oli yksi tärkeä asia tuotteelle, sillä kaikkien kohderyhmään kuuluvien pitäisi pystyä käyttämään sovellusta ilman suurempia vaikeuksia esimerkiksi hahmottamisen kanssa. Värimaailman suhteen ei ollut tarkkoja vaatimuksia, mitä värejä tulisi käyttää, mutta oli erittäin tärkeää huomioida, että valitut värit eivät tuo rajoitteita sovelluksen käyttäjille. Värien suhteen siis määriteltiin, että värien tulisi olla mahdollisimman selkeitä, myös sellaisille käyttäjille, joilla on esimerkiksi värisokeus. Lisäksi määriteltiin niin, että värimaailman tulisi olla mahdollisimman selkeä, jotta kohderyhmä huomioon ottaen se ei aiheuttaisi esteitä pelaamiselle.

Saavutettavuuden lisäksi yksi vaatimus sovellukselle oli se, että se on mahdollisimman helppokäyttöinen. Tällä pystyttäisiin varmistumaan siitä, että oppimisprosessi olisi helpompi. Tärkeää oli myös se, että valistusta pitävät henkilöt pystyvät antamaan neuvoja sovelluksen käytöstä heidän osaamistasostaan riippumatta.

Tässä vaiheessa toiminnallisuuksista oli vain kaksi vaatimusta. Käyttäjälle tuli esittää kysymyksiä eri aihealueista sekä oikeista ja vääristä vastauksista tuli saada palautetta. Jo alkuvaiheessa vaatimuksena oli se, että palaute vastauksiin annetaan heti kysymykseen vastaamisen jälkeen.

Tekniselle toteutukselle ei annettu tarkkoja määritelmiä, vaan opinnäytetyön tekijä sai itse päättää tavan, jolla sovellus toteutettiin. Ainoa tekninen vaatimus sovellukselle oli se, että sitä voidaan käyttää eri laitteilla, kuten tietokoneella, mobiililla tai tabletilla. Kuitenkaan tarkkoja määriä tuettujen selaimien tai käyttöjärjestelmien suhteen ei määritelty.

6.3 Suunnitteluvaihe vaatimusten mukaisesti

Prototyyppi sovelluksen ulkoasusta ja toiminnallisuuksista luotiin Balsamiq-sovellusta hyödyntäen. Sovelluksen ulkoasusta suunniteltiin useampi erilainen vaihtoehto, jotka lähetettiin toimeksiantajalle. Toimeksiantajan palautteen perusteella luotiin uudet prototyypit ulkoasusta ja toiminnallisuuksista, kunnes opinnäytetyöntekijä sekä toimeksiantaja olivat niihin tyytyväisiä. Ulkoasun suunnittelun jälkeen oli sovelluksen toimintojen suunnittelun vuoro. Tähän käytettiin samaa menetelmää, kuin ulkoasun suunnittelussa.

Vaikka ulkoasusta ja toiminnallisuuksista oli tehty prototyypit, niin ne muuttuivat jatkuvasti kehityksen eri vaiheissa. Niitä hiottiin koko prosessin ajan aina, kun joko toimeksiantajalta tai opinnäytetyöntekijällä tuli uusia parannus ideoita.

6.4 Sovelluksen näkymät

Sovelluksen rakenne sisältää vain muutaman erilaisen näkymän, jotka ovat kaikille käyttäjille samanlaisia. Sovelluksen näkymät voidaan jakaa karkeasti aloitussivuun, aihealueen valintasiivuun, kysymyssivuihin sekä loppupalautesivuun. Väriteemana kaikilla sivuilla on mustan, harmaan ja valkoisen eri sävyt pois lukien muutamaa poikkeusta. Koko sivustolla taustalla on väriteemaan mukainen taustakuva mustavalkoisena kuvana. Aihealueen valinnan yhteydessä on eri väriteemoilla kuvaavat symbolit sekä kaikki kysymykseen liittyvät kuvat ovat toteutettu värikköinä. Kaikki tekstit ovat valkoista mustaa taustaa varten, jotta näiden välinen kontrasti olisi suuri. Tämä helpottaa tekstien erottamista taustasta ja tekee siitä näin ollen helpommin lähestyttävän, ottaen huomioon kohderyhmän lukemisen tason. Fontiksi on valittu monospace-fontti, joka on erittäin selkeästi luettavissa.

Aloitussivulla käyttäjälle kerrotaan mitä sivustolla voidaan tehdä ja lisäksi sieltä löytyy lyhyt oheistus käyttäjälle. Tällä sivulla käyttäjälle kerrotaan, että tarkoituksena on kerätä mahdollisimman monta pistettä, joita saadaan vastaamalla kysymyksiin oikein. Lisäksi käyttäjä saa tiedon siitä, että pisteiden perusteella käyttäjälle jaetaan lopuksi tähtiä saavutukseksi oi-

keista vastauksista. Tällä pyritään saamaan käyttäjä kiinnostumaan vastaamaan ja keskittymään kaikkiin kysymyksiin loppuun saakka. Tämä on tärkeää sovelluksen idean kannalta, sillä tarkoituksena on lisätä tietoisuutta arjen turvallisuudesta.

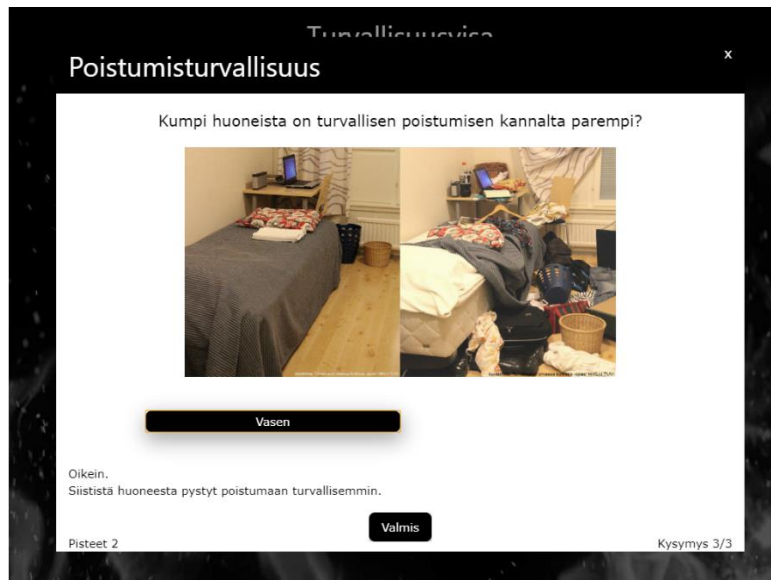
Aihealueen valintasivun näkyvin osa on aihealueita kuvaavat elementit. Aihealueita ovat arjen turvallisuus, poistumisturvallisuus, tulipalo ja hätänumero. Aihealueen valintaa kuvaavat elementit koostuvat väristä, kuvakkeesta sekä tekstistä. Tällä pystytään varmistamaan mahdollisimman hyvää saavutettavuutta. Jokaisella kuvakkeella on oma päävärinsä, jotka ovat keltainen, vihreä, punainen ja sininen. Symboloiden väreissä on vältetty murrettuja sävyjä, jotta myös värisokeiden käyttäjien olisi mahdollisimman mielekästä käyttää kyseistä sivua. Värit valikoituvat osittain ennestään aihealuetta kuvaaviin väreihin, kuten poistumisturvallisuutta kuvaava kuvake on vihreällä taustalla, kuten poistumisreitillä kuvaavissa kylteissäkin. Kehitysprosessin aikana värien eri sävyjä testattiin muutamalla värisokealla testikäyttäjällä ja heidän palautteensa perusteella. Värien testaaminen suoritettiin haastattelulla, jossa testihenkilölle näytettiin eri sävyillä muokattuja kuvia sovelluksesta. Näiden haastattelujen tulosten perusteella valittiin testihenkilöiden mielestä selkein kokonaisuus. Kaikki testihenkilöt olivat yksimielisiä tuloksista. Aihealueen valintasivu kuvattuna kuviossa 4.



Kuvio 4: Aihealueen valinta.

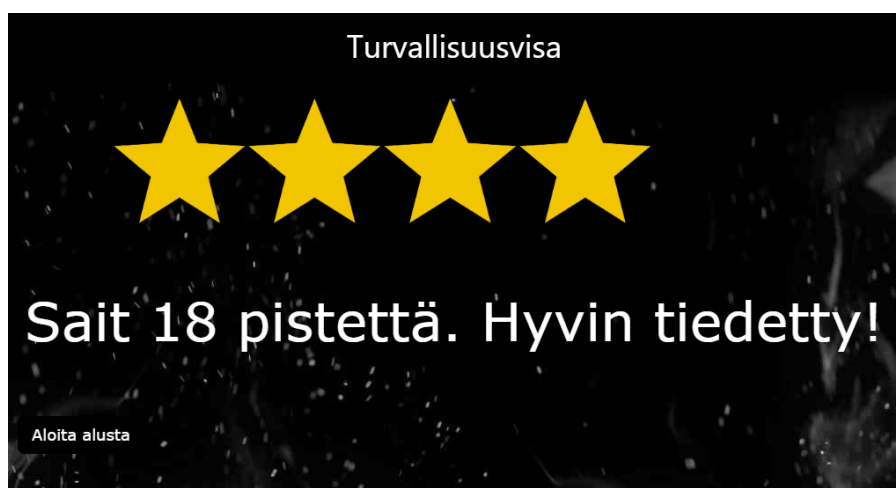
Kysymyssivu koostuu kysymyksestä, kuvasta, vastausvaihtoehdoista sekä vastaamisen jälkeen näkyvästä palautteesta. Lisäksi sivulla näkyy aihealueesta kerätyt pisteet sekä kysymysten määrä reaaliaikaisesti. Muusta sivusta poiketen kysymyssivun tausta on valkoinen, mutta kaikki painikkeet ovat mustia ja teksti niiden sisällä valkoista. Kaikilla kysymyksillä on oma kuvansa, jotka vaihtuvat kysymyksen kuvaan. Kuville ei ole määritelty tarkkaa kokoa leveyden

suhteen, mutta kaikille kuville on määritelty maksimi korkeus, jotta kysymyssivun rakenne pysy määritellyssä koossa kaikkien kysymysten kesken. Osa kuvista tuli suoraan toimeksiantajalta ja osa kuvista otettiin opinnäytetyön tekijän toimesta. Kuvien editoimiseen käytettiin ilmaista kuvienkäsittely ohjelmaa nimeltä GIMP. Kysymyssivu on kuvattuna kuviossa 5.



Kuvio 5: Kysymyssivu

Kun kaikkiin kysymyksiin on vastattu, käyttäjä ohjataan palautesivulle, joka koostuu piste-määrän mukaan yhdestä viiteen tähdestä sekä kirjallista palautteesta. Myös palautesivun tähdet ovat luotu GIMP ohjelmaa hyödyntäen. Lisäksi palautesivulta löytyy painike, jolla käyttäjä pystyy aloittamaan kysymyksiin vastaamisen alusta. Palautesivu on kuvattuna kuviossa 6.



Kuvio 6: Palautesivu

Koska sovellusta on tarkoitus käyttää joko älypuhelimella tai tietokoneella, täytyy sen toimia kaiken kokoisilla näytöillä, eli sen täytyy olla responsiivinen. Tällä tarkoitetaan sitä, että sovelluksen näkymä mukautuu esimerkiksi laitteen näytön koon mukaan. Sovelluksen ulkoasun suunnittelussa ja kehityksessä on käytetty mobile-first- ajatusmallia, jonka ideana on suunnitella käyttöliittymä ensisijaisesti mobiililaitteille soveltuvaksi.

6.5 Sovelluksen toiminnallisuudet

Kaikki toiminnallisuudet ovat tehty JavaScript ohjelmointikieltä käyttäen. Tähän päädyttiin siitä syystä, että opinnäytetyön tekijällä oli siitä eniten kokemusta. Ohjelmointi toteutettiin Notepad ++ ohjelmalla, sillä se on ilmainen ja vastasi projektin tarpeeseen tarpeeksi kattavasti.

Käyttäjän ensimmäinen toiminto sovelluksessa on aihealueen valinta, jossa käyttäjä valitsee, minkä aihealueen kysymyksiin hän haluaa vastata. Aihealueen valintapainiketta painalla sovelluksessa avautuu kysymysnäkymä, jossa käyttäjä pääsee vastaamaan kysymyksiin. Kysymysten vastausvaihtoehtoa painamalla saadaan näkyviin tieto, oliko vastaus oikein vai väärin ja aiheeseen liittyvä tietoisuus (kuvakaappaus). Tämä oli toteutettu niin, että JavaScriptin avulla säädetään, mitkä HTML elementit ovat näkyvissä ja mitkä piilotettuna. Lisäksi kysymyksen vastaamisen jälkeen valittu vastausvaihtoehto jää näkyviin ja ylimääräinen vastausvaihtoehto katoaa näkyvistä. Tällä varmistettiin, että käyttäjä näkee oman vastauksensa ja näin ollen kysymyksen sekä vastauksen liittyvä palaute on helposti yhdistettävissä.

Kysymykset olivat niin sanotusti kovakoodattu, eli ne olivat määritelty suoraan koodissa. Toinen vartenotettava vaihtoehto olisi ollut tietokannan luominen, jota olisi voitu hyödyntää kysymysten kohdalla. Tähän vaihtoehtoon ei kuitenkaan päädytty rajallisten resurssien takia. Kaikki kysymykset ovat siis määritelty etukäteen, eivätkä ne muutu käyttökertojen välillä. Kysymyksiin vastaamisen voi lopettaa erillisestä painikkeesta, vaikka kaikkiin kysymyksiin ei olisi vielä vastattu. Kun aihealueen valitsee uudestaan, kysymykset jatkuvat kuitenkin siitä mihin ne edellisellä kerralla jäivät kesken. Mikäli kysymyksiin vastaamisen haluaa aloittaa alusta, pitää sivu ladata uudelleen. Kysymyksien rakenne on kuvattuna kuviossa 7.

```

let questionsP = [
  {
    questionP : "Mitä tämä merkki tarkoittaa?",
    imgSrc : "Poistumistie.jpg",
    imgCopyrightP : "Kuva: Keski-Uudenmaan pelastuslaitos",
    choiceAP : "Lyhin reitti info-pisteelle",
    choiceBP : "Poistumisreitiltä ulos",
    infoAP : "Oikein. <br> Vihreitä poistumisreitimerkkejä seuraamalla löydät ulos rakennuksesta.",
    infoBP : "Väärin. <br> Vihreitä poistumisreitimerkkejä seuraamalla löydät ulos rakennuksesta.",
  }, {
    questionP : "Joku ei piittaa turvallisuudesta. Mikä kuvassa on pielessä?",
    imgSrc : "Kaaos.jpg",
    imgCopyrightP : "Kuva: Leena Halinen",
    choiceAP : "Tavaroita on oven edessä",
    choiceBP : "Mukenvaunut ovat kaatuneet",
    infoAP : "Oikein. <br> Vihreä poistumisreitimerkki näyttää tien ulos rakennuksesta. <br> Oven eteen ei saa laittaa tavaroita, koska silloin ei pääse ulos.",
    infoBP : "Väärin. <br> Suurin ongelma on tavaroilla tukittu poistumisreitiltä ulos. <br> Tavarat estävät nopean ja turvallisen poistumisen rakennuksesta.",
    correctP : "AP"
  }, {
    questionP : "Kumpi huoneista on turvallisen poistumisen kannalta parempi?",
    imgSrc : "Sotku.png",
    imgCopyrightP : "Kuva: Turvallisesti omassa kodissa -opas/HIKLU TUVI",
    choiceAP : "Vasen",
    choiceBP : "Oikea",
    infoAP : "Oikein. <br> Siististä huoneesta pystyt poistumaan turvallisemmin.",
    infoBP : "Väärin. <br> Sotkun keskellä poistuminen hankaloituu ja voit loukata itseä. <br> Pida huoneesi siistinä!",
    correctP : "AP"
  }
]

```

Kuvio 7: Kysymysten rakenne

Jokaisesta oikeasta vastauksesta saa yhden pisteen. Jokaisen kysymyksen kohdalla on tieto siitä, kuinka monta pistettä aihealueesta on kerättyä sekä monesko kysymys on tällä hetkellä menossa. Kun aihealueen viimeiseen kysymykseen on vastattu, sovellus kerää tiedon siitä, että aihealue on suoritettu. Mikäli kaikkiin aihealueisiin on vastattu sovellus ohjaa käyttäjän loppupalautte sivulle aihealueen valintasivun sijaan.

Loppupalautteet ovat määritelty viiteen eri vaihtoehtoon sen perusteella, kuinka monta pistettä käyttäjä on onnistunut keräämään kysymyksistä. Käyttäjä saa loppupalautteessa tähtiä sekä sanallisen palautteen. Tähtien lisäksi käyttäjälle kerrotaan hänen kokonaispisteensä sekä hän saa sanallisen palautteen. Viisi eri vaihtoehto on määritelty seuraaviin kategorioihin: 0-4, 5-10, 11-14, 15-18 ja 19-20. Lisäksi loppupalautteen yhteyteen on lisätty painike, jolla saa ladattua sivun uudelleen. Tämä toiminnallisuus on toteutettu JavaScriptin reload metodilla.

6.6 Pelillistämisen keinot

Koska kyseessä on hyvin yksinkertainen sovellus ja resursseja projektin toteuttamiselle oli rajallisesti, ei siinä ole vielä hirveästi huomioitu erilaisia pelaajatyyppejä erilaisten pelillistämisen keinoilla. Kuitenkin sovelluksen toteutuksessa on pyritty huomioimaan laaja kohde-ryhmä resurssien sallimassa määrässä.

Jatkuva palaute on yksi pelillistämisen keinoista, joita on hyödynnetty tämän sovelluksen kehityksessä. Sovellus antaa jatkuvasti palautetta käyttäjän edistymisestä sekä siitä, kuinka hyvin käyttäjä on onnistunut vastaamaan kysymyksiin. Lisäksi käyttäjä saa vielä erillisen palautteen, joka kertoo, kuinka hyvin kysymyksiin vastaaminen on sujunut kaiken kaikkiaan. Jatkuvan palautteen lisäksi käyttäjää motivoidaan vastaamaan kysymyksiin mahdollisimman hyvin, sillä jokaisesta oikeasta vastauksesta saa yhden pisteen. Lopullisten pisteiden mukaan saa erilaisen loppupalautteen suoritettuaan kaikki aihealueet.

Sovelluksen käytöstä on pyritty tekemään mahdollisimman yksinkertainen, jotta kaikki omak-suisivat sen käytön helposti. Tällä tavalla pyritään estämään se, että sovellusta ei käytettäisi sen monimutkaisuuden takia. Helppokäyttöisyys on tärkeä osa pelillistämistä, sillä sen avulla pyritään motivoimaan käyttäjiä käyttämään sovellusta. Helppokäyttöisyys vetoaa varsinkin käyttäjiin, jotka ovat niin sanotusti oman tiensä kulkijoita. Sovelluksen käyttöön liittyy myös sosiaalinen elementti, vaikka sitä ei varsinaisesti ole suunniteltu itse sovellukseen, vaan sosiaalisuus tulee sen käyttötarkoituksesta. Sovellusta on tarkoitus käyttää pelastuslaitoksen järjestämän valistuksen yhteydessä, jolloin koko valistukseen osallistuvan luokan olisi tarkoitus käyttää sovellusta.

6.7 Sovelluksen testaus

Vaikka virallista hyväksymistestausta ei toimeksiantajan toimesta tehty, niin toimeksiantajan edustajat kuitenkin testasivat sivuston toimintoja aktiivisesti kehitysprojektin aikana. Aina kun uusia toimintoja tai ominaisuuksia oli tehty, niin toimeksiantajan edustajille lähetettiin testattavaksi uusin versio ja heiltä kerättiin palautetta. Tämän lisäksi opinnäytetyön tekijä testasi jokaisen toiminnon ja ominaisuuden toimimisen aina, kun sivustolle kehitettiin uusia toimintoja. Tämä oli mahdollista, sillä koko sivuston yksinkertaiseen testaamiseen ei mennyt kauaa. Näiden lisäksi opinnäytetyön tekijä teki tarkemman testauksen säännöllisin väliajoin.

Melkein valmista versiota testattiin myös oikealla kohderyhmällä, jotta sen toiminnasta saatiin palautetta myös sen lopullisilta käyttäjiltä. Käyttäjätestausta varten pystytettiin oma palvelu Azuren pilvipalveluun, sillä valmista sovellusta ei ollut julkaistu vielä tässä vaiheessa. Tällä tavalla saatiin luotua testiryhmälle helppo pääsy sovellukseen.

Käyttäjätestauksesta saatiin paljon palautetta niin oppilailta kuin heidän opettajiltaan. Palautteen perusteella tehtiin pieniä muutoksia valmiiseen sovellukseen, josta esimerkkinä toimii kysymysten vaikeusasteen uudelleen miettiminen. Isompia kehitysehdotuksia ja kokonaisuuksia ei tämän opinnäytetyön puitteissa pystytty toteuttamaan, mutta niiden perusteella pystyttiin laatimaan kehitysehdotuksia opinnäytetyön jälkeiselle kehittämiselle. Käyttäjätestauksen vastaukset kerättiin kyselyn avulla. Kyselyn tuloksia analysoitiin luokittelemalla ne visuaaliseen ilmeeseen, sisältöön ja vapaaseen palautteeseen.

7 Kehittämistyön tulokset

Kehittämistyön tuloksena syntyi toimiva sovellus, jossa käyttäjä voi testata omaa tietouttaan arjen turvallisuudesta ja samalla oppia siitä lisää. Sovellusta on tarkoitus hyödyntää 4-luokkalaisille suoritettavien valistusten yhteydessä toimeksiantajan toimesta. Sovellusta pystyy käyttämään kuka tahansa, mutta sen sisältö on suunniteltu tietyille ikäryhmälle.

Sovelluksen toteutuksessa on hyödynnetty pelillistämisen elementtejä käyttäjän motivoimiseksi. Sovelluksessa on tarkoituksena kerätä pisteitä, jonka avulla käyttäjää pyritään motivoimaan vastaamaan kysymyksiin parhaansa mukaan. Pisteiden lisäksi käyttäjä saa jatkuvasti palautetta kysymyksiin vastaamisesta sekä hänen edistymisestään. Kysymyksiin vastattuaan käyttäjä palkitaan lopuksi tähdillä, sen mukaan kuinka monta pistettä hän on onnistunut keräämään.

Sovelluksen toteutuksessa on otettu huomioon sen kohderyhmä sekä yleisesti saavutettavuuteen liittyviä asioita. Sovelluksen käytöstä on pyritty tekemään mahdollisimman yksinkertainen, jotta sen käytössä olisi mahdollisimman vähän haasteita. Tekstin ja sen taustan kontrasti

on valittu niin, että tekstit ovat mahdollisimman helposti luettavissa. Lisäksi värien käyttöä on testattu värisokeilla käyttäjillä, jotta värit ovat mahdollisimman helposti toisistaan erottuvia.

Valmiin sovelluksen lisäksi lisäarvoa tuo se, että tämän opinnäytetyön tuloksena syntyi ensimmäinen versio valmiista sovelluksesta, jota käyttöä ja jatkokehitys mahdollisuuksia voidaan helposti lähteä tutkimaan. Tämänhetkinen sovellus ei ole saavuttanut vielä viimeistä muotoaan, vaan sitä voidaan tarvittaessa lähteä kehittämään entistä paremmaksi.

8 Yhteenveto ja johtopäätökset

Kehittämistyössä suunniteltiin sekä toteutettiin työn toimeksiantajalle, Keski-Uudenmaan pelastuslaitokselle, verkkopohjainen sovellus, jossa käyttäjä voi testata omaa tietouttaan arjen turvallisuudesta. Tätä sovellusta hyödyntäen toimeksiantaja voi lisätä lasten ja nuorten osamista arjen turvallisuudesta.

Kehittämistyö eteni pääsääntöisesti suunnitelmien mukaisesti muutamaa takaiskua lukuun ottamatta. Muutamat esteet saatiin kierrettyä ja kehittämistyö eteni aikataulussa. Sovelluksen kehittäminen toteutettiin HTML-, CSS- ja JavaScript-ohjelmointikielillä, sillä kehittämistyöntekijällä oli niistä eniten kokemusta.

Haastatteluista ja kyselyistä saatujen tietojen luotettavuus perustui siihen, että oikealle kohderyhmälle esitettiin oikeanlaisia kysymyksiä. Haastatteluissa hyödynnettiin värisokeita henkilöitä, jolloin sovelluksen väri yhdistelmät saatiin suunniteltua sellaisiksi, jotta niiden saatuus parani. Kysely suoritettiin oikealle kohderyhmälle, jossa heiltä kerättiin tietoa sovelluksen käytöstä. Tämän palautteen perusteella valmiiseen sovellukseen tehtiin vielä pieniä muutoksia. Mikäli haastattelukysymykset tai kysely suoritettaisiin samoilla henkilöillä uudestaan, olisivat saadut vastaukset hyvin todennäköisesti samankaltaiset, kuin aiemmin.

Kehitystyön tulokset olivat odotettuja, sillä kehitystyön resurssit olivat rajalliset. Kehitystyötä oli tekemässä vain yksi henkilö ja kehitystyö toteutettiin muun työn ohessa. Valmis sovellus täytti sille ennalta määrättyt vaatimukset ja se vastasi sille olevaan tarpeeseen.

9 Jatkokehitysehdotukset

Sovelluksen seuraavaan versioon voitaisiin kehittää tietokanta kysymyksille, jolloin ne eivät olisi niin sanotusti kovakoodattu JavaScriptissä. Tietokantaa hyödyntäen erilaisten kysymysten määrää voitaisiin lisätä ja kysyttävät kysymykset voitaisiin arpoa isommasta määrästä kysy-

myksiä. Näin jokaisesta käyttökerrasta saataisiin erilainen, jolloin mielenkiinto käyttää sovelusta uudestaan kasvaisi. Tietokannan avulla vastauksia ei pystyisi tarkastamaan selaimen kehittäjän työkalua hyödyntämällä, vaikka tämä ei varsinaisesti ongelma olekaan. Lisäksi ylläpidolle voitaisiin tehdä käyttöliittymä, jossa kysymyksiä pääsisi lisäämään, poistamaan ja muokkaamaan helposti.

Tulevaisuudessa opinnäytetyössä kehitettyyn sovellukseen voitaisiin lisätä enemmän erilaisia toimintoja ja hyödyntää niissä eri pelillistämisen keinoja. Esimerkiksi lyhyen aikavälin kehittämiskohteenä voisi olla pistetaulukon kehittäminen, jolloin oppilaat voisivat suoraan verrata omia tuloksia toisten tuloksiin. Lisäksi tällä tavalla pystyttäisiin seuraamaan, ketkä kaikki ovat käyttäneet sovellusta. Pistetaulukon lisäksi voitaisiin tehdä asteittain vaikeutuvat kysymykset, tai kysymysten vaikeusaste voitaisiin valita valikon kautta.

Edellisten asioiden lisäksi myös sovelluksen sisältöä voisi kehittää paremmaksi. Sovellukselle voisi keksiä houkuttelevan nimen, joka saisi oppilaat enemmän kiinnostumaan sovelluksesta. Tämä asia korostuu, mikäli sovellusta aletaan käyttää laajemmin. Myös ääntä ja erilaisia symboleita voitaisiin hyödyntää enemmän. Esimerkiksi ääntä voitaisiin hyödyntää ilmaisemaan, vastattiinko kysymykseen oikein vai väärin. Äänen lisäksi siihen voitaisiin lisätä kuvaava symboli.

10 Oman oppimisen arviointi

Opin projektin edetessä paljon asioita. Projektin parissa työskentely oli erittäin itsenäistä, sillä hoidin kehitystyön yksin. Tämän takia pystyin itse vastaamaan omasta aikataulustani sekä toteutustavasta. Tässä projektissa oppi tuntemaan omat vahvuudet sekä heikkoudet projektityöskentelyyn liittyen entistä paremmin. Esimerkiksi itselleni sopi parhaiten useat lyhyen aikavälin tavoitteet, jotta lopulta kokonaisuuden hallinta onnistui paremmin. Omat haasteensa aikataulukseen toivat työt, jotka olivat täysin erillään opinnäytetyön projektista.

Tämä projekti oli omasta mielestäni erittäin mielenkiintoinen, sillä tämän avulla pääsi tutustumaan sovelluskehitykseen kokonaisvaltaisesti, sillä olin itse vastuussa kaikesta. Tämän takia myös vastuu projektin etenemisestä oli minulla. Projektin aikana pääsin hyödyntämään niin koulussa kuin työelämässäkin opittuja asioita. Projektiin liittyen omasta mielestäni haastavinta oli visuaalinen suunnittelu, sillä sen suunnittelu toteutettiin omien näkemyksieni mukaan.

Tämän projektin ansiosta sain mielestäni hyvät lähtökohdat tulevaisuudessa työskentelemään samankaltaisissa projekteissa. Pystyin hyödyntämään jo projektin aikana opittuja asioita työelämän projekteissa. Olen tyytyväinen projektin lopputulokseen, sillä sen lähtökohtana oli

täysin uusi projekti ilman minkään näköistä pohjaratkaisua, jota olisi voinut kehittää eteenpäin.

Lähteet

Painetut

Beighley, L. & Morrison, M. 2009. Head first PHP & MySQL. Sebastopol (CA): O'Reilly.

Diaconescu, M. & Wagner, G. 2017. Web Applications with JavaScript or Java. Berlin: De Gruyter Oldenbourg.

Duckett, J. 2011. HTML and CSS: Design and build websites (1st ed.). Hoboken: Wiley.

Hirsjärvi S., Remes P. & Sajavaara P. 2004. Tutki ja kirjoita. Jyväskylä: Gummerus Kirjapaino.

Hong, P. 2018. Practical web design: Learn the fundamentals of web design with HTML5, CSS3, bootstrap, jQuery, and vue.js. Birmingham: Packt Publishing

Kasurinen, J. P. 2013. Ohjelmistotestauksen käsikirja. Jyväskylä: Docendo

Marczewski, A. 2015. Even Ninja Monkeys Like to Play: Gamification, Game Thinking and Motivational Design (1st ed.). CreateSpace Independent Publishing Platform

Metsämuuronen, J. 2002. Metodologian perusteet ihmistieteissä. Metodologia -sarja 1. Helsinki: International Methelp Oy.

Suominen, A. 2000. Riskienhallinta. Vantaa: Tummavuoren Kirjapaino Oy.

Sähköiset

Green, C. 2000. Classics in the History of Psychology. Viitattu 31.3.2020. <http://psychclassics.yorku.ca/Maslow/motivation.htm>

Itä-Suomen Yliopisto. 2020. Benchmarking. Viitattu 31.3.2020. <https://www.uef.fi/benchmarking>

Järvenpää, E. 2006. Laadullinen tutkimus. Viitattu 11.3.2020. <http://www.cs.tut.fi/~ihtesem/k2007/materiaali/luento4.pdf>

Keski-Uudenmaan pelastuslaitos. 2020. Viitattu 14.3.2020 <https://www.ku-pelastus.fi/>

Peliviikko. 2013. Pelikasvattajankäsikirja. Viitattu 10.3.2020. <https://peliviikko.fi/pelikasvattajankasikirja.pdf>

Routio, P. 2007. Tuote ja tieto: Kyselevät tutkimustavat. Viitattu 11.3.2020.

<http://www2.uiah.fi/projects/metodi/064.htm>

Saavutettavuus. 2020. Tietoa saavutettavuudesta. Viitattu 31.3.2020. <https://www.saavutettavuusvaatimukset.fi/tietoa-saavutettavuudesta/>

Schwaber, K. & Sutherland, J. 2017. Scrum-opas, scrumin määritelmä ja pelisäännöt. Viitattu 24.3.2020. <https://scrumwell.files.wordpress.com/2018/03/2017-scrum-guide-fi-v102.pdf>

Kuviot

Kuvio 1: HTML-tunnisteen rakenne.....	9
Kuvio 2: CSS säännön rakenne	9
Kuvio 3: Pelaajatyyppien motivoiminen	14
Kuvio 4: Aihealueen valintasivu	21
Kuvio 5: Kysymyssivu	22
Kuvio 6: Palautesivu.....	22
Kuvio 7: Kysymysten rakenne	23

Liitteet

Liite 1: Kysely	33
Liite 2: Haastattelukysymykset	34

Liite 1: Kysely

Paloturvallisuuspelin palautekysely

1. Millä luokalla olet koulussa?

3. luokalla
 4. luokalla
 5. luokalla
 jollain muulla, millä?

2. Olivatko pelin kysymykset mielestäsi

Helppoja Sopivia Vaikeita

3. Opitko pelin avulla tärkeitä turvallisuustaitoja?

En ollenkaan Vähän Paljon

4. Mitä olet mieltä pelin kuvista ja ulkonäöstä?

Heikko Hyvä Tosi hieno

5. Mikä pelissä oli tylsintä?

6. Mikä pelissä oli kivointa?

7. Jos haluat antaa vinkin, miten pelistä tekisi vielä paremman, voit kirjoittaa sen tähän:

Liite 2: Haastattelukysymykset

Haastattelukysymykset

1. Miten selkeitä värit ovat mielestäsi?
2. Miten erotat värit toisistaan?
3. Minkä värisinä näet värit?
4. Mitä muutoksia tekisit väreihin?
5. Onko muita huomioita, joita tulisi ottaa huomioon.