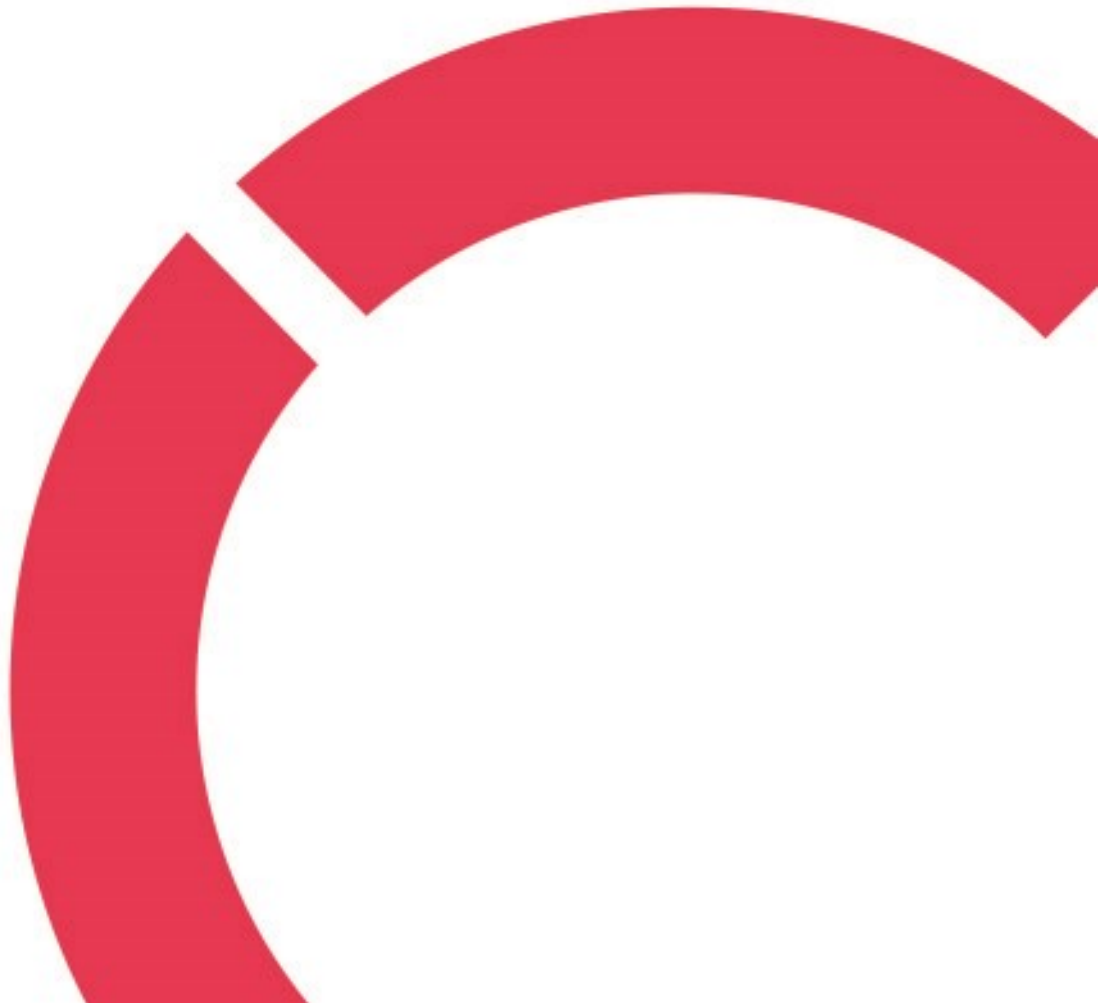


**Jani Järvi**

**KUINKA PELILLISTÄMISEN AVULLA VOIDAAN PARANTAA  
KÄYTTÄJÄKOKEMUSTA JA KÄYTTÖLIITTYMÄÄ**

**Opinnäytetyö  
CENTRIA-AMMATTIKORKEAKOULU  
Insinööri, tieto- ja viestintäteknikan koulutus  
Maaliskuu 2023**



**TIIVISTELMÄ OPINNÄYTETYÖSTÄ**

<b>Centria-ammattikorkeakoulu</b>	<b>Aika</b> Maaliskuu 2023	<b>Tekijä/tekijät</b> Jani Järvi
<b>Koulutus</b> Insinööri, tieto- ja viestintätekniikka		<input checked="" type="checkbox"/> AMK  <input type="checkbox"/> YAMK
<b>Työn nimi</b> KUINKA PELILLISTÄMISEN AVULLA VOIDAAN PARANTAA KÄYTTÄJÄKOKEMUSTA JA KÄYTTÖLIITTYMÄÄ		
<b>Työn ohjaaja</b> Jari Isohanni		<b>Sivumäärä</b> 18
<b>Työelämäohjaaja</b> Jussi Rajaniemi		
<p>Käyttöliittymien ja palveluiden pelillistäminen on yleistynyt lähivuosina paljon ja sen avulla on mahdollista tuottaa lisäarvoa sen käyttökohteille. Tuotettu lisäarvo ilmenee parantuneena käyttäjäkokemuksena. Pelillistämällä tarkoitetaan pelillisten elementtien hyödyntämistä pelien ulkopuolella.</p> <p>Tässä opinnäytetyössä käsitellään sitä, kuinka pelillistämistä suoritetaan ja kuinka sitä voidaan hyödyntää osana verkkoselaimessa toimivia käyttöliittymiä sekä palveluita. Opinnäytetyössä tarkastellaan pelillistämisen suunnitteluprosessia sekä toteutusta. Toteutuksen kannalta oleellisia teknologioita olivat JavaScriptiin pohjautuvat React-kirjastot sekä Next.js-kehityskehys.</p> <p>Opinnäytetyön käytännön osuus suoritettiin Alfame Systems Oy:n DigiFinlandille tuottamassa sähköinen perhekeskus-projektissa. Projektin tavoitteena oli luoda verkkoselaimessa toimiva käyttöliittymä, joka sisältää perheille sekä nuorille suunnattua informaatiota ja palveluita.</p> <p>Projektin pilotoinnin ja siitä kerätyn palautteen pohjalta ei ole mahdollista sanoa varmaksi, onnistuttiinko alussa määritellyt tavoitteet saavuttamaan. Pilotoinnista kerätty palaute ei ollut tarpeeksi kattava ja kysymyksiä pelillistämiseen liittyen oli liian vähän. Tästä huolimatta palaute oli pääosin positiivista ja pelillisistä elementeistä pidettiin. Jatkokehityksen kannalta pelillisten elementtien käyttämistä päätettiin jatkaa ja pelillistämistä arvioivien kysymysten määrää tullaan kasvattamaan ennen seuraavaa pilotointia.</p>		
<b>Asiasanat</b> Helpperi, käyttäjäkokemus, käyttöliittymä, pelillistäminen, sähköinen perhekeskus		

**ABSTRACT**

<b>Centria University of Applied Sciences</b>	<b>Date</b> March 2023	<b>Author</b> Jani Järvi
<b>Degree programme</b> Bachelor of Engineering, Information and Communications Technology		
<b>Name of thesis</b> HOW GAMIFICATION CAN BE USED TO IMPROVE THE USER EXPERIENCE AND USER INTERFACE		
<b>Centria supervisor</b> Jari Isohanni	<b>Pages</b> 18	
<b>Instructor representing commissioning institution or company</b> Jussi Rajaniemi		
<p>Gamifying user interfaces and services has become more popular during the past few years and by doing so companies are able to produce added value to their products. The added value manifests itself as an improved user experience. The definition of gamifying is to use game-like elements in another context than games.</p> <p>The aim of this thesis was to examine how the gamifying process is implemented and how it can be used as a part of making better user interfaces and services. The thesis focuses heavily on the gamifying planning process and also how it is implemented in the user interface. Essential technologies for the practical part of the thesis are JavaScript based React libraries and Next.js framework.</p> <p>The practical part of thesis was carried out as a part of Electric family centre project for DigiFinland which was carried out by Alfame Systems Oy. The goal of the project was to produce a user interface for youth and families that would contain information and services that might be helpful for them in many different parts of their lives.</p> <p>Upon evaluating the initial pilot phase of the project and the subsequent feedback obtained, it remains inconclusive whether the pre-defined objectives were successfully met. The feedback from the piloting was not comprehensive enough, and there were too few questions related to gamification. Nevertheless, the majority of the feedback was favourable, and there was a distinct appreciation for the gamified elements. For further development, the use of gamified elements will continue, and the number of questions evaluating gamification will be increased before the next pilot.</p>		
<b>Key words</b> electric family centre, gamification, Helpperi, user experience, user interface		

## **KÄSITTEIDEN MÄÄRITTELY**

### **CSS-TYYLITTELYKIELI (CASCADING STYLE SHEETS)**

CSS-tyylittelykieli on tarkoitettu visuaaliseen kehittämiseen. Tällä teknologialla voidaan mm. muokata HTML-elementtien värejä, fontteja sekä muita tyylejä.

### **FRONTEND**

Ohjelmiston, palvelun tai käyttöliittymän visuaalinen osuus. Sisältää myös logiikkaa ja toiminnallisuutta.

### **HELPPERI**

Helpperi on nuorille suunnattu asiointikokonaisuus osana DigiFinlandin sähköistä perhekeskusta.

### **HTML-MERKINTÄKIELI (HYPERTEXT MARKUP LANGUAGE)**

HTML eli hypertekstin merkintäkieltä käytetään käyttöliittymien rakentamiseen. Sen avulla luodaan rakenteellinen struktuuri esim. nettisivulle.

### **JAVASCRIPT-OHJELMOINTIKIELI**

JavaScript-ohjelmointikielellä tehdään käyttöliittymistä dynaamisempia ja lisätään interaktiota. Tällaisia asioita voivat olla erilaiset visuaaliset elementit, automatisoidut toiminnallisuudet sekä kommunikointi palvelimen ja käyttöliittymän välillä.

### **KÄYTTÖLIITTYMÄ**

Käyttöliittymä on tietotekniikassa, väline tai mekanismi, jonka avulla käyttäjä vuorovaikuttaa järjestelmän kanssa. Käyttöliittymä voi olla esimerkiksi nettisivu, joka on esimerkki graafisesta käyttöliittymästä.

### **LOPPUKÄYTTÄJÄ**

Loppukäyttäjällä tarkoitetaan lopullista kohderyhmää, joka tulee hyödyntämään tuotettua ohjelmistoa tai palvelua.

## **MUI (MATERIAL UI)**

Material UI on React-komponenttikirjasto, joka sisältää useita valmiiksi rakennettuja komponentteja, joita käytetään hyödyksi käyttöliittymien rakentamisessa.

## **PILOTTI / PILOTOINTI**

Pilotoinnilla tarkoitetaan kehitetyn ohjelmistoprototyypin laajempaa testikäyttöönottoa, jossa kohde-ryhmä pääsee käyttämään tuotettua palvelua ja antamaan palautetta. Pilotoinnista saadulla palautteella on tärkeä merkitys tuotetta ohjaavan kehityksen kannalta.

## **SÄHKÖINEN PERHEKESKUS (SPK)**

Sähköinen perhekeskus, eli SPK, on Alfame Systems Oy:n kehittämä palvelukokonaisuus DigiFinlandille. Sähköinen perhekeskus koostuu palveluista kuten Omaperhe.

**TIIVISTELMÄ**  
**ABSTRACT**  
**KÄSITTEIDEN MÄÄRITTELY**  
**SISÄLLYS**

<b>1 JOHDANTO</b> .....	<b>1</b>
<b>2 PELILLISTÄMINEN</b> .....	<b>2</b>
2.1 Pelillistämisen elementtejä .....	3
2.2 Pelillistämisen käyttökohteet ja hyödyt .....	4
2.3 Pelillistämisprosessi .....	5
<b>3 TARVITTAVAT TEKNOLOGIAT</b> .....	<b>8</b>
3.1 HTML-ohjelmointikieli .....	9
3.2 CSS-ohjelmointikieli.....	9
3.3 JavaScript-ohjelmointikieli .....	9
3.4 React-kirjasto .....	10
3.5 Material UI (MUI).....	10
3.6 Next.js .....	11
<b>4 SÄHKÖISEN PERHEKESKUKSEN PELILLISTÄMINEN</b> .....	<b>12</b>
4.1 Pelillistämisen suunnittelu.....	13
4.2 Pelillistämisen toteuttaminen .....	14
<b>5 HELPPERIN PILOTTI &amp; TULOKSET</b> .....	<b>17</b>
<b>6 JOHTOPÄÄTÖKSET &amp; POHDINNAT</b> .....	<b>18</b>
<b>LÄHTEET</b> .....	<b>19</b>
<b>KUVAT</b>	
KUVA 1. Pelillistämisprosessin vaiheet .....	7
KUVA 2. Tarinakomponentti Helpperin käyttöliittymässä .....	16

## 1 JOHDANTO

Tässä opinnäytetyössä tutkitaan, mitä tarkoittaa pelillistäminen sekä mitä hyötyä sen käyttämisestä käyttöliittymissä voi olla. Voidaanko pelillistämisen avulla esimerkiksi parantaa käyttäjäkokemusta tai sitouttaa loppukäyttäjä hyödyntämään palvelua helpommin? Opinnäytetyön toiminnallinen osuus toteutettiin osana DigiFinlandin sähköistä Perhekeskusprojektia, jonka alihankkijana toimi Alfame Systems Oy.

Opinnäytetyössä tarkastellaan erilaisia pelillistämiseen liittyviä käsitteitä sekä näkökantoja sekä sitä, minkälaisia teknologioita toiminnallisen osuuden suorittaminen vaatii. Kyseisiä teknologioita ovat mm. Reactiin pohjautuvat kirjastot, HTML-merkintäkieli, CSS-tyylittelykieli ja JavaScript-ohjelmointikieli. Työn toteuttaminen vaatii myös ymmärrystä käyttöliittymien suunnittelusta sekä saavutettavuuden parantamisesta.

Pelillistämällä tarkoitetaan pelillisten elementtien sekä tekniikoiden hyödyntämistä erilaisissa ympäristöissä ja tilanteissa pelien ulkopuolella. Pelillistämisen avulla tavoitellaan parempaa käyttäjäkokemusta ja pyritään sitouttamaan käyttäjä hyödyntämään palvelua. Pelillistämistä voidaan mm. käyttää hyödyksi yritys- tai opetustoiminnassa ja erilaisissa käyttöliittymissä. Pelillistämisen avulla voidaan myös edistää hyvinvointia, uuden oppimista, motivaatiota ja markkinointia. (Hamari 2015.)

## 2 PELILLISTÄMINEN

Pelillistämällä tarkoitetaan pelillisten elementtien käyttämistä muihin asioihin kuin peleihin. Pelillistämällä tavoitellaan samankaltaisia positiivisia psykologisia vaikutuksia kuin pelien pelaamisella voisi olla ja parannettua käyttäjäkokemusta. Erityisesti pelillistämällä voidaan parantaa käyttäjien motivaatiota ja sitoutumista tuotteen tai palvelun käyttämiseen. Pelillistäminen voi olla esimerkiksi erilaisten pistetaulukoiden tai sosiaalisen median kaltaisten tarinallisten elementtien käyttämistä ja pelillistäminen voidaan jakaa kolmeen pääkategoriaan elementtien ominaisuuksien mukaan. Pelilliset elementit voidaan siis jakaa immersiota, suoriutumista tai sosiaalisuutta vahvistaviin kategorioihin. (Hamari 2015.)

Pelillistämisen tuottamia psykologisia tunteuksia on monia ja muutamia niistä ovat yhteenkuuluvuuden, vapaaehtoisuuden ja tavoitteellisuuden sekä pätevyyden tunne. Vapaaehtoisuudella tarkoitetaan asiayhteydessä sitä, että käyttäjä ei koe painostusta tehdä jotain asiaa vaan hänellä on valinta myös jättää joitain asioita tekemättä. Tavoitteellisuus ja pätevyyden tunne taas muodostuvat usein hyödyntäen erilaisia pisteityksiä, jotka ovat todella yleisiä peleissä. Käyttäjä voi esimerkiksi saada automatisoidun kehun tai erityisen maininnan hyvin suoritetusta edistymisestään, mikä muodostaa käyttäjälle pätevyyden tunteen. Pelillistämällä voidaan luoda vahva immersio, joka olennaisesti liittyy myös asioiden luontaiseen ja optimoituun kulkuun. Tätä hyödyntämällä asiat tuntuvat etenevän kuin itsestään ja käyttäjä ei tunne itseään eksyneeksi järjestelmässä. (Hamari 2015.)

Pelillistämisen avulla on mahdollista kehittää palveluiden ja käyttöliittymien laatua, koska siinä hyödynnetään asiakaslähtöistä palvelulogiikkaa. Tällä tavoin erilaisten käyttäjien tarpeet on mahdollista huomioida paremmin ja lopullisesta tuotteesta tai palvelusta tulee toimiva kokonaisuus. Palvelu, jonka luomiseen on hyödynnetty pelillistämistä, tulisi tuntua mielekkäältä ja luonnolliselta käyttää. (Hamari 2015.)



## 2.1 Pelillistämisen elementtejä

Pelillistämistä voi tehdä usealla eri tavalla, mutta tärkeintä on kartoittaa, mitkä tekniikat ovat parhaimmat asiayhteyteen. Oikeiden elementtien valintaan vaikuttaa myös kohderyhmä, jolle tuotetta tai palvelua ollaan tuottamassa, sillä eri ikäluokat arvostavat erilaisia asioita. Erityisen suosittuja pelillisiä elementtejä ovat mm. pistejärjestelmät, tarinallistaminen ja tehtävät. (Xi & Hamari 2019.)

Pelillistämiselementit voidaan jakaa kolmeen selkeään kategoriaan niiden piirteiden mukaan, ja nämä kategoriat koostuvat immersio-, suoritus- ja sosiaalipohjaisista ominaisuuksista. Immersiokategoriaan luokitellaan esimerkiksi tarinallisuus. Tarinallisuuden avulla on etenemisen ja kokonaisuuden hahmottaminen helpompaa loppukäyttäjällä, koska tarinan kerronta noudattaa selkeää ja helposti seurattavaa kaavaa. Muita immersiota vahvistavia elementtejä ovat erilaiset käyttäjähahmot, personalisointimahdollisuus, suoritusmerkit, visualisoinnit ja pistejärjestelmät. Immersiokategorian elementit keskittyvät erityisesti vahvistamaan loppukäyttäjän tunnetta itsehallinnostaan, eli he kokevat pystyvänsä vaikuttamaan riittävän paljon prosessin kulkuun ja sen etenemiseen. (Xi & Hamari 2019.)

Toinen pääkategoria kohdistuu suoritusta vahvistaviin ominaisuuksiin. Tämän kategorian elementit vaikuttavat vahvasti siihen, kuinka loppukäyttäjä kokee suoriutuvansa palvelun käytöstä ja elementit voivat myös vahvistaa autonomian tunnetta. Tällaisia elementtejä ovat mm. edistymismittarit, tehtävät, sosiaaliset verkostot ja pistelistat. Nämä elementit auttavat siis visualisoimaan käyttäjän edistymistä ja ne myös voivat johdattaa käyttäjää oikeaan suuntaan perustuen yksilöityihin tarpeisiin. Pistelistat voivat toimia hyvänä motivaattorina kilpailuhenkisille ihmisille, jolloin asioiden edistäminen tuntuu mielekkäämmältä. (Xi & Hamari 2019.)

Viimeinen pääkategoria koostuu sosiaalisuuteen pohjautuvista elementeistä. Peleissä usein muodostetaan tiimejä, joissa tehdään yhteistyötä ja näin luodaan yhtenäisyyden tunnetta. Erilaiset tiimit tai yksilöt voivat myös kilpailla keskenään, mikä lisää käyttäjien kanssakäymistä. Yhteistyö ja sosiaalinen kilpailu kahdestaan muodostavat tämän pääkategorian. Niiden avulla pyritään saamaan käyttäjän olotila rennoksi ja avoimeksi. Loppukäyttäjän mielentilalla palvelua käyttäessä voi olla suuri merkitys käyttäjäkokemukseen ja rentoutuneessa mielentilassa asioiden käsittely on huomattavasti mielekkäämpää. (Xi & Hamari 2019.)

## 2.2 Pelillistämisen käyttökohteet ja hyödyt

Pelillistämisen suosio erilaisten palveluiden kehittämisessä on ollut kasvussa viime vuosina. Onnistuneella pelillistämällä voidaan tuottaa lisäarvoa ja huomattavasti parantaa käyttäjäkokemusta. Pelillistämistä voidaan esimerkiksi hyödyntää opetuksessa, terveydenhuollossa, monimuotoisissa palveluissa sekä sähköisissä käyttöliittymissä. (Morschheuser, Hassan, Werder & Hamari 2018.)

Pelillistämistä harkittaessa on kuitenkin tärkeä kartoittaa, mitä ollaan pelillistämässä. Pelillistäminen on tutkitusti antanut positiivisia tuloksia oppimisen edistämässä ja parantanut käyttäjäkokemusta sähköisissä käyttöliittymissä. Pelillistämistä ei kuitenkaan kaikkiin palveluihin kannata toteuttaa, koska virallisemmat palvelut voivat menettää uskottavuuttaan, jos ne lisäävät palveluihinsa tiettyjä pelillisiä elementtejä. Vaatimusten määrittely ja kohderyhmän valinta ovat siis kriittinen osa valittaessa käyttökohdetta, jota halutaan pelillistää. (Morschheuser ym. 2018.)

Pelien pelaamisella on useita vaikutuksia ihmiseen, mutta suurin osa niistä on psykologisia tai sosiaalisia vaikutuksia. Näitä samoja pelien tuottamia positiivisia vaikutuksia tavoitellaan myös pelillistämällä. Tällaisia positiivisia vaikutuksia ovat mm. motivaation kasvaminen, yhtenäisyyden tunne, rentoutuminen ja asiayhteyteen sitoutuminen taikka syvällisempi keskittyminen. Tutkimukset ovat myös osoittaneet, että pelillistämällä voi olla positiivinen vaikutus markkinointiin ja sen kasvamiseen. Kokonaisuudessaan pelillistämällä pyritään siis tyydyttämään loppukäyttäjien tarpeet mahdollisimman kattavasti. (Xi & Hamari 2019.)

## 2.3 Pelillistämisprosessi

Pelillistäminen on haastava ja pitkä prosessi, johon liittyy monia huomioon otettavia asioita, kuten pelillistettävä kohde, kohderyhmä ja se, minkälaista lisäarvoa pelillistämisen tavoitellaan kyseisessä tapauksessa. Pelien pääasiallinen tehtävä on käyttäjän viihdyttäminen, mutta pelillistämisen tavoitellaan usein muutakin kuin pelkkää viihdyttämistä. Tämä tekee pelillistämisen prosessista vaikeaa, koska alun perin puhtaasti viihdyttämiseen suunniteltuja elementtejä pyritään hyödyntämään pelien ulkopuolella. Hyvin suunniteltuna ja sitten toteutettuna pelillistäminen voi kuitenkin tuoda huomattavaa lisäarvoa käyttökohteelle. (Morschheuser ym. 2018.)

Jokainen pelillistämisen prosessi alkaa vaatimusten määrittelyllä. Tässä vaiheessa kartoitetaan, mitä halutaan saavuttaa, kuinka saavutettavat asiat priorisoidaan ja tuovatko halutut ominaisuudet riittävästi lisäarvoa käyttökohteelle, jotta prosessin suorittaminen olisi kannattava. Nämä työvaiheet ovat elintärkeitä dokumentoida, jotta haviteltu lopputulos on mahdollisimman selkeä ja kaikki prosessiin osallistujat ymmärtävät, mitä he ovat toteuttamassa ja miksi. (Morschheuser ym. 2018.)

Vaatimustenmäärittelyn jälkeen suoritetaan yleistä tutkimusta aiheeseen liittyen ja selvitetään, kuinka vaiheiden onnistumista voidaan mitoittaa. Aiheen riittävä ymmärrys ja tutkiminen on esivaatimus onnistumisen mitoittamisen kannalta. On mahdotonta tietää, onko lisäarvoa tuotettu onnistuneesti, jos ymmärrys aiheesta ei ole riittävä ja sen seuraamiseen ei ole luotu mittareita. (Morschheuser ym. 2018.)

Edellä mainittujen toimenpiteiden jälkeen hahmotellaan haluttu kohderyhmä, joka tulee käyttämään tuotettavaa palvelua tai tuotetta. Valittu kohderyhmä muovaa suuresti tuotetun palvelun ulkonäköä ja vaatimuksia, koska eri ikäluokilla on täysin erilaiset tarpeet, jotka täytyy tyydyttää. Haluttua kohderyhmää voidaan mallintaa luomalla keksittyjä henkilöitä ja määrittelemällä mitkä heidän motivaationsa ovat ja miksi he käyttäisivät tuotettua palvelua. Tällä tavalla hahmotetaan vastaavatko aiemmin määritellyt vaatimukset halutun kohderyhmän tarpeita ja onko tarvetta palata prosessissa taaksepäin. Kohderyhmän mallintamisen lisäksi organisaation täytyy ottaa myös huomioon toteutuksen vaatimat teknologiset rajoitteet ja onko pelillistäminen oikea ratkaisu heidän tuottamaansa palveluun. (Morschheuser ym. 2018.)

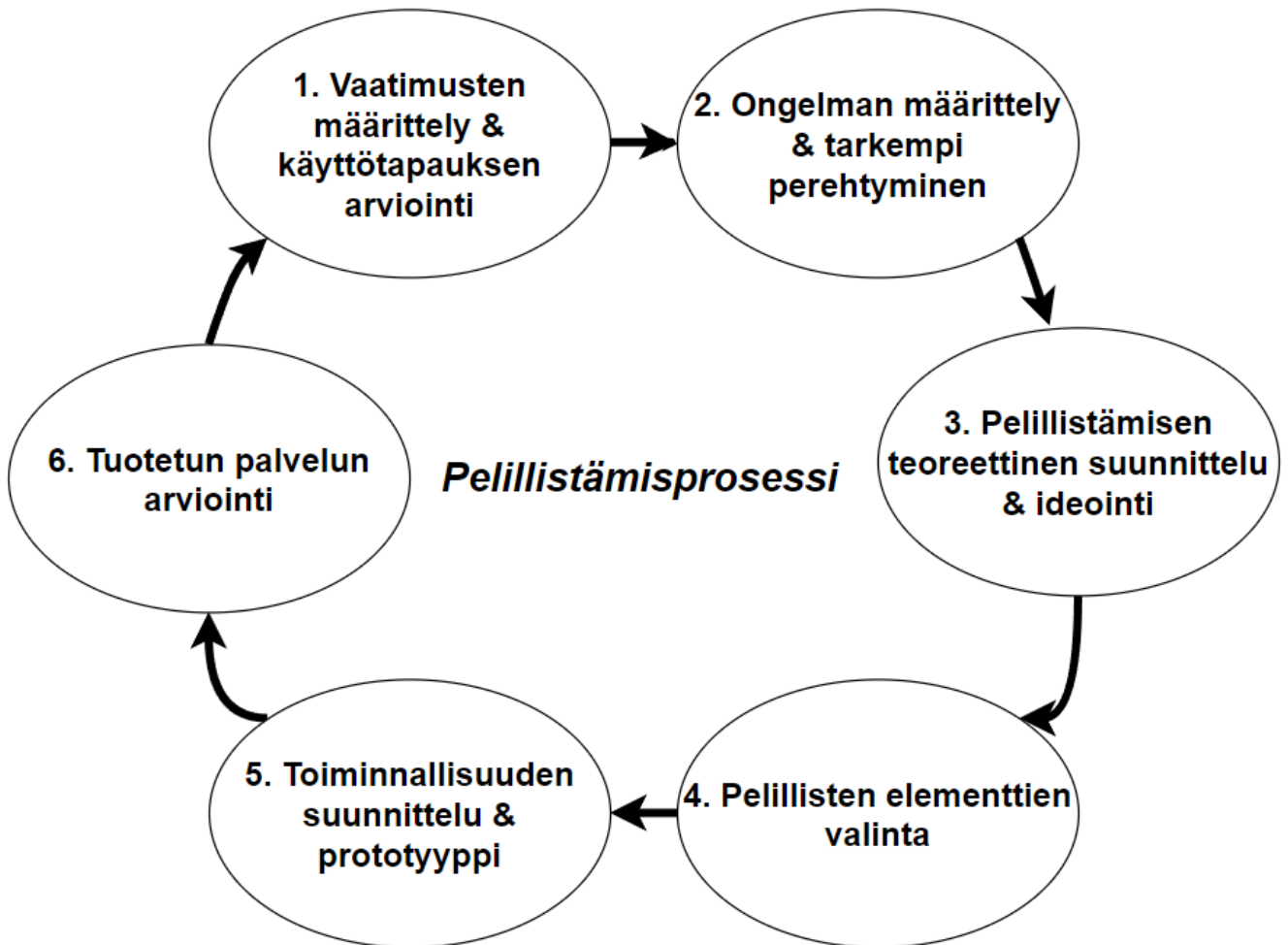
Seuraavaksi alkaa ideointivaihe, jossa palvelun tuottajat yrittävät tuottaa alustavia hahmotelmia ja ideoita siitä, kuinka pelillistäminen toteutetaan. Mikäli vaatimukset ja kohderyhmä on onnistuneesti ra-

jattu on oikeiden elementtien valinta helpompaa. Ideoinnissa hyödynnetään ideointipajoja, joissa palvelun tuottajat jakaantuvat ryhmiin, joissa he kehittävät erilaisia ideoita siitä, kuinka vaatimusten mukaista pelillistämistä voitaisiin toteuttaa. Nämä ideat kootaan lopuksi yhteen, minkä jälkeen ne käydään koko ryhmän kesken lävitse. Tässä vaiheessa valitaan parhaimmat ideat yleensä äänestämällä, minkä jälkeen niitä lähdetään jalostamaan pidemmälle. Ideointiin on olemassa monia valmiita viitekehityksiä ja työkaluja, joita kannattaa hyödyntää. Tällaisia ideoinnin työkaluja ovat mm. suunnittelupelit sekä -kortit, erilaiset visualisoinnit ja tarinan kerrontatyökalut, jotka voivat sisältää valmiita kysymyksiä, joiden pohjalta kehittäjät lähtevät muodostamaan omaa tuotekehityspolkuaan. (Morschheuser ym. 2018.)

Kun ideoinnista on syntynyt pitemmälle jalostettuja tapoja liittää pelillisiä elementtejä osaksi palvelua, on aika luoda alustava asiakaspolku. Asiakaspolulla kuvataan loppukäyttäjän matka palvelun ensimmäisestä havaitsemisesta käyttöönottoon ja palvelun pitempi aikaiseksi hyödyntäjäksi. Tämän jälkeen on viimein aika luoda ensimmäinen prototyyppi määrittelyjen pohjalta, jota aloitetaan kehittämään. Prototyypin kehityksen edetessä on tärkeää antaa tuote testattavaksi halutulle kohderyhmälle varhaisessa vaiheessa, jotta saadun palautteen perusteella kehitystä pystytään viemään tarvittavaan suuntaan ja turhalta työltä välttyttäisiin. Prototyyppiä kehittäessä täytyy sille myös määritellä varhaisessa vaiheessa budjetti, aikataulut ja sopivan kokoiset kehitysryhmät, jotta prosessi pysyisi kustannustehokkaana. Prototyypin elinkaari huipentuu lopuksi pilotointiin, jossa kehitetty palvelu asetetaan saatavaksi pienelle kohderyhmälle ja kohderyhmältä pyydetään palautetta siitä, vastaako tuotettu kokonaisuus heidän vaatimuksiaan sekä tarpeita. (Morschheuser ym. 2018.)

Onnistuneen pilotoinnin jälkeen aloitetaan asiakokonaisuutta arvioimaan alussa määriteltyjen onnistumismittareiden mukaisesti. Mikäli saavutettu tulos ja palaute vastaavat määrittelyjä, voidaan tuotteen pelillistäminen todeta onnistuneeksi. Tästä alkaa tuotteen julkaisuprosessi ja laajempi käyttöönotto. Käyttöönoton jälkeen alkaa ylläpitovaihe, jossa tarkkaillaan palvelun toimintaa ja suoritetaan tarvittaessa korjauksia ja lisätään uusia ominaisuuksia, mikäli ne koetaan kuuluvaksi osaksi kyseistä tuoteperhettä. Jos arviointi hylätään, palataan takaisin prosessin alkuun ja puututaan ongelmakohtiin. (Morschheuser ym. 2018.)

Kuva pelillistämisen prosessin vaiheista ja niiden etenemisjärjestyksestä:



KUVA1. Pelillistämisen prosessin vaiheet (mukaillen Antonaci, Klemke, Kreijns & Specht. 2018.)

### 3 TARVITTAVAT TEKNOLOGIAT

Opinnäytetyön toiminnallinen osuus vaatii laajan kirjon osaamista erilaisista teknologioista. Käytettävät teknologiat sekä tekniikat pohjautuvat web-pohjaisten ratkaisujen kehittämiseen, joita ovat mm. HTML, CSS, JavaScript sekä React. Toiminnallisen osuuden suorittaminen vaatii myös perustason ymmärtämistä käyttöliittymien suunnittelusta ja saavutettavuuden parantamisesta, jotta laajempi kohderyhmä on mahdollista tavoittaa tuotetulla palvelulla.

Työn tekemisen kannalta tarvittavia ohjelmistoja ovat myös jonkinlainen ohjelmointieditori, kuten Visual Studio Code, sekä ympäristö, jossa ylläpidetään versiohallintaa. Versiohallinta on keskeinen osa ohjelmistotuotantoa, jotta työn kehitystä voidaan seurata ja viimeisimmät toimivat versiot pysyvät saatavilla, mikäli parhaillaan kehitettävä versio hajoaisi. Opinnäytetyössä versiohallintaa suoritettiin Github palvelun avulla ja hyödyntäen git-komentoja.

Git on versiohallintatyökalu, jota hyödynnetään ohjelmistokehityksessä. Sen avulla on mahdollista parantaa tuotekehityksen laatua, helpottaa ylläpidettävyyttä ja suorittaa ohjelmistokehitystä ketterästä tiimeissä. Gitin avulla voidaan yhdelle ohjelmistoprojektille luoda useita kehityshaaroja, joita eri kehittäjät voivat edistää omaan tahtiinsa ja myöhemmin ne on mahdollista yhdistää osaksi projektin päähaaraa, kun ne todennetaan toimiviksi ja valmiiksi. Git-ympäristö sisältää myös integroitua laadunvalvontaa, koska kehittäjät voivat muutoksia tehdessään pyytää tiimin jäseniään suorittamaan vertaisarviointia ja ilman vertaisarviointia ei uutta toiminnallisuutta voida liittää osaksi toimivaa päähaaraa. Näin tuotetun koodin tarkistaa useampi kehittäjä, jolloin myös havaitaan virheet helpommin ja korjauksia voidaan tehdä. (Github, Inc 2023.)

Git-komentojen avulla kehittäjät hallinnoivat projektin paikallista versiota. GitHubissa oleva projekti on mahdollista kloonata paikallisesti tietokoneelle hyödyntäen git-komentoja. Projektin kloonauksen jälkeen voi kehittäjä tehdä pilvestä kloonattuun ohjelmistoon haluamiaan muutoksia, niin etteivät ne vaikuta GitHubissa olevaan toimivaan versioon. Kun kehittäjä uskoo muutoksiansa olevan valmiita liitettäväksi osaksi päähaaraa, hän voi git-komentojen avulla luoda paikallisista muutoksistaan paketin, jonka hän lähettää vertaisarvioitavaksi GitHubiin tiimiläisilleen. (Github, Inc 2023.)

### 3.1 HTML-ohjelmointikieli

HTML, eli HyperText Markup Language, on ohjelmointikieli, joka on web-käyttöliittymien rakentamisen kannalta olennaisin työkalu, koska sen avulla luodaan käyttöliittymän rakenne. HTML-ohjelmointikieli koostuu useista elementeistä, joilla kaikilla on uniikki tehtävänsä rakenteessa. Näitä elementtejä ovat esimerkiksi head, body, div ja nav. Elementit täytyy aina sulkea, niin sanottujen tagien sisään, eli elementin pelkistetyin rakenne käyttämisen kannalta olisi ”<elementti>”. Käytetyllä elementillä täytyy olla avaus- ja sulkuelementti, mutta tähän on poikkeuksia elementin mukaan. Sulkuelementti on muotoa ”</elementti>”. (Mozilla mdn web docs 2023b.)

### 3.2 CSS-ohjelmointikieli

CSS-ohjelmointikieli tulee sanoista Cascading Style Sheets ja kyseistä ohjelmointikieltä käytetään käyttöliittymien tyyliasetusten määrittämiseen. Tyylimääryksiä käytetään erilaisten HTML-elementtien muokkaamiseen. CSS-ohjelmointikielen avulla on esimerkiksi mahdollista muokata elementtien värejä, kokoja ja rivityksiä. Myös sisällön animointi on mahdollista kyseisellä ohjelmointikielillä. Kielen pääasiallinen tehtävä on siis muodostaa HTML-rakenteen visuaalinen ulkoasu. (Mozilla mdn web docs 2023a.)

Toinen tärkeä CSS-ohjelmointikielen tehtävä on parantaa käyttöliittymän responsiivisuutta. Responsiivisuudella tarkoitetaan, että käyttöliittymä toimii tarkoitetulla tavalla, vaikka näytön koko vaihtelisi, jolla käyttöliittymää käytetään. Näin parannetaan palvelun saavutettavuutta, koska käyttöliittymää voidaan katsella tietokoneelta tai puhelimelta. Käyttöliittymän suunnittelussa on siis tärkeää määritellä, kuinka suurelle kohderyhmälle palvelu halutaan tuottaa, jotta rakenteellinen kartoitus on mahdollista tehdä aikaisin. (Mozilla mdn web docs 2023d.)

### 3.3 JavaScript-ohjelmointikieli

JavaScript-ohjelmointikieli on yksi yleisimmistä skriptien kirjoitus ohjelmointikielistä ja sitä käytetään vahvasti käyttöliittymien, että muiden ohjelmistojen rakentamisessa. JavaScriptin kirjoittaminen sisäl-

tää paljon erilaisten funktioiden tekemistä, oliopohjaista ohjelmointia ja toiminnallisuuksien parametrisointia. JavaScriptin monipuolisuus ja jatkuva kehittyminen on johtanut monien erilaisten JavaScript-pohjaisten valmiiden ohjelmistokirjastojen syntymiseen, joiden avulla ohjelmistokehityksestä voidaan tehdä tehokkaampaa. JavaScriptiä ei kuitenkaan tule sekoittaa Java-ohjelmointikieleen, sillä ne ovat kaksi täysin erilaista ohjelmointikieltä. (Mozilla mdn web docs 2023c.)

Tämän opinnäytetyön kannalta tärkeässä roolissa ovat erityisesti aiemmin mainitut valmiit JavaScript-pohjaiset kirjastot. Tuotettu käyttöliittymä rakentuu kokonaan React-ohjelmistokirjaston päälle ja kyseisen kirjaston käyttäminen käyttöliittymien rakentamisessa on muodostunut nykypäivänä standardiksi. Reactin avulla on mahdollista luoda dynaamisia ja monipuolisia käyttöliittymiä vastaamaan uusimpia vaatimuksia.

### **3.4 React-kirjasto**

React on JavaScriptiin pohjautuva ohjelmointikirjasto, joka on yhdistelmä JavaScriptiä sekä HTML:ää. React-kirjasto sisältää valtavan määrän valmiita komponentteja, joiden avulla rakennetaan tehokkaita sekä responsiivisia käyttöliittymiä. Reactin käyttämä syntaksi on nimeltään JSX ja kyseinen syntaksi eroaa hieman JavaScriptin syntaksista. Reactin hyödyntäminen käyttöliittymissä tuo palveluun joustavuutta sekä dynaamisuutta, koska se toimii hyvin muiden ohjelmointikielien kanssa. Reactin avulla voidaan rakentaa käyttöliittymiä sekä sovelluksia useille käyttöjärjestelmille, sillä se on yhteensopiva suosituimpien verkkoselainten ja käyttöjärjestelmien kanssa. Reactilla toteutetaan pääasiassa frontend-kehitystä, joten sen kanssa yleensä käytetään Next.js nimistä rajapintaa, jolla hoidetaan toiminnallisuutta, joka ei näy käyttäjälle. Tällainen toiminnallisuus voi olla esimerkiksi tietojen haku sekä käsittely. (Meta Open Source 2023.)

### **3.5 Material UI (MUI)**

Material UI on Reactista pohjautuva avoimen lähdekoodin komponenttikirjasto, joka sisältää suuren määrän valmiiksi rakennettuja komponentteja. Valmiit komponentit ovat kuitenkin joustavasti muokattavissa käyttötarkoituksen mukaisesti ja ne on rakennettu noudattaen Googlen määrittelemiä suunnitellustandardeja, joten kirjaston komponenttien laatu on korkea. Material UI:n suurin hyöty on, että hyö-



dyntäen valmiiksi rakennettuja ja laadukkaita komponentteja, on tehtävän työn määrää mahdollista vähentää ja kyseisen kirjaston käyttöönotto on helppoa sen kattavan dokumentaation vuoksi. Komponenttikirjasto kehittyy nopealla tahdilla ja pysyy siksi myös ajankohtaisena. Tämä on mahdollista, koska kyseinen kirjasto on avointa lähdekoodia ja täten suosittu ohjelmistokehittäjien keskuudessa. (Material UI SAS 2023.)

### 3.6 Next.js

Next.js on joustava sekä monipuolinen React-pohjainen kehityskehys, jonka avulla rakennetaan käyttöliittymiä. Next.js:n avulla voidaan mm. parantaa käyttöliittymien optimointia, ylläpitää koodin rakennetta ja pitää huolta konfiguraatioista, joita käyttöliittymä noudattaa. Next.js-kehityskehyksellä on mahdollista myös hoitaa tiedon hakua, parantaa kehittäjäkokemusta ja integroida kolmannen osapuolen palveluita osaksi omaa ohjelmistoa. (Vercel, Inc 2023.)

Tarkemmin selitettynä Next.js toimii käyttöliittymän ja serverin välisenä kommunikointivälineenä. Erityisen olennainen osa käyttöliittymää tai palvelua on usein jonkinlainen tietokanta, johon voi olla tallennettuna mm. käyttäjien tunnuksia ja salasanoja tai sivuilla näytettävää tietoa. Käyttöliittymä ja serveri eivät kuitenkaan ymmärrä toisiaan suoraan ja tätä varten niiden välistä kommunikointia hoitaa Next.js. (Vercel, Inc 2023.)

## 4 SÄHKÖISEN PERHEKESKUKSEN PELILLISTÄMINEN

Sähköinen perhekeskus on palvelukokonaisuus, jonka tarkoituksena on keskittää perheille ja nuorille suunnattua informaatiota heille räätälöityyn käyttöliittymään. Perhekeskus koostuu Omaperhe-palvelusta, ammattilaisten osiosta ja nuorille suunnatusta Helpperistä, jonka pelillistämiseen tämä opinnäytetyö keskittyy. Helpperistä nuoret voivat etsiä tietoa heitä askarruttaviin kysymyksiin ja vaihteleviin elämän tilanteisiin.

Helpperi-palvelu haluttiin pelillistää, jotta loppukäyttäjät, eli tässä kohderyhmänä nuoret, saataisiin sitoutettua palvelun pitkäaikaisiksi hyödyntäjiksi. Pelillistämällä pyritään myös parantamaan yleistä käyttäjäkokemusta. Palvelun pelillistämällä arvioitiin, että nämä vaatimukset olisi mahdollista saavuttaa ja lisätä sen suosiota nuorille suunnatussa palvelutarjonnassa.

Helpperi kokonaisuutena koostuu kahdesta eri käyttöliittymästä, joita ovat Helpperin sisällönhallinta ja portaali. Portaali on käyttöliittymä, joka on suunnattu loppukäyttäjille eli tässä tapauksessa nuorille, josta he pääsevät etsimään ja selaamaan tietoa. Sisällönhallinta on suunnattu Helpperin ylläpitäjille, ja siitä heidän on mahdollista vaikuttaa siihen, mitä tietoa nuorille näytetään portaalissa. Sisällönhallinnan avulla tietojen lisäämisestä, poistamisesta ja muokkaamisesta tehdään tehokkaampaa.

## 4.1 Pelillistämisen suunnittelu

Helpperin pelillistäminen alkoi vaatimusten määrittelyllä, koska muuta suunnitteluja työtä on hankala lähteä toteuttamaan ja kehittämään, jos tavoitteet eivät ole selkeitä tai niiden edistymistä ei pystytä seuraamaan. Vaatimusten määrittelyä johti tuoteomistaja ja apuna määrittelyissä olivat mukana myös muut projektin kannalta tärkeät henkilöt sekä kehittäjät. Helpperin pelillistämisen vaatimuksiksi määriteltiin, että pelillistämällä pyritään parantamaan yleistä käyttäjäkokemusta sekä sitouttamaan palvelun kohderyhmä tuotteen pitkäaikaisiksi hyödyntäjiksi. Pelillistämällä haluttiin myös kohdentaa Helpperi enemmän nuorille ja erottaa se selkeästi Omaperhe-palvelusta, koska pelillistämisen avulla ajateltiin, että nuorempi kohderyhmä on helpompi tavoittaa.

Vaatimusten määrittelyn jälkeen aloitettiin varsinainen suunnittelutyö, jonka tarkoituksena oli ideoita erilaisia keinoja, joilla Helpperi voitaisiin pelillistää. Tätä varten järjestettiin muutamia työpajoja, joihin projektin jäsenet kokoontuivat yhdessä ideoimaan, millaisia pelillisiä elementtejä kannattaisi Helpperiin valita ja olisiko niiden toteuttaminen realistisesti mahdollista. Työpajoissa mukana oli myös projektin ulkopuolisia pelillistämisen asiantuntijoita, jotka jakoivat omia näkemyksiään sekä pyrkivät näyttämään oikean suunnan kehitykselle. Asiantuntijoiden näkemysten lisäksi tutkivat myös projektin jäsenet omatoimisesti pelillistämistä ja millaisia asioita siihen kuuluu, jotta työpajat sujuisivat tehokkaammin. Työpajat tuottivat tulosta ja niistä nousi muutamia hyviä ideoita, joita päätettiin lähteä jalostamaan pidemmälle. Näihin ideoihin kuului mm. visuaalisten ja tarinallisten elementtien käyttöä, kuten paralaxit tai sosiaalisen median tapaiset päivitykset. Muita valittuja ideoita olivat käyttäjä-avatatit sekä mahdollisuus antaa reaktioita. Näistä ideoista ensimmäiseen pilottiin valittiin toteutettavaksi visuaalinen paralaxi sekä mahdollisuus esittää Instagram-tyylisiä tarinapäivityksiä Helpperin eri aiheiden ja teemojen yhteydessä.

Työpajojen ja ideoinnin jälkeen lähdettiin suunnittelemaan, kuinka pilottiin valitut pelilliset elementit voitaisiin toteuttaa. Toteutus tulisi vaatimaan niin frontend- kuin backend-kehittämistä, jotta toteutus olisi mahdollinen. Tärkeää oli myös hahmotella alustavat aikataulut, jotta kehitys etenisi riittävällä nopeudella ja oikea määrä resursseja osattaisiin varata työlle.

Frontend-työ tässä vaiheessa koostui alustavien käyttöliittymäkomponenttien mallintamisesta ja prototyyppien luomisesta. Määrittelyissä täytyi myös huomioida tarvittavien ohjelmointikielien sekä teknologioiden tarve ja löytyisikö niihin riittävästi osaamista tiimin sisältä. Määrittelyjen pohjalta arvioitiin,

että toteutus tulisi vaatimaan osaamista Reactista, JavaScriptistä, CSS:stä ja HTML:stä ja MUI-komponenttikirjaston käyttämisestä. Komponenttien ohjelmoinnin lisäksi täytyisi myös olla osaamista käyttöliittymien saavutettavuuden parantamisesta sekä visuaalisesta suunnittelemisesta.

Backend-työ koostui tietokantayhteyksien mallintamisesta ja suunnittelusta, kuinka palvelun ylläpitäjät pystyisivät lisäämään kuvia tai muuta mediaa portaaliin sisällönhallinnasta käsin. Pilotitiin tuotettavaa prototyyppiä varten, päätettiin kaikki kuvat sekä paralaxit sisällyttää suoraan käyttöliittymään, mutta tämä tulisi korjata pilotin jälkeen siten, että kaikki media haettaisiin käyttöliittymään dynaamisesti sisällönhallinnasta. Tämä parantaisi ohjelmiston laatua sekä käytettävyyttä pitemmällä aikavälillä.

Backend-työ tulisi siis vaatimaan riittävää ymmärrystä ja osaamista tietokannoista, ohjelmistoarkkitehtuurin suunnittelemisesta ja rakentamisesta sekä Next.js:n käyttämisestä. Next.js:n avulla siis liitettäisiin Helpperin käyttöliittymä keskustelemaan sisällönhallinnan mediapankkirajapinnan kanssa, joka sisältäisi kaikki mahdolliset kuvat sekä videot, joita liitoksen jälkeen olisi mahdollista käyttää portaalissa dynaamisesti.

## 4.2 Pelillistämisen toteuttaminen

Helpperin prototyyppiin tuotettiin kaksi staattista sivua, joiden sisällön yhteydessä demonstroitiin välittäviä pelillisiä elementtejä. Tämä tarkoitti sitä, että ratkaisu ei olisi hyvin ylläpidettävä, koska kaikki sisältö on suoraan rakennettu Helpperin käyttöliittymään eikä sitä voisi tarkastella taikka päivittää sisällönhallinnasta käsin. Kyseinen toteutustapa kuitenkin riitti pilotointia varten, jossa testaajaryhmän oli mahdollista saada käsitys siitä, millaista sisältöä sivuille olisi mahdollista tuottaa nuorten nähtäväksi. Ensimmäinen staattinen sivu sisälsi esimerkin Instagramin kaltaisen tarinan käyttämisestä, jonka avulla tietoa olisi mahdollista välittää nuorille visuaalisemmin ja monipuolisemmin. Toinen staattinen sivu sisälsi urheiluteemaisen paralaxin, joka elävöitti sivua tuomallaan visuaalisella olemuksella.

Edellä mainittujen sivujen rakentaminen tehtiin useassa osassa ja tähän työhön osallistui useita kehittäjiä. Jotta nämä sivut oli mahdollista toteuttaa, täytyi luoda useita komponentteja, joista rakennettiin koherentti kokonaisuus tyhjille staattisille sivuille palapelimaisesti. Osa tuotetuista komponenteista vastasi siitä, kuinka käyttöliittymän saama tieto esitettäisiin näytöllä ja muut komponentit käsitelivät sivuilla tarvittavaa logiikkaa sekä tiedonhakua. Komponenttien rakenne noudatti JSX-syntaksia, joka vaati niin HTML:n kuin JavaScriptin osaamista.

Sivuja suunniteltaessa otettiin myös huomioon erilaisia rajoitteita, jotka voisivat vaikuttaa sisällön saavutettavuuteen. Kaikille sivuille pyrittiin tuottamaan mahdollisuus selata sisältöä käyttäen pelkästään näppäimistöä sekä visuaalisesti näkymättömät tekstit, joita ruudunlukijat voivat hyödyntää, mikäli loppukäyttäjällä olisi näkörajoitteita. Tällä tavoin voitiin varmistaa, että yksi alussa määritelty tavoite täyttyy, mikä oli laajemman loppukohderyhmän tavoittaminen parantamalla myös käyttöliittymän saavutettavuutta.

Helpperiä rakennettaessa oli myös tärkeää seurata koko ajan, että tuotetut komponentit toimisivat pääasiallisesti oikein mobiililaitteissa ja pienemmillä näytöillä, koska palvelu on kohdennettu nuorille. Vaikka pääpaino oli mobiilitoiminnassa, niin täytyi käyttöliittymän silti toimia myös suuremmilla näytöillä sillä tasolla, että mikään ei kaatuisi ja näytettävä tieto olisi luettavassa muodossa. Tämä aspekti vaati tarkkaa suunnittelua ja kokeilemista koodin rakenteellisella tasolla frontendissä, jotta sivu osaisi näyttää itsensä oikein riippumatta näytön koosta. Tämän saavuttamisessa hyödynnettiin CSS-tyylittelykieltä sekä erilaisilla ehdollisuuksia, jotka yhdessä määrittelivät näyttökoon reuna-arvot. Nämä reuna-arvot määrittelivät, miten näytölle tulostettava tieto esitetään perustuen näytön kokoon. CSS-tyylittelykieli mahdollisti myös erilaisten väriteemojen, fonttien, marginaalien sekä monien muiden tyyliasetusten lisäämisen osaksi käyttöliittymää, joka ilman näitä tyylittelyjä muuten näyttäisi todella yksinkertaiselta ja pelkistetyltä.

Paralaxin ja tarinakomponentin rakentamisessa hyödynnettiin pohjana Material UI:n tarjoamia valmiita komponentteja, joiden avulla oli mahdollista esittää kuvia sekä videoita käyttöliittymässä. Näiden valmiiden komponenttien hyödyntäminen pilotointia varten oli kustannustehokkain ratkaisu, koska pilotoinnissa täytyisi vain pystyä esittelemään haluttuja ominaisuuksia ja toiminnallisuuksia, jolloin tuotetun ratkaisun ylläpidettävyydellä ei ollut paljon merkitystä. Tämän ratkaisun toteuttaminen kuitenkin vaati JSX-syntaksin ymmärtämistä sekä perustasoa korkeampaa CSS-tyylityskielen osaamista, koska paralaxin toteuttaminen oli monimutkaisempi tuottaa kuin tarinakomponentti. Paralax-komponentti sisälsi huomattavasti enemmän erilaisia ehdollisuuksia, jotka määrittelivät kuinka paralaxin tulee toimia, kun taas tarinakomponentti ei ollut kovin monimutkainen, mutta sisälsi paljon saman koodin toistoa. Pilotoinnin jälkeen, mikäli tarina- ja paralax-komponentit haluttaisiin pysyvään käyttöön, olisi tarpeellista tehdä tutkimusta siitä, kuinka kyseiset ominaisuudet voitaisiin toteuttaa selkeämmin ja tehokkaammin.



KUVA 2. Tarinakomponentti Helpperin käyttöliittymässä

## 5 HELPPERIN PILOTTI & TULOKSET

Helpperin pilotointi suoritettiin toukokuussa 2023 ja pilotointitestaus toteutettiin pääosin ohjatusti, mutta myös osittain vapaamuotoisemmin, mikä tarkoittaa, että testaajat saivat vapaasti selata Helpperiä. Pilotoinnissa esiteltiin Helpperin prototyypiversiota käyttäen mobiililaitteita, kuten puhelimia ja tabletteja ja testausryhmään kuului kouluikäisiä nuoria sekä myös nuoria aikuisia, jotka mahdollisesti voisivat myös olla tulevan palvelun käyttäjiä. Pilotoinnista kerättiin palautetta anonyymisti ja palaute muodostui pääosin siten, että testaajat vastailivat etukäteen määritettyihin kysymyksiin, jotka liittyivät Helpperin käyttäjäkokemukseen, visuaaliseen ilmeeseen, käytettävyyteen sekä esiteltyyn sisältöön. Palautteen yhteydessä kerättiin vain vähän tietoa liittyen pelillistämiseen, koska pilotin pääpaino oli muissa ominaisuuksissa enemmän, mutta tuotetuista staattisista sivuista saatiin kuitenkin jonkin verran palautetta. Palautteessa oleviin kysymyksiin vastattiin numeroarvoilla yhdestä viiteen, mutta lomakkeessa oli myös vapaasanakenttä monipuolisempaa palautetta varten.

Palautelomakkeessa ei ollut suoraa kysymystä parallaxista, mutta tarinakomponentista oli kysymys, jossa käyttäjää pyydettiin arvioimaan yhdestä viiteen, kuinka mielenkiintoiseksi hän koki sisällön esitystavan tällä tavoin. Tämä sai keskiarvokseen arvon 3,6, mikä on hieman keskivertoa parempi. Suurin anti pelillistämiseen liittyen tuli vapaamuotoisesta palautteesta. Testaajat kommentoivat pitävänsä Instagram-tyylisten tarinoiden käyttämisestä nuorille suunnatussa palvelussa, koska se toi sisältöön monipuolisuutta ja tarinallisuus teki sisällöstä mielenkiintoisempaa. Aiheeseen liittyen tuli myös rakentavaa palautetta, mutta se liittyi kokonaan teknisiin haasteisiin, joita käyttäjät olivat kohdanneet selatesaan staattista sivua, jolla tarinapäivitykset olivat. Pelillisyyden kannalta tarinallinen osuus sai siis pelkkää positiivista palautetta, koska rakentava osuus keskittyi teknisiin haasteisiin, mitkä osittain johtuivat siitä, että sivuilla esitettyä tietoa ei haettu dynaamisesti.

Vapaassa palautteessa kommentoitiin myös Helpperin nuorekasta ja visuaalista ilmettä. Osa testaajista koki tämän hyvänä asiana ja antoi vaikutelman, että palvelu on suunnattu nuorille, mikä oli tavoiteltua. Joukossa oli myös muutamia kommentteja, joissa mainittiin, että joillain sivuilla lyhyin väliajoin vaihtuvat väriteemat saattoivat hieman sekoittaa loppukäyttäjää. Tähän tulisi puuttua jatkokehityksessä ja uudelleenarvioida väriteemojen käyttöä. Seuraavaa testausta ajatellen, päätettiin myös pelillistämiseen liittyvien kysymysten määrää kasvattaa ja tutkia asiaa enemmän, jotta kyettäisiin paremmin arvioimaan, saavutetaanko pelillistämällä määritellyt tavoitteet riittävällä tasolla.

## 6 JOHTOPÄÄTÖKSET & POHDINNAT

Helpperin pilotoinnista kerättiin suhteellisen pieni määrä palautetta pelillistämiseen liittyen, mutta perustuen saatuun otantaan, vaikuttivat tulokset olevan oikealla suunnalla. Tuotettu tarinallisuus koettiin monipuoliseksi ja nuorekkaaksi esitystavaksi, joskin teknisesti vielä prototyypissä hieman kankeaksi. Visuaalisuus ja tehokas värien käyttö saivat myös sekä positiivista, että rakentavaa palautetta, mutta positiivista palautetta saatiin enemmän. Tulevaisuuden kannalta pelillistämisen jatkokehittämistä kannattaa varmasti jatkaa ja tutkia mahdollisesti suunnittelussa mainittujen avatarien hyödyntämistä palvelussa.

Ensimmäisen pilotoinnin pohjalta ei voida kuitenkaan vielä täysin varmaksi sanoa, onko pelillistämisen suunnitteluun ja sen käyttöliittymään implementointiin käytetty aika kannattavaa Helpperissä, koska aiheeseen liittyvää palautetta kerättiin todella vähän. Näin pienen otannan perusteella ei voida siis sanoa tullaanko alun tavoitteita täysin saavuttamaan halutussa skaalassa, mutta alustavat viitteet ovat positiiviseen suuntaan.



## LÄHTEET

Antonaci, A., Klemke, R., Kreijns, K. & Specht, M. 2018. Get Gamification of MOOC right!: How to Embed the Individual and Social Aspects of MOOCs in Gamification Design. *International journal of serious games*. 5(3), 61–78. Saatavissa: <https://doi.org/10.17083/ijsg.v5i3.255?nosfx=y>. Viitattu 26.4.2023.

Github, Inc. 2023. Git Guide. Saatavissa: <https://github.com/git-guides>. Viitattu 12.3.2023.

Hamari, J. 2015. Gamification: Motivations & effects. Helsinki: Aalto University. Aalto University publication series DOCTORAL DISSERTATIONS, 11/2015. Väitöskirja. Saatavissa: <http://urn.fi/URN:ISBN:978-952-60-6056-9>. Viitattu 19.3.2023.

Meta Open Source. 2023. React. Saatavissa: <https://react.dev/>. Viitattu 22.3.2023.

Material UI SAS. 2023. Material UI – Overview. Saatavissa: <https://mui.com/material-ui/getting-started/overview/>. Viitattu 11.3.2023.

Morschheuser, B., Hassan, L., Werder, K. & Hamari, J. 2018. How to design gamification? A method for engineering gamified software. *Information and software technology*. Saatavissa: <https://doi.org/10.1016/j.infsof.2017.10.015>. Viitattu 21.3.2023.

Mozilla mdn web docs. 2023a. CSS: Cascading Style Sheets. Saatavissa: <https://developer.mozilla.org/en-US/docs/Web/CSS>. Viitattu 11.3.2023.

Mozilla mdn web docs. 2023b. HTML: Hypertext Markup Language. Saatavissa: <https://developer.mozilla.org/en-US/docs/Web/HTML>. Viitattu 11.3.2023.

Mozilla mdn web docs. 2023c. JavaScript. Saatavissa: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>. Viitattu 11.3.2023.

Mozilla mdn web docs. 2023d. Responsive design. Saatavissa: [https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS\\_layout/Responsive\\_Design](https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Responsive_Design). Viitattu 12.3.2023.

Vercel, Inc. 2023. What is Next.js?. Saatavissa: <https://nextjs.org/learn/foundations/about-nextjs/what-is-nextjs>. Viitattu 11.3.2023.

Xi, N. & Hamari, J. 2019. Does gamification satisfy needs? A study on the relationship between gamification features and intrinsic need satisfaction. *International journal of information management*, 46. Saatavissa: <https://doi.org/10.1016/j.ijinfomgt.2018.12.002>. Viitattu 19.3.2023.