OULUN AMMATTIKORKEAKOULU

Jukka Heikkinen

eJEESUS-RAAMATTUSOVELLUS

eJEESUS-RAAMATTUSOVELLUS

Jukka Heikkinen
Opinnäytetyö
Syksy 2019
Tietotekniikan tutkinto-ohjelma
Oulun ammattikorkeakoulu

TIIVISTELMÄ

Oulun ammattikorkeakoulu
Tietotekniikan tutkinto-ohjelma, ohjelmistokehitys

Tekijä: Jukka Heikkinen
Opinnäytetyön nimi suomeksi: eJeesus-raamattusovellus
Opinnäytetyön nimi englanniksi: eJesus Bible Application
Työn ohjaaja: Kari Laitinen
Työn valmistumislukukausi ja -vuosi: Syksy 2019
Sivumäärä: 26

Opinnäytetyön idea syntyi henkilökohtaisesta tarpeesta kunnolliseen sähköiseen Raamatunlukusovellukseen. Ajatuksena oli helpottaa Raamatun lukemisen aloittamista satunnaisten jakeiden avulla.

Tavoite eJeesus-sovelluksella oli saavuttaa mahdollisimman monia Raamatusta kiinnostuneita lukemaan ja arvioimaan Raamatun sisältöä. Sovellus ei kuitenkaan ole pelkästään uskovaisille tarkoitettu.

Työn tuloksena syntyi Android- ja iOS-mobiilisovelluksista, web-sovelluksesta ja palvelinympäristöstä koostuva järjestelmä. Järjestelmä mahdollisti käyttäjälle satunnaisten Raamatun jakeiden lukemisen suomeksi ja englanniksi.

Kaikkia järjestelmälle suunniteltuja toiminnallisuuksia ei saatu projektissa toteutettua. Teknisesti järjestelmä tukee kaikkia tämänhetkisiä jatkokehitystarpeita.

Asiasanat: ohjelmistokehitys, mobiilisovellukset, Android, iOS, web, Raamattu

ABSTRACT

Oulu University of Applied Sciences
Information and communication technologies, software development

Author: Jukka Heikkinen
Title of thesis: eJesus Bible Application
Supervisor: Kari Laitinen
Term and year when the thesis was submitted: Autumn 2019
Pages: 26

The idea for this thesis came from a personal need for a proper electronic Bible Reading app. The idea was to make it easier to start reading the Bible through random verses.

The goal of the eJesus application was to reach as many Bible readers as possible to read and evaluate Bible content. However, the app is not just for believers.

The work resulted in a system consisting of Android and iOS mobile applications, a web application and a server environment. The system allowed the user to read random Bible verses in Finnish and English.

Not all functionalities designed for the system could be implemented in the project. From a technical point of view, the system supports all current development needs.

Keywords: software development, Android, iOS, web, Bible

ALKULAUSE

Haluan kiittää Oulun ammattikorkeakoulua ymmärryksestä ja joustamisesta tämän opinnäytetyön aikataulun suhteen sekä erityisesti opinnäytetyön ohjaajaa Kari Laitista hyvistä neuvoista ja tuesta kirjallisen osuuden tekemisessä.

Oulussa 19.11.2019

Jukka Heikkinen

SISÄLLYS

SANASTO

AJAX	Asynchronous Javascript and XML, asynkronisia HTTP-pyyntöjä käyttävä tekniikka dynaamisten web-sovellusten kehittämiseen.
CLI	Command Line Interface, tekstimuotoisilla komennoilla toimiva komentotulkki.
CSS	Cascading Style Sheets on tekniikka, joka mahdollistaa tyyliehdotusten yhdistämisen HTML-kielellä kuvatulle rakenteelle.
JavaScript	Dynaaminen komentosarjakieli, jota käytetään pääasiassa web-ympäristössä.
JSON	JavaScript Object Notation on yksinkertainen standardoitu datan esitysmuoto. Ei ole sidoksissa JavaScript-ohjelmointikieleen, mutta JSON-data on helposti muunnettavissa JavaScriptin JSON-objektiksi.
PWA	Progressive Web Application, mobiililaitteessa natiivisovelluksen tavoin toimiva verkkosovellus.
REST	Representational State Transfer on yleinen arkkitehtuurimalli ohjelmointirajapintojen toteuttamiseen, joka pohjautuu HTTP-protokollaan.
TypeScript	JavaScriptiin varaan rakentuva ohjelmointikieli
XML	Extensible Markup Language on rakenteellinen kuvauskieli, jolla kuvataan tiedon rakennetta ilman ennalta määrättyjä koodeja.

XMLHTTPRequest	JavaScript-objekti, joka toimii AJAX-kutsuissa rajapintana palvelimen ja selaimen välillä.

1 JOHDANTO

Tämän opinnäytetyön puitteissa toteutetaan eJeesus-niminen uskonnollinen sovellus ja julkaistaan se Android-käyttöliittymällä varustetuille mobiililaitteille. Sovellus on suunnattu uskonnollisille ihmisille, jotka haluavat lukea Raamatun jakeita mobiililaitteeltaan. Sovellus mahdollistaa satunnaisten jakeiden lukemisen, jakeiden hakemisen ja arvostelun. Käyttäjät voivat katsoa palvelussa suosituimpia jakeita, tallentaa omat suosikkinsa ja jakaa niitä sosiaalisessa mediassa ystävilleen.

Ajatus sovelluksen toteuttamisesta tuli omasta tarpeesta. Puhelimelle on saatavilla joitakin Raamattua käsitteleviä sovelluksia, mutta niiden taso on yleensä omaan kokemukseen perustuen lähinnä keskinkertainen, eikä montaakaan ole tarjolla suomen kielellä. Satunnaisten jakeiden lukeminen ei myöskään ole muissa sovelluksissa ollut mahdollista.

Sovellus tehdään aluksi suomeksi, mutta mikäli käyttäjät kokevat sovelluksen hyväksi, voidaan siitä kohtuullisella vaivalla toteuttaa myös erikielisiä versioita. Ei ole myöskään mahdoton ajatus toteuttaa samalla periaatteella muihin uskontokuntiin vetoavia sovelluksia, onhan myös muilla omat pyhät kirjansa, joista he mielellään lukevat otteita.

Jokainen käyttäjä voi tallentaa omat suosikkijakeensa, mutta sovellus ei kuitenkaan tallenna käyttäjästä yksilöityä tietoa, jolloin tietosuojaan liittyvästä lainsäädännöstä ei tarvitse huolehtia. Teknisesti itse sovellus ja sen käyttöliittymä toteutetaan NativeScript-kehitysympäristöä, MySQL-tietokantoja ja PHP-ohjelmia hyödyntäen. Sisältö säilytetään ensisijaisesti käyttäjän laitteessa, jolloin hakuja voi tehdä myös silloin, kun verkkoa ei ole saatavilla – esimerkiksi lentokoneessa. Sovellus kuitenkin tallettaa tietokantaan ihmisten antamat arviot eri jakeille ja päivittää ajantasaisen tiedon käyttäjän päätelaitteeseen.

Raamatun koko sisältö on maksutta saatavilla digitaalisessa muodossa lähes kaikilla maailman eri kielillä, mikä mahdollistaa keskittymisen itse sovelluskehitykseen. Opinnäytetyössä käytetään Suomen evankelisluterilaisen

kirkon vapaasti saataville asettamaa vuoden 1992 uutta Raamatun käännöstä (1).

2 EJEESUS-SOVELLUS

2.1 Yleistä

Niin sanottujen pyhien kirjoitusten lukeminen on uskonnollisesti suuntautuneiden ihmisten keskuudessa suosittu harrastus. Esimerkiksi Barna Groupin ja Yhdysvaltojen Raamattusäätiön tekemän tutkimuksen (2) mukaan noin puolet aikuisista Yhdysvaltain asukkaista selailee joskus Raamattua ja 35 prosenttia aikuisväestöstä tutkii Raamattua vähintään kerran viikossa.

Kirjoituksiin tutustuminen luonnollisesti vaihtelee uskonnosta ja kulttuurista riippuen, mutta varovaisestikin arvioiden sadat miljoonat ihmiset päivittäin tutustuvat uskontonsa pyhiin kirjoituksiin. Aiemmin mainitusta tutkimuksesta (2) käy myös ilmi, että yli 40 prosenttia Raamatun ”käyttäjistä” tutustuu Raamatun sisältöön jonkin mobiilisovelluksen avulla.

Luku on yllättävän suuri siihen nähden, että jokapäiväiseen käyttöön soveltuvien sovellusten määrä on suhteellisen rajattu. Internethaku Googlen sovelluskauppaan (20.3.2019) paljastaa, että Bible-hakusanalla löytyy n. 100 eri sovellusta, joista iso osa on erikielisiä Raamatun painoksia. Joukkoon mahtuu myös pelejä ja muita aihetta vain ohuesti sivuavia sovelluksia.

Opinnäytetyössä tehtävä eJeesus-sovellus mahdollistaa satunnaisten Raamatun jakeiden lukemisen, jakeiden hakemisen ja arvostelun. Käyttäjät voivat selata palvelusta suosituimpia jakeita, tallentaa lempijakeensa myöhempää käyttöä varten sekä jakaa Raamatun jakeita sosiaalisessa mediassa ystävilleen. Kaikki ominaisuudet eivät tule ohjelman ensimmäiseen versioon mukaan vaan myöhempien ohjelmapäivitysten mukana tai mahdollisesti maksullisina lisäominaisuuksina.

Samasta sovelluksesta on myöhemmin tarkoitus tehdä eri uskontokunnille soveltuvia versioita, joiden avulla käyttäjät voivat helposti ja nopeasti tutustua esimerkiksi Koraaniin tai vastaaviin uskonnollisiin kirjoituksiin.

2.2 Toiminnalliset vaatimukset

On yleistä tietoa, että Raamattu on perinteisesti jaettu kirjoihin, jotka on jaettu edelleen lukuihin ja luvut jakeisiin. Sovelluksen keskeinen toiminta on Raamatun satunnainen lukeminen, minkä vuoksi jokainen jae ja luku täytyy olla eriteltynä tietokannassa.

Käyttäjän tulee kyetä selaamaan satunnaisia jakeita ja jatkaa Raamatun lukua haluamastaan kohdasta. Satunnaisen selaamisen lisäksi käyttäjät voivat hakea haluamiaan kirjoja, lukuja tai jakeita hakutoiminnon avulla. Käyttäjä voi jatkaa lukemista haluamastaan kohdasta ja lukea vaikka koko Raamatun löydettyään ensin häntä kiinnostavan kohdan.

On tärkeää, että käyttäjä voi jatkaa lukemista siitä kohdasta, mihin hän on aiemmalla kerralla jäänyt. Tämän lisäksi käyttäjän tulee kyetä tallentamaan sekä arvostelemaan mieleisensä jakeet tai luvut. Käyttäjien arvostelut kerätään tietokantaan, minkä jälkeen käyttäjät voivat halutessaan keskittyä selaamaan muiden käyttäjien keskimäärin parhaina pitämiä kohtia.

Käyttäjillä ei ole ainakaan alkuvaiheessa kuitenkaan henkilökohtaisia tilejä tai muuta yksilöintiä. Tämä mahdollistaisi tietojen tallennuksen palvelimelle, jolloin käyttäjä voisi saada tallentamansa tiedot esimerkiksi vaihtaessaan puhelinta. Tämä ei kuitenkaan ole tarkoituksenmukaista opinnäytetyön mittakaavassa.

2.3 Tekniset vaatimukset

Tavoittaakseen mahdollisimman suuren yleisön tulee sovelluksen olla käytettävissä sekä Android- että iOS-käyttöjärjestelmillä varustetuilla laitteilla. Tämän vaatimuksen vuoksi sovellus tulee toteuttaa sellaisella arkkitehtuurilla, joka mahdollistaa yhteisen koodipohjan molemmille sovelluksille. Tämän mahdollistaa NativeScript, joka on JavaScript -pohjainen sovelluskehys. Lisää tietoa käytetyistä tekniikoista on luvussa 3: ”Käytetyt tekniikat ja työkalut”.

Jokaisella käyttäjällä tulee olla mahdollisuus käyttää sovellusta myös offline-tilassa ilman verkkoyhteyttä. Tämä edellyttää sisältöjen tallentamista puhelimeen, sovelluksen sisäiseen tietokantaan. Sovelluksen sisäisen

tietokannan ohella tulee olla yksi yhteinen palvelimella sijaitseva tietokanta, johon kerätään käyttäjien arvosteluja ja muuta tilastollista tietoa. Tekstipohjaisena Raamattu ei vie merkittävästi tallennustilaa puhelimelta, eikä päätelaitteen kapasiteetin pitäisi olla rajoite tässä suhteessa. Tässä lueteltuja kaikkia teknisiä vaatimuksia ei tulla toteuttamaan ohjelman ensimmäisessä julkaisuversiossa.

Tietoliikenne sovelluksen ja palvelimella olevan tietokannan välillä toteutetaan ensisijaisesti REST-rajapintaa käyttäen (REST API). REST on lyhenne sanoista ”Representational State Transfer”. Kyseessä on tietoverkkojen arkkitehtuurimalli. REST-rajapinta (API) taas on rajapinta, jonka avulla sovellukset voivat keskustella palvelimen kanssa (3, s. 5—6).

3 KÄYTETYT TEKNIIKAT JA TYÖKALUT

3.1 Tekniikat

3.1.1 Tiedonharavointi

Tiedonharavointi tarkoittaa automatisoitua tietojen keräämistä internet-sivun lähdekoodista. Ohjelmoitu sovellus hakee halutun sisällön, analysoi ja erittelee siitä halutut tiedot. Haravoitu tieto voidaan lisätä esimerkiksi tietokantaan.

Tämän opinnäytetyön tiedonharavoinnissa käytettiin PHP-ohjelmaa ja tietojen tallentamiseen MySQL-tietokantaa.

3.1.2 NativeScript

NativeScript on ohjelmistokehys, jonka avulla voidaan luoda usealla eri alustalle toimivia sovelluksia käyttäen yhteistä lähdekoodia. Nativescriptillä voi luoda iOS- ja Android-sovellusten lisäksi web-sovelluksia. NativeScript-sovelluksia voi ohjelmoida käyttäen TypeScriptiä, JavaScriptiä, Vue.js:ää ja Angularia. NativeScript kääntää ohjelmoidun koodin natiivikoodiksi toisin kuin hybridiohjelmistokehykset, jotka näyttävät vain web-ohjelman kehyksessä. NativeScriptillä kehittäminen on suhteellisen helppoa, koska se muistuttaa läheisesti web-ohjelmointia. Sen vuoksi sillä on mahdollista luoda ohjelmistoja nopeasti ja tehokkaasti nopealla aikataululla.

3.1.3 Ubuntu Linux

Palvelimen käyttöjärjestelmäksi valittiin Ubuntu Linuxin versio 15.04. Ubuntu on eräs Linuxin jakeluversioista ja on Debianin ohella yksi suosituimpia alustoja serverikäyttöön. Vaikka versio on vanha ja sille ei enää löydy tukea, toimii se serverin ylläpitoon tässä vaiheessa riittävän hyvin.

3.1.4 Apache

Apache on internetin suosituin (4) HTTP-palvelinohjelma, joka perustuu avoimeen lähdekoodiin. Apache tukee suoraan vain staattisten tiedostojen jakoa

HTTP-protokollan yli. PHP ja SSL esimerkiksi täytyy ottaa erikseen käyttöön moduulina. Tässä palvelinkonfiguraatiossa on käytössä Apachen versio 2.4.10.

3.1.5 PHP-ohjelmointikieli

PHP (lyhenne sanoista PHP: Hypertext Preprocessor) on web-palvelinympäristöissä laajalti käytetty ohjelmointikieli, jolla voi luoda dynaamisia web-sivustoja ja komentoriviohjelmia. PHP-koodia ei käännetä suoritettavaksi ohjelmaksi vaan se tulkitaan suoritusvaiheessa.

3.1.6 cURL

cURL on komentoriviohjelma tiedonsiirtoon. Se tukee seuraavia protokollia: FTP, FTPS, HTTP, HTTPS, SCP, SFTP, TFTP, TELNET, DICT, LDAP, LDAPS ja FILE. Tämän opinnäytetyön kannalta tärkeimpinä HTTP ja HTTPS.

3.1.7 AJAX

AJAX eli Advanced Javascript and XML on tekniikka, jolla voidaan tehdä palvelimelle asynkronisia HTTP-pyyntöjä ja palauttaa dataa esimerkiksi XML-dokumenttina tai nykyään yleisemmin yksinkertaisemmassa JSON-tiedostomuodossa. Puhtaalla JavaScriptillä AJAX-kutsujen tekeminen on hieman työlästä käyttäen XMLHTTPRequestia, mutta tässä sovelluksessa käytetty Angular-sovelluskehys sisältää siihen sopivat rajapinnat ja helpottaa huomattavasti. Tämän sovelluksen käyttämät AJAX-kutsut palauttavat PHP-rajapinnalta kaiken datan JSON-muodossa.

3.1.8 REST-rajapinta

REST eli Representational State Transfer on arkkitehtuurimalli, jota käytetään yleisesti HTTP:n kautta käytettävien rajapintojen toteuttamiseen. Se määrittää operaatiot, joilla palvelimilla olevaa dataa käsitellään. Data saa olla missä muodossa tahansa eikä sille ole mitään määriteltyä standardia mutta rajapinnan täytyy täyttää tietyt ehdot ollakseen RESTful. Yleisimmin datan formaatti on XML tai JSON. REST-rajapinta ei säilytä tietoa tilasta, vaan jokainen pyyntö on toisistaan riippumaton ja tieto ohjelman tilasta on rajapintaa käyttävän sovelluksen vastuulla. REST pohjautuu HTTP-protokollan ominaisuuksiin ja

käyttää sen metodeja GET, POST, PUT ja DELETE tietojen käsittelyyn. Pyyntöjen data ei sisällä metatietoja ja kaikki pyynnöt menevät metodien luonnetta kuvaaviin osoitteisiin, joten data on erittäin helppolukuista. REST ei ole kovinkaan tiukka standardi ja se antaa paljon mahdollisuuksia suunnitteluun. Tiukka arkkitehtuurin noudattaminen ei takaa hyvää rajapintaa.

3.1.9 MySQL-tietokanta

MySQL on maailman suosituin relaatiotietokantaohjelmisto (5), jota käytetään useasti web-pohjaisissa sovelluksissa. MySQL:ssä tieto tallennetaan tauluihin ja tauluissa jaoteltuihin sarakkeisiin. Tietoa voidaan yksilöidä käyttäen avaimia ja linkittää eri taulujen tietoja viiteavaimilla toisiinsa. MySQL on rakenteellisesti tässä sovelluksessa järkevin vaihtoehto tietokannaksi, koska tietorakenne on aina samanlainen eikä ole tarvetta dynaamiselle datalle.

MySQL:n käyttö on ilmaista ja sen lähdekoodi avointa. MySQL:n käyttöönotto on vaivatonta, koska suurin osa web-hotelleista ja palvelintilaa tarjoavista palveluntarjoajista tarjoaa sitä tukevaa palvelinympäristöä tai valmista tietokantaa halpaan hintaan ja työkalut tietokannan hallintaan ovat valmiina tai helposti asennettavissa.

3.2 Työkalut

3.2.1 NativeScript Playground

NativeScript Playground on helppokäyttöinen selainpohjainen editori, jolla on kätevää nopeasti testata NativeScriptillä luotuja ohjelmia ja korjata nopeasti mahdollisia virheitä. Se mahdollistaa sovelluksen esikatselun laitteissa, joihin on asennettu NativeScript Playground -sovellus käyttäen QR-koodia. Kaikki muokattu ja tallennettu koodi päivittyy esikatseluun laitteessa lähes reaaliaikaisesti. Esimerkkiohjelmia lataamalla ja muokkaamalla voi oppia helposti tekemään omia ohjelmia. Tämä ympäristö on kuitenkin osittain rajoittunut eikä ihan kaikki mahdollinen natiivikoodi esimerkiksi toimi esikatseluohjelman kautta. Web-selaimessa ohjelmointi ei myöskään ole kovin mielekästä ja siksi tämän työkalun käyttö on lähinnä prototyyppien ja nopeiden demoversioden kehitystä varten järkevää.

3.2.2 Atom-tekstieditori

Tärkein työkalu ohjelmoinnissa on pätevä tekstieditori. Atom on hyväksi havaittu, helppokäyttöinen ja yksinkertaisesti kaunis alustariippumaton ohjelma tehokäyttäjille, ja sen lisäosia ja ulkoasua on helppo muokata. Tässä projektissa Atom on kaikista tehokkain työkalu senkin vuoksi, ettei mitään koodia tarvitse alkaa itse erikseen kääntämään. Atom on ladattavissa ilmaisena osoitteesta atom.io.

3.2.3 NativeScript CLI

NativeScriptin oma komentoriviohjelma mahdollistaa ohjelman koodaamisen ja ajamisen paikallisesti ilman internetyhteyttä. NativeScript CLI tukee kaikkia lisäosia, jotka eivät oletuksena ole tuettuina NativeScript Playgroundissa.

3.2.4 Certbot

TLS-suojausta varten sertifikaatti (kuva 1) on hankittu Let’s Encrypt -nimiseltä palvelulta, joka on ilmainen ja sen asennus on hoidettu Certbot-ohjelmalla. Certbotia käytetään komentoriviltä ja sertifikaatti täytyy uusia 60 vuorokauden välein. Suojatun yhteyden käyttö kryptaa kaiken dataliikenteen käyttäjän ja palvelun välillä ja tekee palveluista luotettavan ja ammattimaisen. Suojaamattomana palvelun käyttö olisi mobiililaitteissa käytännössä mahdotonta Google Playn ja varsinkin App Storen tarkkojen turvallisuusvaatimusten vuoksi.

KUVA 1. Certbotin asentama sertifikaatti ejees.us-sivustolla

3.2.5 FTP- ja SSH-asiakasohjelmat

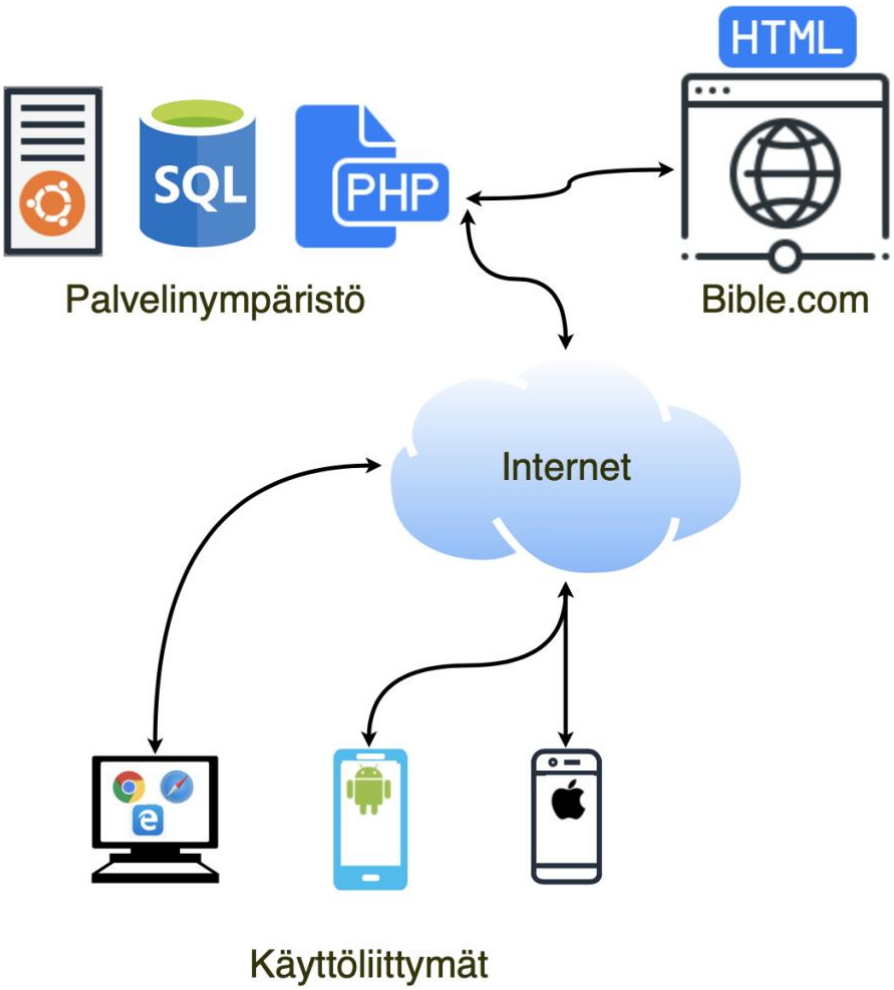
Tiedostojen siirto on hoidettu FileZilla-FTP-ohjelmalla ja palvelimen etäkäyttö komentoriviltä SSH-pääteohjelmalla.

3.2.6 phpMyAdmin

PhpMyAdmin on selaimen kautta käytettävä MySQL-tietokannan hallintatyökalu. SQL-kyselyiden suorituksen lisäksi se sisältää moneen toimintoon valmiin käyttöliittymän. Useimmissa webhotelleissa phpMyAdmin on valmiiksi asennettuna.

4 JÄRJESTELMÄN TOTEUTUS

Seuraavissa kohdissa kutakin järjestelmän osaa tarkastellaan erikseen.



KUVA 2. Järjestelmän rakenne

4.1 Palvelinympäristö

Palvelimelle on asennettu käyttöjärjestelmäksi Ubuntu Linux 15.04.

4.1.1 Tietokanta

Ennen ohjelmiston prototyypin kehittämistä oli pakollista saada tietokanta Raamatun jakeista. Sitä varten täytyi suunnitella PHP-skripti, joka osaa kerätä ja parsia bible.com:sta jakeet käyttäen cURL-komentoriviohjelmaa sekä lisätä ne

MySQL-tietokantaan. Tietokannassa data ryhmitellään kirjoittain, luvuittain ja jakeittain. Tällä hetkellä tietokanta sisältää Raamatun kokonaisuudessaan kielillä suomi ja englanti mutta uusien kielten lisäämien tulevaisuudessa on helppoa, koska tarvitsee vain lisätä yksi kenttä tietokantatauluun ja ajaa PHP-ohjelma uudestaan. Haku kestää muutaman minuutin. Kaikkia kieliä ei kannata ladata mahdollisten tekijänoikeusrikkomusten vuoksi.

Tietokannan täytyi olla rakenteeltaan relaatiotyyppinen, joten MySQL-tietokanta oli järkevin vaihtoehto. Tietokannan muokkaamiseen ja hallintaan käytettiin web-pohjaista phpMyAdmin-työkalua (KUVA 2).

book	chapter	verse	FI	EN
GEN	1	1	Alussa loi Jumala taivaan ja maan.	In the
GEN	1	2	Ja maa oli autio ja tyhjä, ja pimeys oli syvyyden ...	The e
GEN	1	3	Ja Jumala sanoi: "Tulkoon valkeus". Ja valkeus tul...	God s
GEN	1	4	Ja Jumala näki, että valkeus oli hyvä; ja Jumala e...	God s
GEN	1	5	Ja Jumala kutsui valkeuden päiväksi, ja pimeyden h...	God c
GEN	1	6	Ja Jumala sanoi: "Tulkoon taivaanvahvuus vetten vä...	God s
GEN	1	7	Ja Jumala teki taivaanvahvuuden ja erotti vedet, j...	God m
GEN	1	8	Ja Jumala kutsui vahvuuden taivaaksi. Ja tuli ehto...	God c
GEN	1	9	Ja Jumala sanoi: "Kokoontukoot vedet, jotka ovat t...	God s
GEN	1	10	Ja Jumala kutsui kuivan maaksi, ja paikan, mihin v...	God c
GEN	1	11	Ja Jumala sanoi: "Kasvakoon maa vihantaa, ruohoja,...	God s
GEN	1	12	maa tuotti vihantaa, ruohoja, jotka tekivät siemen...	The e
GEN	1	13	Ja tuli ehtoo, ja tuli aamu, kolmas päivä	There

KUVA 3. Tietokannan sisältöä phpMyAdminissa

4.1.2 REST-rajapinta

REST-rajapinta ohjelmoitiin PHP-ohjelmointikieltä käyttäen. Muutamia parametreja käyttäen se osaa hakea MySQL-tietokannasta dataa ja palauttaa sen tekstimuodossa JSON-formaatissa.

Tässä sovelluksessa käytettävä rajapinta ei ole täysin RESTful vielä julkaisuvaiheessa, mutta myöhemmin käytettäviä ominaisuuksia varten se rakentuu tukemaan sitä täysin. Alkuvaiheessa käytetään pelkästään GET-

metodia tiedon hakemiseen ja myöhemmin otetaan käyttöön POST, PUT ja DELETE, kun halutaan lisätä, muokata ja poistaa käyttäjien luomaa dataa. REST-rajapinnan käyttö on tässä sovelluksessa järkevää ja helppoa datan yksinkertaisen rakenteen vuoksi ja myös siksi, ettei ohjelmiston tarvitse mitenkään merkittävästi skaalautua tulevaisuudessa ainakaan tietokantarakenteiden osalta. Kaikki vastaanotettu tai lähetetty data rajapinnan kautta käyttää JSON-formaattia mutta on myös helposti muutettavissa tukemaan esimerkiksi XML-muotoa. REST-rajapintaa voisi myöhemmin käyttää tarvittaessa integraatioihin.

4.2 Mobiilisovellukset

Mobiilisovellus Androidille ja iOS:lle on tehty NativeScriptillä. Tässä projektissa syy NativeScriptin käyttöön oli matala oppimiskynnys, valmiiksi tutut ohjelmointikielet ja monen alustan tuki. Myös suorituskyky, ylläpidettävyys ja laajennettavuus ovat NativeScriptin suurimpia hyviä puolia. NativeScript-sovelluksissa tuetut ohjelmistokehykset Vue.js ja Angular molemmat tukevat suunnilleen samoja lisäosia, mutta Vue.js:n kanssa havaittujen ongelmien vuoksi tässä päädyttiin käyttämään Angularia. NativeScript Playground ja CLI olivat käytössä työkaluina, ja koodin kirjoituksessa käytettiin Atom-tekstieditoria. NativeScriptin käytössä oli tärkeintä ymmärtää web-tekniikoita, koska lähes kaikki koodi tehtiin käyttäen JavaScriptiä ja web-ohjelmoinnista tuttua Angular-ohjelmistokehystä.

4.3 Web-sovellus

Web-sovelluksen täytyi sisältää alkuvaiheessa prototyypille vaadittavat ominaisuudet. Periaatteessa tarvittiin vain painike, jota painamalla saa satunnaisen jakeen Raamatusta esille. Käyttämällä HTML:ää, CSS:ää ja JavaScriptiä saatiin helposti aikaan sivusto, joka haki JavaScriptin AJAX-tekniikalla REST-rajapinnalta tarvittavan datan JSON-muodossa ja pilkkoi sen sivustolle näkyviin.

4.4 Käyttöliittymät

4.4.1 Web-sovellus

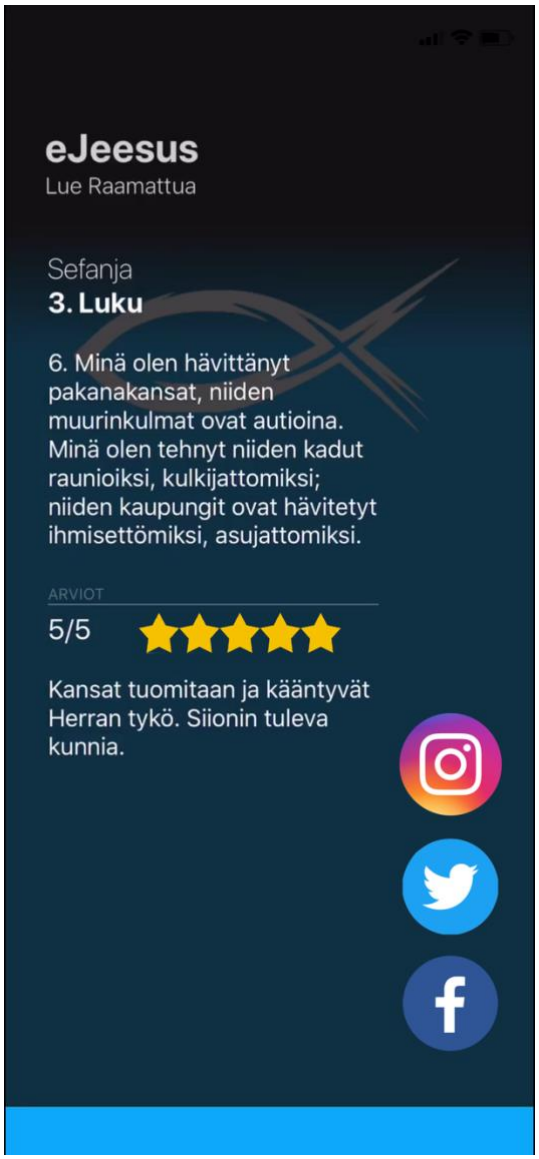
Yksinkertainen HTML-käyttöliittymä (kuva 4) tehtiin käyttäen HTML/CSS-tekniikoita. Web-sovelluksen käyttöliittymän täytyi olla toimiva sekä tietokoneen selaimella että mobiilissa ja CSS-määrittelyillä molemmat saatiin toimiviksi.

KUVA 4. Web-käyttöliittymä mobiililaitteissa

4.4.2 Mobiilisovellukset

Käyttöliittymässä pyrittiin yksinkertaiseen ja helppoon käyttökokemukseen ja intuitiiviseen oppimiseen. Ohjelmassa on aina kerrallaan yksi Raamatun jae esillä ja uusia voi hakea pyyhkäisemällä vasemmalle. Jakeet jäävät puhelimen muistiin sovelluksen ollessa käynnissä ja niitä voi selata edestakaisin pyyhkäisemällä. Jakeen voi klikkaamalla avata koko ruudulle ja saada lisäksi luvun kuvauksen ja arviointinapit näkyviin. Jakeista voi tykätä ja niitä voi jakaa sosiaaliseen mediaan pikapainikkeilla (kuva 5). Myöhemmin julkaistavissa päivityksissä on tarkoitus lisätä mahdollisuus jakeiden tallentamiseen puhelimeen ja hakutoiminto hakusanoin ja suosituimpien jakeiden katselu. Raamatun suoraa lukemista ensimmäiseen ohjelmaversioon ei tule, vaan linkin kautta pääsee lukemaan Raamattua kyseisestä kohdasta internetistä.

Käyttöliittymän ulkoasu tehtiin käyttäen NativeScriptin omia valmiita elementtejä ja CSS-tyylimuotoiluja. Käyttöliittymän kosketussyötteet käyttävät NativeScriptin Gestures-moduulia.



KUVA 5. Mobiilikäyttöliittymä

4.5 Testaus

Ohjelmiston toimivuutta testattiin web-pohjaisesti pelkillä selainten omilla JavaScript-konsoleilla ja NativeScript Previewin virheenkorjaustyökalulla mobiililaitteissa.

Sovellusta testattiin käyttäjillä web-pohjaisena jo yli vuosi todeten se erittäin toimivaksi. Natiivit mobiilisovellukset tällä hetkellä testattu lähinnä itse. Testausta ennen julkaisua tehdään beta-ohjelmien kautta. Testauksessa on lähinnä saatu varmaa tietoa siitä, että itse idea toimii hyvin, ja seuraavaksi täytyy saada varmuus mobiiliversion toiminnasta.

5 JULKAISU MOBIILIALUSTOILLE

Sovellus on tarkoitus julkaista 2020 keväällä Android-alustalle ja syksyllä iOS:lle. Maksullisena ohjelmaa ei ainakaan aluksi julkaista, enintään myöhemmin tulevat lisäominaisuudet voivat tulla maksullisena. Mainoksilla ei ole tarkoitus rahastaa vaan kerätä mieluummin lahjoituksia.

5.1 Google Play

Sovellus julkaistaan ensin Google Playhin, koska se on halvempaa. Ensin julkaistaan kokeiluversio eli betasovellus rajoitetulla käyttäjämäärällä, jotta sovellus saadaan testattua kunnolla ennen virallista julkaisua. Mahdollisen palautteen kautta ja raporttien avulla sovellusta korjaillaan julkaisukelpoiseksi ja lopulta versio, joka toimii tarpeeksi luotettavasti, laitetaan kauppaan.

Google ei erityisen hyvin tarkista sovellusten laatua ennen julkaisua, joten betatestaus on tärkeää. Julkaisuun tarvitaan rekisteröity maksullinen kehittäjätili. Rajoituksena Google Playssä on ohjelman koko (100 megatavua) ja ohjelman tietosivu täytyy täyttää oikein. Tietosuojakäytäntö, luokittelu, graafiset sisällöt ja ikärajoitukset täytyy määritellä. Kuitenkin ohjelman päivitysversioiden julkaiseminen on myöhemmin melko vaivatonta ja nopeaa.

5.2 App Store

Koska iOS-sovellusten julkaisu on kalliimpaa ja App Storeen julkaiseminen vaatisi enemmän testaamista, julkaisua täytyy lykätä siihen asti, että Android-versio tuottaa jotain tai kysyntää on tarpeeksi iOS-versiolle. App Storeen julkaisua varten täytyy olla tarkkana, ettei sovellusta hylätä. Applen yleisten ohjeiden mukaisesti tehdyt sovellukset, jotka eivät kaatuile, menevät parhaiten läpi. Täytyy myös hankkia maksullinen kehittäjätili saadakseen julkaista mitään. Kauppaan lataamisen jälkeen odotetetaan hyväksyntää ja voidaan aloittaa TestFlight-betatestaus. Mahdollisten korjausten jälkeen sovellus julkaistaan myyntiin.

6 YHTEENVETO

Työn päätavoitteena oli luoda sovellus Raamatun satunnaisten jakeiden lukemiseen. Sitä varten vaatimuksena oli luoda yksinkertaisin toimiva järjestelmä, joka sisälsi tietokannan Raamatun jakeista, rajapintaohjelmiston ja graafisen käyttöliittymän.

Opinnäytetyön tuloksena syntyi järjestelmä, joka sisältää palvelinohjelmiston, Android- ja iOS-sovellukset sekä web-sovelluksen. Ohjelmiston tarvitsema tietokanta luotiin PHP-ohjelmalla hakemalla Raamatun sisältö bible.comista. REST-rajapinta luotiin myös PHP:lla ja käyttäjäohjelmistot web-käyttöliittymässä ja mobiilikäyttöliittymissä käyttäen JavaScriptiä, HTML:ää, CSS:ää.

Opinnäytetyössä asetetut tavoitteet saavutettiin ja prototyyppiasteelle asti suunnitellut ominaisuudet saatiin aikataulun mukaisesti toimimaan järjestelmän jokaisessa osassa.

Suuria ongelmia ei suoranaisesti missään vaihteessa ollut, mutta mobiilikäyttöliittymien ja tekniikoiden yhteensopimattomuuksien kanssa oli vaikeuksia jonkin verran. Kaikki käyttöliittymän elementit ja laitetason toiminnallisuudet eivät olleet suoraan yhteentoimivia Androidissa ja iOS:ssä. Nämä ominaisuudet vaativat hieman hiomista myöhempiin versioihin. Tietoturvallisuutta REST-rajapintaan täytyisi lisätä esimerkiksi Oauth2-suojauksella.

Testaaminen jäi teknisten ongelmien takia hieman vähäiseksi johtuen resurssipulasta ja ajankäytön rajallisuudesta. Ainoastaan web-sovelluksen toimintaa pystyttiin testaamaan tehokkaasti.

Lopputulos on hyvä ja tulevaisuuden parannukset tekevät ohjelmistosta paljon paremman. Julkaiseminen on todennäköistä 2020 keväällä ja riippuen ohjelman suosiosta ja käytöstä sitä kehitetään vielä lisää.

LÄHTEET

1. Raamattu 1992. Suomen evankelilais-luterilaisen kirkon kirkolliskokouksen vuonna 1992 käyttöön ottama suomennos. Saatavissa: https://raamattu.fi/kaannokset/KR92. Hakupäivä 25.11.2019.

2. Barna Group 10.07.2018. State of the Bible 2018: Seven Top Findings - Barna Group. Saatavissa: https://www.barna.com/research/state-of-the-bible-2018-seven-top-findings. Hakupäivä 20.3. 2019.

3. Mass , Mark 2012. REST API design rulebook. Sebastopol, CA: O'Reilly.

4. W3Techs 2019. Usage of web servers. Saatavissa: https://wtechs.com/technologies/overview/web_server. Hakupäivä 19.11. 2019.

5. ScaleGrid 2019. 2019 Database Trends – SQL vs. NoSQL, Top Databases, Single vs. Multiple Database Use. Saatavissa: https://scalegrid.io/blog/2019-database-trends-sql-vs-nosql-top-databases-single-vs-multiple-database-use. Hakupäivä 19.11.2019.