Projektisuunnitelma

Johdanto

Tässä dokumentissa esitellään projektisuunnitelma Ohjelmistotuotanto 1 -kurssin projektin tuottamiseksi. Projektin aiheena ryhmällä on kielenopiskelusovellus, johon viitataan jatkossa nimellä “LanguageApp”. Suunnitelmassa kerrotaan projektin vaatimuksista, tavoitteista, laajuudesta, teknologioista, riskeistä, työskentelytavoista ja muista tärkeistä alueista, joita toteuttamisessa otetaan huomioon.

1. Projektin tavoitteet ja päämäärät

Ryhmä on määritellyt LanguageApp-sovelluksen keskeiseksi tavoitteeksi englannin kielen opettamisen lapsille. Kohderyhmänä voidaan pitää lapsia, jotka osaavat jo lukea ja jotka haluavat aloittaa englannin opiskelun alkeista. Tavoitteena on luoda pelinomainen oppimisympäristö, joka kiehtoo lapsia ulkonäkönsä ja sisältönsä puolesta. Pelistä on ollut tarkoitus luoda niin hyödyllinen, että sitä voitaisiin käyttää myös esimerkiksi “palkintona” englannin kielen oppitunneilla 1. tai 2. luokan opiskelijoilla.

Tavoitteiden tarkemmat määrittelyt löytyvät alta:

1. **Sanaston esittely**

   Helpotetaan oppimista mahdollistamalla käyttäjien opiskella erilaisia englanninkielisiä sanoja. Sanat on kategorisoitu, esimerkiksi “hedelmät”, “eläimet” jne. Sanasto esitellään klikattavien korttien (flashcards) avulla.

2. **Sanaston ymmärtämisen vahvistaminen**

   Vahvistetaan sanaston omaksumista interaktiivisten harjoitusten avulla. Käyttäjät esimerkiksi kuulevat sanan ja valitsevat sanaa vastaavan kuvan, näkevät kuvan ja valitsevat siihen liittyvän tekstin tai kuulevat ja lukevat sanan ja valitsevat sitä vastaavan kuvan.

3. **Sanaston laajentaminen**

   Tarjotaan ensimmäisessä vaiheessa noin 10 erilaista sanaluokkaa, jotta käyttäjät voivat tutkia erilaisia aiheita. Sanaluokkien lisääminen jatkossa on helppoa ja näin voidaan laajentaa sanastoa entisestään.

4. **Englannin opiskelun tukeminen myös kouluissa**

   Suunnitellaan sovellus monipuoliseksi, jotta se voidaan integroida englannin kielen oppitunneille suomalaisissa kouluissa.

5. **Käyttökokemuksen personointi**

   Tarjotaan saumaton käyttökokemus mahdollistamalla käyttäjien rekisteröityminen ja sisäänkirjautuminen sovellukseen.Anetaan käyttäjille mahdollisuus päivittää henkilötietojaan sovelluksessa, mukaan lukien avatarin valinta, joka seuraa heitä harjoitusten aikana.

6. **Oppismisympäristön osallistavuus**

   Kehitä visuaalisesti houkutteleva ja vangitseva peli, joka houkuttelee lapsia eloisilla väreillään ja vuorovaikutteisilla elementeillään.

7. **Laajennetaan sanasto lauseisiin ja keskusteluihin**

   Seuraavilla tasoilla (2 ja 3) sisällytetään lauseiden ja yksinkertaisten keskustelujen oppimista edistämään kielitaitoa. Tämä toteutetaan myöhemmässä vaiheessa, ei OTP1-kurssilla.

Nämä tavoitteet huomioon ottaen kielisovelluksen tavoitteena on tarjota hauska ja arvokas oppimiskokemus suomenkielisille lapsille samalla kun kehitetään heidän englannin kielen taitoaan.

2. Projektin soveltamisala ja laajuus (scope)

Projektin laajuus Ohjelmistotuotanto 1 -kurssilla rajoittuu sanojen opettamiseen käyttäjälle. Lauseiden ja keskustelujen opettaminen ei kuuluu projektin laajuteen ensimmäisessä vaiheessa. Tehtäväpohjat ja ominaisuudet toteutetaan niin, että sovelluksen laajentaminen lauseiden ja keskutelujen opettamiseen voidaan toteuttaa jatkossa helposti ilman suuria koodimuutoksia, lähinnä materiaalia lisäämällä.

Projektin laajuuteen ensimmäisessä vaiheessa kuuluu englannin opettaminen suomenkielisille lapsille. Laajuuteen ei kuulu tässä vaiheessa muiden kielten opiskelu, mutta sovelluksen päivittäminen tukemaan useita eri kieliä on mahdollista myöhemmässä vaiheessa. Tämä vaatii kuitenkin jonkin verran myös koodin päivittämistä.

LanguageApp-sovelluksen laajuuteen ensimmäisessä vaiheessa kuuluu seuraavat asiat:

1. **Mielekkäiden harjoitusten luominen**

   Tarvitaan neljä erilaista pohjaa, millä sanoja opetetaan lapsille, kielinä suomi ja englanti.

2. **Selkeän käyttöliittymän suunnittelu**

   Suunnitellaan käyttöliittymä, joka tarjoaa käyttäjille sulavan kokemuksen harjoittelusta.

3. **Visuaalisesti houkuttelevan peliympäristön luominen**

   Luodaan visuaalisesti houkuttelevia pelejä, jotka houkuttelevat lapsia eloisilla väreillään ja vuorovaikutteisilla elementeillään.

4. **Käyttäjän omien sivujen luominen**

   Käyttäjä voi seurata palkintojaan ja päivittää omia tietojaan omilla sivuilla.

5. **Prisman hyödyntäminen tietokannanhallinnassa**

   Hyödynnetään Prismaa tietokannanhallintajärjestelmänä tehokkaan tiedon tallentamisen ja noutamisen varmistamiseksi.

6. **Admin-paneelin rakentaminen**

   Rakennetaan sovellukseen admin-puoli, josta voidaan hakea tietoa käyttäjistä ja päivittää tehtävien sanastoa.

Lauseiden ja keskustelujen opettelun tukeminen vaiheessa 2

Vaikka tämänhetkiseen tavoitteeseen ei kuulu lauseiden ja keskustelujen opettaminen, nykyiset tehtävärungot tukevat niiden lisäämistä tulevaisuudessa esimerkiksi OTP2-projektin aikana..

3. Vaatimukset

Tässä kappaleessa kerrotaan sekä sovellukselle asetetuista vaatimuksista, että projektin toteuttamistapaan liittyvistä vaatimuksista. Projektin sidosryhmään kuuluu Product Owner, joka antoi sovellukselle vaatimukseksi sen, että siellä tulee olla admin-näkymä ja mahdollisuus kerätä raportteja käyttäjien lukumäärästä ja yleensä käyttäjistä. Lisäksi koko projektin toteuttamistavan vaatimuksena on käyttää tekoälyä sovelluksen tekemisessä. Tämä voi tarkoittaa esimerkiksi ChatGPT:n tai Github Co-pilotin käyttöä koodauksessa. Toinen Product Ownerin antama projektiin liittyvä vaatimus on, että se toteutetaan ketteränä SCRUM-projektina ja jokainen ryhmäläinen vuorollaan on Scrum-master. Kolmantena projektin toteuttamiseen liittyvänä vaatimukseen on Trellon käyttäminen projektinhallinnassa. Tehtävät tulee jakaa neljään eri sprinttiin. Vaatimuksena on pitää Trello ajan tasalla sen suhteen kuka tekee mitä, mitkä asiat on vielä aloittamatta aj mitkä jo tehty loppuun. Jokainen käyttäjä kirjaa omat tuntinsa Trelloon.

Projektin jäsenet ovat keskustelleet muista tavoitteista, joita sovellukselle haluttiin asettaa. Sovellukseen liittyvät vaatimukset käydään läpi alla:

1. **Käyttäjän rekisteröityminen ja kirjautuminen**: Sovelluksen on mahdollistettava käyttäjien rekisteröityminen ja sisäänkirjautuminen. Rekisteröinnin ja kirjautumisen on toimittava loogisella ja helpolla tavalla, eikä ylimääräisiä henkilötietoja käyttäjistä tule kerätä.

2. **Sanaston esittely**: Sovelluksen on tarjottava käyttäjille mahdollisuus opiskella erilaisia englanninkielisiä sanoja erilaisiin kategorioihin liittyen.

3. **Sanaston ymmärtämisen vahvistaminen**: Sovelluksen on sisällettävä interaktiivisia harjoituksia, jotka auttavat käyttäjiä vahvistamaan sanaston omaksumista. Tavoitteena on kehittää ainakin neljä erilaista moduulia, miten sanoja opetetaan lapsille. Sanojen tulee toistua tehtävästä toiseen, eli yhtä sana kategoriaa voidaan opettaa neljällä eri tavalla.

4. **Sanaluokkien laajentaminen**: Sanaston lisäämisen tulee olla helppoa ja tapahtua varsinaisesti koodia muuttamatta, pelkästään aineistoja lisäämällä.

Näin sovellus mahdollistaa itsensä jatkuvan uudistamisen helpolla ja nopealla tavalla.

5. **Lauseiden ja keskustelujen opettelu**: Myöhemmässä vaiheessa on mahdollista lisätä opeteltaviin lauseiden ja keskustelujen opiskelu. Ensimmäisessä vaiheessa tehdyt tehtäväpohjat mahdollistavat sovelluksen laajentamisen keskusteluihin ja lauseisiin helpolla ja yksinkertaisella tavalla.

6. **Englannin kielen oppituntien tuki suomalaisissa kouluissa**: Sovellus on suunniteltava monipuoliseksi, jotta se voidaan integroida englannin kielen oppitunneille suomalaisissa kouluissa.
7. **Henkilökohtaisten tietojen päivittämisen mahdollistaminen**: Sovelluksen on annettava käyttäjille mahdollisuus päivittää henkilötietojaan, mukaan lukien avatarin valinta, joka seuraa heitä harjoitusten aikana.
8. **Visuaalisesti houkutteleva peliympäristö**: Sovelluksen on tarjottava visuaalisesti houkutteleva peliympäristö, joka houkuttelee lapsia eloisilla väreillään ja vuorovaikutteisilla elementeillään.
9. **Prisman hyödyntäminen tietokannanhallinnassa**: Sovellus hyödyntää Prismaa tietokannanhallintajärjestelmänä tehokkaan tiedon tallentamisen ja noutamisen varmistamiseksi.

   Yllä listattujen sovellusvaatimusten lisäksi tavoitteena koko projektin toteuttamisessa on tärkeänä osana se, että jokainen ryhmän jäsen vuorollaan toimii Scrum-masterin roolissa ja ottaa vastuuta projektista. Yhteistyön ja kommunikoinnin tulee oltava tiivistä.

4. Toiminnalliset vaatimukset

Toiminnalliset vaatimukset kielenoppimisen sovellukselle, joka sisältää sovituksen (customization), voivat vaihdella sovelluksen tarkoituksesta ja käyttäjän tarpeista riippuen. Toiminnalliset vaatimukset määrittelevät, mitä sovelluksen on tarkoitus tehdä. Ne ovat avain toimintoja, joita ilman sovellus ei toimisi tai tekisi mitä sen pitäisi tehdä.

Nämä ovat vaatimuksia, jotka on täytettävä, eikä niitä voida ilman.

- **Todennus**: Käyttäjien tulisi pystyä luomaan tilejä ja kirjautumaan sisään turvallisesti. Todennusprosessin tulisi olla luotettava ja suojattu.
- **Liiketoiminnan ydin**: Sovelluksen ydintoiminnallisuus liittyy kielen oppimiseen. Tämä voisi sisältää oppitunnit, harjoitustehtävät, sanaston opettelun ja kieliopin kertaamisen.
- **Kaupat**, **kassat**: Jos sovelluksessa on kauppaominaisuus, käyttäjien tulisi pystyä ostamaan lisäominaisuuksia tai oppimateriaaleja turvallisesti ja helposti.
- **Valtuutukset**: Käyttäjillä voi olla erilaisia rooleja ja oikeuksia sovelluksessa. Esimerkiksi opettajilla voi olla erilaiset oikeudet kuin oppilailla, ja nämä valtuutukset tulisi olla asianmukaisesti hallittuja.
- **Historiatiedot**: Sovelluksen tulisi tallentaa käyttäjien edistymistiedot, kuten suoritetut oppitunnit, testitulokset ja saavutukset. Käyttäjien tulisi pystyä tarkastelemaan omaa edistymistään ja historiaansa.

5. Mitkä ovat ei-toiminnalliset vaatimukset

Ei-toiminnalliset vaatimukset ovat vaatimuksia, jotka määrittelevät "miten" sovelluksen on suoritettava tietty toiminto. Pohjimmiltaan ne ovat sovelluksen laatuattribuutteja, jotka määrittelevät sovelluksen käyttökokemuksen. Ne tunnetaan myös ei-käyttäytymisvaatimuksina, ja ne tulee toteuttaa niiden prioriteetin mukaan sovellustoimintoon nähden. Tämä tekee niistä joustavia jossain määrin, jolloin on mahdollista ohittaa muutama aika, budjetti tai teknologian rajoitusten vuoksi.

Joitakin esimerkkejä ei-toiminnallisista vaatimuksista ovat:

- **Latausnopeus** - meillä on otettu käyttöön nextjs, joka on nopeampi reactjs verrattuna. Kaikki kuvat on sopivassa muodossa ja eivät ole isoja, että käyttäjillä ei olisi pitkä odotusaika, kun se tuli etusivulle vain tutustua meidän sovelluksen.

- Palvelimen **vastauksen toimittamiseen kulunut aika** - valittu tietokanta toimii nopeasti ja ei kuulu paljon aika. Se oli ensimmäinen kohta, miten Prisma oli valittu alusta asti.

- Käyttäjän **vasteaika** mittaa, kuinka nopeasti järjestelmä reagoi käyttäjän toimintaan. Vasteajan ollessa lyhyt käyttäjä kokee järjestelmän nopeaksi ja responsiiviseksi. Onneksi, meillä on nextjs järjestelmä käytössä, se toimii todella nopeasti.
- **Tiedonkulutusrajat** - tämä käsite on erityisen merkityksellinen mobiiliyhteyksissä, kuten älypuhelimissa tai tableteissa, joissa tiedonsiirron määrä voi olla rajoitettu käyttäjän tilaussuunnitelman tai operaattorin määrittämien rajojen vuoksi. Meillä on nyt yksinkertainen sovellus valmiina, missä ei ole paljon käyttäjien data ja pieni määrä niistä ovat pisteitä, muuten se on käyttäjiä omat tiedot.

Ei-toiminnalliset vaatimukset ovat itse asiassa erittäin tärkeitä sovelluksen suorituskyvyn ja käyttökokemuksen kannalta. Jos sovelluksen toiminnallinen vaatimus on hakea luettelo lähialueen sushiravintoloista, mutta sovelluksella kestää yli 30 sekuntia, käyttäjä todennäköisesti hylkää sovelluksen, vaikka toiminnallinen vaatimus lopulta täyttyisi.

6. Resurssien allokointi

Resurssien kohdentaminen (engl. resource allocation) tarkoittaa resurssien, kuten ajan, rahan, henkilöstön ja materiaalien, suuntaamista ja jakamista tietylle tarkoitukselle tai projektille. Tämän toimenpiteen tarkoituksena on optimoida resurssien käyttö ja varmistaa, että ne käytetään tehokkaasti saavuttaaksesi tietyt tavoitteet tai suorittamalla tietyt tehtävät.

Resurssien kohdentaminen voi olla tärkeä osa projektinhallintaa ja liiketoiminnan suunnittelua.

- **Budjetti**:Meidän projektissa ei ollut budjettia, koska se on kouluprojekti.
- **Henkilöstö**: Meillä oli ollut neljä ohjelmistokehittäjiä, jotka osaavat reactjs ja halusivat kehittää itsestä nextjs koodamisella. Kaikki oli koodavat frontend juttuja, ja yksi vahvas tiimijäsen tei suurin osan backendiä.
- **Laitteisto**: Vaikka Next.js on kevyt kehitysympäristö, meillä on Apple ja Windows koneet.
- **Ohjelmistotyökalut**: Next.js-sovelluksen kehittämiseen käytimme WebStorm(IDE), Git.
- **Infrastruktuuri**: tietokantapalvelu, kuten MongoDB Atlas, joka tarjoaa pilvipohjaisen MongoDB-tietokannan.

Kun olimme määrittänyt tarvittavat resurssit, aloitimme projektin suunnittelun ja kehityksen varmistaen, että projektin aikana on kaikki tarvittavat elementit projektin onnistuneeseen toteuttamiseen.

7. Riskienhallinta ja tunnistaminen

- **Teknisiä riskejä projektissa**

Ohjelmistovirheet: Vaikka käytämme tunnettua ja testattua teknologiaa, on aina mahdollista, että ohjelmistossa on virheitä, jotka voivat vaikuttaa sen toimintaan tai tietoturvaan.

Yhteensopivuusongelmat: Projektissa käytettävät teknologiat eivät välttämättä ole yhteensopivia kaikkien laitteiden tai käyttöjärjestelmien kanssa.

Tietoliikenneongelmat: Verkkoyhteyksien katkeaminen tai hidastuminen voi vaikuttaa palvelun saatavuuteen.

Käyttäjätietojen säilyttäminen: Käyttäjätietojen turvallinen säilyttäminen on ensisijaisen tärkeää. Projektissamme otamme tämän huomioon käyttämällä bcrypt-kirjastoa, joka on erikoistunut salasanojen suojaamiseen. Bcrypt on algoritmi, joka hajauttaa käyttäjien syöttämät salasanat sellaiseen muotoon, joka mahdollistaa niiden turvallisen tallentamisen. Tämä tarkoittaa, että henkilökohtaiset tiedot, kuten sähköpostiosoitteet ja salasanat, ovat suojattuja ja turvassa.

Tietojen katoaminen: Tietokantavirheet tai tekniset ongelmat voivat johtaa tietojen menetykseen.

Tietojen väärinkäyttö: On mahdollista, että joku pääsee käsiksi käyttäjätietoihin ja käyttää niitä väärin.

- Aikataululliset riskit

Viivästykset: Tekniset ongelmat, resurssipulat tai muut odottamattomat tapahtumat voivat viivästyttää projektin aikataulua.

Muutokset vaatimuksissa: Jos projektin vaatimukset muuttuvat kesken kaiken, se voi vaikuttaa aikatauluun.

- Resursseihin liittyvät riskit

Henkilöstö: Avainhenkilöiden sairastuminen tai lähtö voi vaikuttaa projektin etenemiseen.

- Muut Riskitekijät

Lainsäädäntö ja määräykset: Lainsäädännön muutokset voivat vaikuttaa projektin toteutukseen tai vaatimuksiin.

Kilpailijat: Kilpailijoiden toiminnat voivat vaikuttaa projektin menestykseen tai markkina-asemaan.

Teknologian muutokset: Uusien teknologioiden ilmestyminen voi tehdä projektissa käytetyn teknologian vanhentuneeksi.

8. Tiimin muodostaminen ja kehitys

Tiimimme koostuu monipuolisista ihmisistä jotka tuovat pöytään erilaisia taitoja ja näkökulmia, jotka yhdessä muodostavat dynaamisen ja tehokkaan yksikön. Säännölliset palaverit, retrospektiivit ja katselmoinnit varmistavat, että kommunikaatio on avointa ja jokainen tiimin jäsen on ajan tasalla projektin etenemisestä.Tiimin kasvun ja kehityksen tukeminen on keskeistä projektin menestyksen kannalta.

9. Teknologiastackin valinnat: Ohjelmointikielet, kehitysympäristöt ja toimintojen järjestys

Kun rakennetaan modernia web-sovellusta, teknologiastackin valinta on kriittinen osa projektin alkuvaihetta. Valitsemalla oikeat työkalut voidaan varmistaa sovelluksen suorituskyky, skaalautuvuus ja ylläpidettävyys.

Tässä projektissa olemme valinneet teknologiastackiksi seuraavat komponentit:

Ohjelmointikieli

JavaScript - JavaScript on dynaaminen, monikäyttöinen ohjelmointikieli, joka on erityisen suosittu web-sovellusten kehittämisessä. Sen avulla voidaan rakentaa sekä palvelin- että asiakaspuolen sovelluksia, mikä tekee siitä hyvän valinnan projektiin.

Palvelinympäristö:

Node.js - Node.js on JavaScriptin suoritusympäristö, joka mahdollistaa JavaScript-koodin suorittamisen palvelimella ja mahdollistaa tehokkaan ja skaalautuvan sovelluksen rakentamisen.

Kehys:

Next.js - Next.js on React-pohjainen kehys, joka mahdollistaa palvelinpuolen renderöinnin ja staattisen sivuston tuotannon. Se tarjoaa kehittäjille työkalut nopeaan ja tehokkaaseen front-end kehitykseen, samalla kun se hyödyntää Reactin komponenttipohjaista arkkitehtuuria.

Tietokanta:

MongoDB - MongoDB on dokumenttipohjainen NoSQL-tietokanta, joka on suunniteltu skaalautuvuus ja joustavuus mielessä. Sen dynaaminen skeema mahdollistaa nopean kehityksen ja monimutkaisten tietorakenteiden tallentamisen.

ORM:

Prisma - Prisma on avoimen lähdekoodin tietokanta-ORM (Object-Relational Mapping), joka tekee tietokannan käsittelystä helppoa ja tehokasta. Se tarjoaa vahvan tyypityksen ja tukee monia tietokantoja, mukaan lukien MongoDB.

Tyylittely:

SASS - SASS (Syntactically Awesome Style Sheets) on CSS:n laajennus, joka mahdollistaa muuttujien, sekoitusten ja muiden edistyneiden ominaisuuksien käytön. Se tekee tyylittelyn hallinnasta ja ylläpidosta helpompaa ja joustavampaa.

Jenkins:

Käytämme Jenkinsiä CI/CD-työkaluna, joka automatisoi koodin integraation, testauksen ja toimituksen. Jenkinsin avulla voimme varmistaa, että koodimuutokset integroidaan säännöllisesti päähaaraan, ja että ne testataan automaattisesti ennen tuotantoon siirtymistä. Tämä vähentää virheiden riskiä ja nopeuttaa kehitysprosessia.

Docker:

Sovelluksen kontitus on nykyään standardi. Dockerin avulla voimme luoda kevyitä ja itsenäisiä kontteja sovelluksellemme, mikä takaa yhdenmukaisen toimintaympäristön kehityksestä tuotantoon. Konttien avulla voimme myös skaalata sovellusta tarpeen mukaan ja varmistaa sen sujuvan toiminnan eri ympäristöissä.

eCloud:

Kaikki infrastruktuurimme pyörii koulun tarjoamalla ecloud-pilvialustalla.

Toimintojen järjestys:

Aloitamme projektin määrittelemällä vaatimukset ja suunnittelemalla tietokannan rakenteen. Tämän jälkeen asennamme tarvittavat työkalut ja kirjastot, kuten Node.js ja Next.js. Kun palvelinympäristö on pystytetty, integroimme MongoDB:n Prisman avulla ja aloitamme back-end kehityksen. Samanaikaisesti aloitamme käyttöliittymän suunnittelun ja kehityksen hyödyntäen Next.js:ää ja SASS:ia. Kun molemmat osat ovat valmiita, yhdistämme ne ja teemme tarvittavat testaukset.

10. Kommunikointi suunnitelma

Kommunikointi suunnitelmamme perustui tehokkaaseen ja monipuoliseen viestintään projektiryhmän jäsenten välillä. Käytimme Discordia päivittäiseen reaaliaikaiseen keskusteluun ja tiedon jakamiseen. Kun aikataulumme ja olosuhteet sallivat, kokoonnuimme myös koulussa välitunneilla ja opetuksen jälkeen.

Discordissa kommunikoimme siitä mitä kukakin tekee ettei työt sotkeentuisi toisiinsa, jos moni työstää samaa asiaa samaan aikaan.

11. Budjetointi ja kulujen arvio

Projektillamme ei ollut erillistä budjettia, mutta voimme hahmotella mahdollisia kustannuksia joita menisi vastaavaan tai tavalliseen kielenoppimis sovellukseen tai -sivuston kehittämiseen. Näitä voisivat olla esimerkiksi:

- Ohjelmointityöhön liittyvät kulut, kuten ohjelmoijien palkat tai ohjelmistolisenssit.
- Graafisen suunnittelun ja käyttöliittymän kehittämisen kustannukset.
- Hosting- ja palvelinkustannukset verkkosivuston ylläpidossa.
- Mahdolliset markkinointi- ja mainoskustannukset, jos sovellusta tai sivustoa on tarkoitus mainostaa.

12. Client involvement. Legal and compliance considerations.

Asiakkaan, joka tässä tapauksessa oli opettajamme, osallistuminen oli keskeinen osa projektiamme. Opettaja toimi tuotteen omistajana ja asetti meille vaatimuksia ja pyyntöjä liittyen sovelluksen tai sivuston sisältöön ja toiminnallisuuteen. Tämä varmisti, että tuote täytti kurssin tarpeet ja tehtävänannon.

Oikeudellisia ja noudattamiseen liittyviä näkökohtia tuli myös ottaa huomioon. Koska tuote oli suunnattu lapsille, meidän oli varmistettava, että se noudattaa tietosuojasäännöksiä ja sisältää asianmukaiset suojaustoimenpiteet. Oikeudellisten ja noudattamiseen liittyvien näkökohtien osalta:

- **Tietosuoja**: Koska sovellus suunnattiin lapsille, meidän tulisi noudattaa tiukasti tietosuoja säännöksiä, erityisesti lasten tietosuojaa koskevia lakeja. Tämä sisältääi asianmukaiset käytänteet henkilötietojen käsittelyssä ja vanhempien suostumuksen hankkimisen.
- **Tekijänoikeudet**: Arvioimme kaikki sovelluksessa tai sivustossa käytetyt kuvat, äänet ja tekstit tekijänoikeuksien varmistamiseksi. Käytimme vain sellaista sisältöä, joka oli lisensoitu tai johon meillä oli lupa käyttää sekä, että ne ovat lapsille sopivia.

Nämä toimenpiteet varmistivat, että projekti ei ainoastaan täyttänyt opetus- ja oppimistavoitteita, vaan säilytti myös laillisuuden ja turvallisuuden näkökulmat huomioon ottaen.

13. Tavoitteiden määrittely. Identifioidaan ohjelmistoprojektin keskeiset tavoitteet.

Projektimme päätavoite oli kehittää kielen oppimissovellus tai -sivusto suomenkielisille lapsille, jotka haluavat oppia englannin kielen alkeita. Päämääränämme oli tehdä kielenoppimisesta pelinomaista, hauskaa ja helppoa. Konkreettiset tavoitteemme olivat:

- Luoda oppimateriaalia, joka on kiinnostavaa ja motivoivaa lapsille.
- Tarjota vuorovaikutteisia tehtäviä ja pelejä, jotka auttavat oppijoita oppimaan englantia leikin varjolla.
- Seurata oppijoiden edistymistä ja tarjota palautetta ja palkintoja heidän saavutuksistaan.
- Tarjota helppokäyttöinen ja intuitiivinen käyttöliittymä, joka soveltuu erityisesti lapsille.
- Toivomme, että tämän projektin tuloksena suomenkieliset lapset voivat innostua ja oppia englannin kieltä iloisesti ja tehokkaasti.