

Mobiilianalytiikan oppimateriaalin kehitys tietojenkäsittelyn koulutukselle



Ammattikorkeakoulututkinnon opinnäytetyö

Visamäki, Tietojenkäsittely tradenomi

Syksy, 2014

Lasse Pulkki

Tietojenkäsittelyn Koulutusohjelma
Visamäki

Tekijä	Lasse Pulkki	Vuosi 2017
Työn nimi	Mobiilianalytiikan oppimateriaalin kehitys tietojenkäsittelyn koulutukselle.	
Työn ohjaaja/t	Tommi Saksa	

TIIVISTELMÄ

Tämän opinnäytetyön toimeksiantajana oli Hämeen ammattikorkeakoulun tietojenkäsittelyn koulutusohjelma. Koulutusohjelma sisältää Mobile Analytics kurssin, jota varten opinnäytetyö tehdään. Tämä kurssi on osa Mobile Programming moduulin kokonaisuutta ja on tarkoitettu tietojenkäsittelyn kolmannen vuosikurssin opiskelijoille sekä vaihto-opiskelijoille.

Työn tarkoituksena oli tutkia mobiilianalytiikkaa aiheena ja tuottaa siitä lisää tietämystä. Opinnäytetyössä tutkittiin myös Business Enterprise-sovellusten ja mobiilipelien mobiilianalyttisiä eroavaisuuksia. Tutkittiin myös kolmea eri mobiilianalytiikan työkalua, joista yhdellä tehtiin käytännön osuus. Työkalua varten tehtiin sovellus, jossa seurattiin painikkeiden ja linkkien availujen tapahtumia.

Opinnäytetyössä käytettiin mobiilianalytiikasta julkaisuista erilaisia e-kirjoja ja internetlähteistä saatua teoriatietoa.

Käytännön osuudessa saatiin selville, miten mobiilianalytiikan työkalu toimii ja kuinka automaattisesti se myös kerää tietoa sovelluksesta. Tämän työkalun toimintaperiaatteeseen ja käyttöön ottoon perehdyttiin sekä teoriassa että käytännössä.

Avainsanat mobiilianalytiikka, sovellus, suorituskykymittari

Sivut 33 s. + liitteet 2 s.

Degree Programme in Business Information Technology
Visamäki

Author	Lasse Pulkki	Year 2017
Subject	Development of Mobile Analytics Teaching Material for Degree Programme in Business Information Technology.	
Supervisors	Tommi Saksa	

ABSTRACT

This Bachelor's thesis was commissioned by the Degree programme in Business Information Technology. Degree programme contains Mobile Analytics course where this Bachelor's thesis is based on. This course is part of the Mobile Programming module that is intended for third class students and exchange students.

The purpose of the thesis was to research mobile analytics as a subject and develop more knowledge. The thesis studied mobile analytic differences in Business Enterprise applications and mobile games. Familiarization with three different mobile analytic tools which one was picked for practical use in a Bachelor's thesis. A simple application was made for the tool, which tracks button and link events.

The theoretical information was gathered from several e-books and internet sources.

The results in a thesis show how mobile analytic tool works and how automatically it gathers information from the application. This tool and its introduction was studied both in theory and practice.

Keywords mobile analytics, application, key performance indicator

Pages 33 p. + appendices 2 p.

SISÄLLYS

1	JOHDANTO.....	1
2	MOBIILIANALYTIikka.....	2
2.1	Yleistä	2
2.2	Mobiilianalytiikan hyödyt.....	3
2.3	Kehitysprosessi.....	4
2.4	Työkalut.....	5
2.4.1	Flurry.....	6
2.4.2	Universal Analytics	6
2.4.3	Apple App Analytics.....	6
3	MITTARIT	7
3.1	Suorituskykymittari	7
3.2	Käyttäjäseuranta	7
3.3	Sovelluksen toiminta.....	8
3.4	Liiketoiminta.....	8
4	TOIMEKSIANTAJAN TAVOITTEET	10
4.1	Tämän hetkinen opetusmateriaali.....	10
4.2	Koulutusohjelman tarpeet	10
4.3	Tavoitteet - Opintomateriaalin kehitys.....	11
5	SOVELLUSTYYPPIEN VERTAILU	12
5.1	Sovellustyypit	12
5.1.1	Enterprise Business-sovellus	12
5.1.2	Mobiilipeli.....	14
5.2	Yhteenvedo sovellustyypeistä	17
6	FLURRY.....	18
6.1	Yleistä Flurrystä	18
6.2	Toimintaperiaate.....	18
6.2.1	Eventit.....	19
6.2.2	Funnelointi.....	20
6.3	Käyttöönotto	22
7	FLURRY ANDROID SOVELLUKSESSA.....	24
7.1	Sovellus.....	24
7.2	Käyttötapaus	26
8	YHTEENVETO	29
	LÄHTEET	30

Liitteet

Liite 1

Rekisteröinti luokan koodi

1 JOHDANTO

Työn toimeksiantajana toimii Hämeen ammattikorkeakoulun tietojenkäsittelyn koulutusohjelma, jonka toiminta on Hämeenlinnan Visamäessä. Tuotan tietojenkäsittelyn koulutusohjelmalle lisää opetusmateriaalia olemassa olevalle Mobile Analytics-kurssille, jossa perehdytään mobiilianalytiikkaan. Kurssi on osa Mobile Programming moduuliin kokonaisuutta.

Tämän opinnäytetyön tavoitteena on tutustua mobiilianalytiikkaan. Miten sitä voi kerätä ja analysoida, tutkia aihealuetta ja kehittää tietämystä. Samalla tavoitteena on tutustua ilmaiseen Flurry Analyticsiin, joka on Yahooon omistama mobiilianalytiikka-työkalu, siihen mitä tällä työkalulla voidaan tehdä ja mitä hyötyjä se tuo yritysten mobiilisovelluksiin. Opinnäytetyössäni tutkin myös sovellustyyppien mobiilianalyttisiä eroavaisuuksia ja minkälainen vaikutus kerätyllä tiedolla on mobiilisovellusten kehittämisprosessissa.

Työn tavoitteena on myös kehittää omaa tietämystäni aiheesta, jotta ymmärrän paremmin mobiilianalytiikkaa ja luoda sen pohjalta hyvä opinnäytetyö. Aiheesta löytyy paljon linkkejä sekä muita lähteitä ja työn tarkoituksena on tiivistää tätä aihealuetta sopivan kokoiseksi kokonaisuudeksi. Tavoitteena on myös tutustua mobiilipeleihin sekä Enterprise Business-sovelluksiin ja pohtia eroavaisuuksia. Käytännön osalta rakennan Android sovelluksen, mihin sisällytän tietokantaan rekisteröitymisen ja kirjautumisen. Tällä sovelluksella seurataan linkkien ja painikkeiden eventtejä eli tapahtumia. Näillä nähdään, kuinka monta kertaa niitä on painettu sekä kuinka usein keskimääräisesti nämä tapahtumat laukeavat.

Mobile Programming on englanninkielinen moduuli Hämeen ammattikorkeakoulussa ja se on tarkoitettu tietojenkäsittelyä kolmannella vuosikurssilla opiskeleville ja sitä tarjotaan myös koulutusohjelman vaihto-opilaille.

2 MOBIILIANALYTIikka

2.1 Yleistä

Analytiikka on tietojen keräämistä ja niiden hyödyntämistä. Mobiilianalytiikassa se pätee sovellusten parantamiseen, ongelmanratkaisuihin, taloudellisiin ja sovelluksen käyttäjäkuntaa kasvattaviin tarkoitetuilla mitareilla mobiililaitteilla. Tämä mahdollistaa menneiden tietojen ja tulevaisuudennäkymien tarkastamista kehittäjille. Ennen analyysi tapahtui tutuilla Excel ja SQL:llä ja ne ovat nykyään korvattu monilla työtaakkaa helpottavilla työkaluilla. (Amplitude n.d.) Ne ovat sisäänrakennettu sovellukseen ja ne lähettävät pieniä tietoja sisältäviä viestejä keskuspalvelimelle sovelluksen ollessa päällä. Kaksi poikkeusta analytiikan keräämisessä on kysyä käyttäjiltä, että onko heillä halua antaa tietoa sovelluksessa. Toinen hyvin yleinen ongelma on se, että ei ole internet-yhteyttä palvelimille. (Harty & Aymer 2016, 15–16.)

Maailmanlaajuisesti vuonna 2015, internetin levinneisyys mobiilikäyttäjille on ylittänyt puolet maailman väestöstä ja keskivertoinen nuori mobiililaitteen kuluttaja käyttää päivittäin hieman päälle kolme tuntia aikaansa verkkosisältöön. (Statista n.d-a.) Sovellusten käyttö on yleistynyt parempien 4G-verkkojen sekä nopeiden langattomien verkkoyhteyksien lomassa. Nämä antavat tarvittavan nopeuden sovelluksiin. Tämänhetkessä mobiilimaailmassa käyttäjät odottavat älykästä ja kontekstiteitoista sovellusta. Hyvä sovellus kerää käyttäjien tarpeiden mukaista dataa ja on käytettävissä missä vain ja millä tahansa laitteella. Silloin, kun nämä vaatimukset eivät täyty, niin kuluttajat etsivät uuden sovelluksen tarpeidensa täyttämiseksi. (Appcelerator n.d., 1.)

Nykyajan sovellukset ovat uniikkeja mahdollisuuksia ja eikä niitä pitäisi sekoittaa sovellusversioon niiden internet-sivusta. Parhailta sovelluksilla on tarttuvuutta, joilla saavat käyttäjät pysymään ja myös tarjoavat jotain arvokasta käyttäjälleen. (Drell 2013.) Vuonna 2009 sovellusten latausmäärä oli maailmanlaajuisesti 2,52 miljardia ja vuodelle 2017 on ennustettu latausmäärän kasvavan 268,69 miljardiin. (Statista n.d-b.) Tämä myös tarkoittaa, että jollei tälläisen kilpailun aikana ei seuraa sovellustaan tarpeeksi tehokkaasti ja kerää oikeaa dataa, sovelluksesta ei välttämättä tule suosittu. (Appsee n.d.) Sovelluksen latauskerrat ensisijaisena mittarina ei kerro tarpeeksi hyvin, kuinka menestynyt se on. Nykyään lähemmäs 22 prosenttia ladatuista sovelluksista ei ole avattu kuin vain keran. (Localytics n.d., 6.)

Sovelluskauppojen arvosteluna toimii tähtimenetelmä, jonka avulla kuluttajat arvostelevat sovelluksen 1–5:een tähdellä. Yhden tähden sovellus ei kovin kauan pysy elossa, kun taas viiden tähden sovellus lyö itsensä läpi markkinoilla. Tämä arvostelutapa on erittäin karkea sovelluskehittäjille. Tähdet pelkästään eivät anna sovelluskehittäjille ollenkaan ymmär-

rystä mitä heidän tuotteessaan on vikana ja mikä on paras ratkaisu seuraavalle versiolle (Appcelerator n.d., 2.) Tämän perusteella analytiikka ratkaisuna tarjoaa helpon tavan selvittää syvemmin ongelman juuren.

2.2 Mobiilianalytiikan hyödyt

IBM:llä työskentelevä yhdysvaltalainen Michael Darden kiteyttää mobiilianalytiikan tuomat hyödyt sovellusten kehittäjälle seitsemään kohtaan blogissaan. Toistuvat kaatumiset voivat vaikuttaa sovelluksen luokitukseen tai kehittäjän maineeseen. On tärkeää olla tietoinen kaatumisista ja nopeasti ratkaista niitä. Hyvällä monitoroinnilla voi priorisoida korjausten tärkeyttä tarkastamalla, kuinka moneen käyttäjään ongelma vaikuttaa. (Darden 2016.)

Ongelmien kartoittaminen voi ohjata sovelluksen kehittäjiä ongelman juureen. Kaatumisten tapahtuessa perustieto kaatuneista laitteista, käyttöjärjestelmä- ja sovellusversioista voi olla hyödyllistä. Enemmän hyödyllistä on nopea katsaus tiettyyn tietoon stack traceissa, jossa seurataan ajonaikaisia virheitä raporttina. Mikäli se on riittämätön, voidaan antaa sovelluskehittäjille pääsyn asiakkaidensa lokeihin analytiikan konsoliin ja sillä voidaan säästää aikaa rutkasti. (Darden 2016.)

Sovellukset ovat entistä kehittyneempiä, mikä on johtanut lisääntyneeseen riippuvaisuuteen palvelimen ohjelmiston palveluiden suorittamaan raskaimpia tehtäviä. Kehittäjien täytyy olla hyvin tietoisia sovelluksen vuorovaikutuksista, koska latenssi tai katkokset vaikuttavat suoraan sovelluksen käyttäjään. Monitoroimalla kuinka paljon aikaa kuluu verkkopyynnöllä saavuttaa sovellus, voidaan nopeasti tunnistaa ja tutkia poikkeamia. (Darden 2016.)

Sovelluskehittäjät ovat usein rasittuneita siitä, että he joutuvat tukemaan monta versiota, heidän sovellukselle. Kehittäjän on jossain vaiheessa päätettävä, milloin lopettaa virheidenkorjausta ja vanhempien sovellusversioiden päivittämistä. Mobiilianalytiikka vertailee sovelluksen käyttöä eri versioissa sekä hankkii kehittäjille tarvittavan tiedon, minkä perusteella voidaan tehdä tietoinen päätös. Sen sijaan aikataulua heikentävällä ei tuetulla vanhemmalla versiolla, mobiilianalytiikkaa käyttävät sovelluskehittäjät voivat valita enemmän käyttäjäkeskeisemmän lähestymisen, jossa he asettavat minimin aktiivikäyttäjien kynnyksen tukemattomalle versiolle. (Darden 2016.)

Sovellusalustan myyjät parantavat kehittäjien tarjontaa ja jokainen uusi julkaisu, tuo suuren määrän uusia toimintoja. Kuitenkaan mobiililaitteiden käyttäjät eivät heti omaksu uusia sovellusalustan versioita, joten kehittäjät joutuvat olemaan varovaisia, heidän jakautumisessa sovellusalustan omaksumisessa heidän käyttäjäkannassa. He ymmärtävät minkä kokoinen ryhmä nykyisistä käyttäjistä aikovat tai eivät aio nauttia uuden version toiminnoista. (Darden 2016.)

Mobiilianalytiikan paketit tulevat tiettyinä paketteina kehittäjille, ellei niitä muokkaa, osalle sovelluskehittäjille yksi koko ei riitä. Kehittäjät voivat käyttää mukautettuja tietoja parantamaan virheenkorjausta tai ilmoittaa mitä ominaisuuksia käytetään eniten ja mitä käytetään vähiten. (Darden, 2016.)

Nykypäivän mobiiliympäristössä sovellukset joko onnistuvat tai epäonnistuvat. Sovelluksen nopeuden, toteutuksen tai asiakastytyvyyden perusteella on suuri vaikutus onnistumiseen. Lyhytikäinen huono käyttökokemus voi vaikuttaa sovelluksen maineeseen. Siksi monitoroiminen tärkeistä luvuista esimerkiksi kaatumisten määrästä ja sen avulla voidaan toimia virheenkorjauksessa nopeasti. Hälytykset sidottuna mobiilianalytiikkaan, kohentavat sovelluskehittäjien toiminta-aikaa, koska ne ilmoittavat heti ongelman ilmaantuessa tai jopa käynnistävät toimia kehittäjän puolesta. (Darden, 2016.)

Mobiilianalytiikan hyödyt ovat sovelluskehittäjille ja mobiilisovelluksille erinomaisia. Ne alentavat heidän työtaakkaansa, ongelmanratkaisun nopeutta ja nostavat käyttäjäkokemusta sekä sovelluksen toimivuuden luottavuutta merkittävästi.

2.3 Kehitysprosessi

Maailmanlaajuisilla sovellusmarkkinoilla kilpailu on kovaa, ja se aiheuttaa sovellustenkehittäjille vaikeuksia erottaa oma sovellus tuhansista samankaltaisista. (Beer 2016, 5.) Kaikkien mobiililaitteiden, tablettien ja muiden samankaltaisten tuotteiden käyttöjärjestelmien usea päivittäminen vaikuttaa kehitysprosessiin.

Australialainen Orange Analytics nimeää viisi kohtaa heidän kehitysprosessin vaikutuksesta sovellukseen. Tuotestrategia, minkä näköinen sovellus on, sovelluksen kehitys, analytiikka ja julkaisun jälkeistä mainostamista. Tuotestrategian kannalta on hyvä rakentaa innovatiivinen mobiilisovellus ja varmistaa oikeiden asioiden kohdistaminen kehityksen alussa. Joihinkin kriittisiin kysymyksiin täytyy antaa vastaus tässä vaiheessa ”Kuka aikoo käyttää sovellusta?”, ”Kuluttaja vaiko yritys?”, ”Mitkä ovat tarvittavia toimintoja?”, ”Mitä toimintoja halutaan sisällyttää sovelluksessa?”. Kehityksen puolesta kertovat jakavansa koodin lyhyillä aikaväleillä ja antaa myöntää toistuvat muutokset tuotekehityksen aikana. (Orange Analytics n.d.)

Analytiikan puolelta asiakkaiden pitävyyden ja käyttäjäkunnan käytöksen seuraamisella saadaan maksimoitua liikevaihtoa yritykselle. Sovellustestausta, monitorointia ja analysointia tapahtuu kehityksen aikana ja julkaisun jälkeen. Orange Analytics kertoo myös keräävänsä tietoa sovelluksen tilastoista sisältäen käyttäjähankinnat, lähestymistä, käyttöä ja asiakkai-

den sitoutumista. Tästä antaen tarvittavaa syvällisempää tietoa hyödyn-tään, joka auttaa tunnistamaan käytettävyyden sekä muiden asioiden kannalta. Näin saadaan tietoa, onko heillä ongelmia sovelluksen kanssa. Analysointi on usein tehty aluksi strategisessa vaiheessa, mutta tekeillä olevassa sovelluksessa tehdään viikoittain tai kuukausittain. Näistä ana-lysoinneista saatavat tilastot kertovat kaikki tulevaisuuden vaihtelut ke-hitysprosessissa. Julkaisun jälkeen monet yritykset ja asiakkaat sivuutta-vat mainostamisen. Hyvä sovelluksen design ja kehitys, ei välttämättä riitä markkinoilla. Tähänkin vielä pystytään vaikuttamaan seuraamalla mittareita ja tilastoja jatkossa. (Orange Analytics n.d.)

Suorituskyvyn mittareissa seurataan aktiivisia käyttäjiä ja kuinka usein he käyttävät sovellusta. Samalla seurataan istuntojen aikaa sekä miten monta sovelluksen sivua aukeaa. Näin nähdään kuinka pitäviä käyttäjiä ovat ja miten hyödyllistä sovelluksen sisältö on. Mittarin seuraamisella saadaan tietoa ostoista, tilauksista tai rekisteröitymisestä. Hankintojen menolla ja kuinka paljon saadaan tuloja per käyttäjä, kertovat vastauksia mittarissa. Vastauksia on, kuinka tuottoisa sovellus on sekä saadaan kuva, mitä tulisi kehittää saadakseen positiivisen tuloksen. Suurimpana näistä on käyttäjäkokemus, mikä on tärkeä menestyksen takaaja sovel-luskauppojen kilpailuissa. (Orange Analytics n.d.)

Yksinkertaiset ja siistit ratkaisut sovelluksen muotoilussa ovat kehityk-sessä tärkeää. Käyttäjakeskeinen muotoilu ja sen testaaminen kertovat sovelluksen käytettävyydestä ja tärkeimpänä asiana olevan parhaan käyt-täjäkokemuksen kuluttajille. Tärkeää tämän rakentamisessa on saada sel-lainen sovellus mitä kuluttajat käyttävät, mikä on nykyajan markkinoilla erittäin tärkeää. Huonon sovelluksen kannalta kuluttajat unohtavat hel-posti käyttäjäkokemuksen sekä miettivät saavatko siitä tarvittavan hyö-dyn tarpeeseensa. Käyttäjäkokemus sisällyttää tuotesuunnittelun järjes-tyksen, kuten käyttäjä tutkimuksen, käytettävyydestestauksen, prototyypit ja kehityksen muun muuassa. (Orange Analytics n.d)

2.4 Työkalut

Tässä luvussa esitellään ilmaiset mobiilianalytiikan työkalut Flurry, Googlen Universal Analytics ja Apple App Analytics. Nämä ovat isoimpien yritysten työkaluja. Luvussa käsitellään yleisluonteisesti niiden tarjoamat palvelut sekä niiden hyvät ja huonot puolet.

Työkalut ovat sovelluskehittäjille tarkoitettuja hyödykkeitä, joilla voidaan seurata sijoitetun pääoman tuoton (ROI) suorituskykyä sekä käyttäjäseu-rantaa. Monet näistä työkaluista ovat ilmaisversioita. Ne sisältävät SDK:n, jolla pystytään lisäämään sovelluksen koodiin, mikä mahdollistaa seurannan. Työkalu aloittaa analysoimaan tietoa mitä sovelluksessa ta-pahtuu ja antaa näkymän kojelautaan (Dashboard), josta voidaan seurata haluattua tapahtumaa. (Gauchet n.d.)

2.4.1 Flurry

Alkujaan Flurry ollut oma yrityksensä, mutta nykyään on Yagoon omistama ilmainen mobiilianalytiikan työkalu. Hetkellä Flurry on osa Yagoon Mobile Development Suitea. Android laitteen sovelluksessa SDK asennetaan jCenterin kautta, josta saa Apache Maven paketin. Apache Mavenia tarvitaan build.gradle tiedostossa sekä AndroidManifest.xml-tiedoston muokkaamisessa. Lisätään kutsuja FlurryAgent.init -tiedostoon, johon API-avain eli ohjelmointirajapinta liitetään, jotta sovellus ja Flurry pystyvät keskustelemaan toisensa kanssa. (Flurry n.d-d.)

Hyvänä puolena sen sisältäessä myös monta käyttäjäpolkua sekä funnelointi mahdollisuutta. Haittapuolina voidaan mainita Flurryn tarvetta klikkailun paljous saadakseen tarvittavan tiedon kojelaudasta. Kanssa virheiden ja kaatumisten analyysin tunnistamisen vaikeus ja se, ettei sisällä joukkoanalyysiä. (Gauchet n.d.)

2.4.2 Universal Analytics

Googlen oma työkalu, joka julkaistiin kesäkuussa 2012. Sen SDK julkaistiin ensiksi betassa ja se kuuluu nykyään Universal Analyticsiin. Raportit mitareista jaetaan neljään ryhmään, Hankintaan, käyttäjäseurantaan, käyttäjien käytösseurantaan ja pidettävyyteen. Tärkeä on myös muutosten seuranta, joka on Googlen oma funnelointitapa. Työkaluna Universal Analytics on hyödyllinen, jos tuote on monelle laitteelle ja sisältää sisäänkirjautumisen käyttäjätunnisteen. Esimerkiksi, jos käyttäjä on kirjautunut kolmelle eri laitteelle, niin siitä lasketaan vain yksi uniikki vierailija. (Gauchet n.d.)

Universal Analytics sisältää paljon tarvittavia toimintoja. Työkaluun on helpompi päästä mukaan, jos on tottunut webanalytiikassa toimivaan Google Analyticsiin sekä integraatio Google Playn kanssa, muutoksen seurantaan. (Gauchet n.d.)

2.4.3 Apple App Analytics

Apple App Analytics on Applen oma mobiilianalytiikan työkalu, joka julkaistiin vuonna 2015. Työkalu ainoastaan toimii Applen omassa iOS ympäristössä, mukaan lukien sisältää tvOS -älytelevision käyttäjärjestelmän sovellusten analysoinnin. Pääasiassa seurataan käyttäjien pitävyyttä, marketointi kampanjoita sekä tulojen ja menojen analysointia eli monetisaatiota. Ei sisällä koodia ollenkaan sovellukseen vaan toimii iTunes Connectin avulla keräten tietoja niiltä, jotka ovat antaneet myönteisen vastauksen tietojen jakoon sovelluksessa. Huonona puolena on sovelluksen sisäisen analytiikan rajoitettu tieto, erityisesti liittyen käyttäjien käytösseurantaan. (Gauchet n.d.)

3 MITTARIT

Mittarilla tarkoitetaan keskeistä suorituskyvyn mittaria (KPI). KPI:n avulla pystytään huomiomaan yksityiskohtaiset mittarit, jotka ovat tärkeimpiä mobiilisovelluksen markkinointi suunnitelmassa. Tarkoitus on auttaa olemaan päättäväisempi ja estää ”analyysi paralyysiä”, mikä on tärkeää välttää sovelluksen markkinoinnissa ja voi tapahtua, kun on tarpeeksi eri mittareita kertomassa erilaisia tuloksia. (Kwakyi 2016.) Yhdysvaltalaisen Red Hatin teettämässä tutkimuksessa 3/4 yhtiöistä, jotka käyttävät suorituskyvyn mittareita sovelluksiensa mittaamiseen, näkevät positiivisena pääoman tuoton. (Bourne 2016.)

Yhdysvaltalaisen AppDynamics yhtiön ”16 Metrics to Ensure Mobile App Success”, e-kirjassa jaetaan korkeammalla tasolla mittarit neljään isompaan ryhmään eli suorituskykymittariin, käyttäjäseurantaan, sovelluksen toimintaan ja liiketoimintaan.

3.1 Suorituskykymittari

Suorituskykymittarissa tarkkaillaan käyttäjän kokemusta, käyttäjää ja käyttäjän sovelluksen käyttöä. Myös miten sovelluksessa toimitaan ja miten sovelluksen liiketoimintaa mitataan. Mittarina suorituskyky on ideaalinen ensimmäisenä analysoinnin aloittavana sovelluksen tarkkailijana. Tämä sisältää sovelluksen sisäisten ongelmien mittareita, jotka sisältävät kriittistä dataa toimivuudesta. Tietoverkko ongelmat ovat palveluntarjoajan tai HTTP:n aiheuttamia ongelmia, joka vaikuttaa kaatumisiin tai hitaaseen vastaus aikaan. Sovelluksen keskiarvoinen kaatumisen määrä jokaiselta avaamiskerralta on 1 – 2 prosenttia, joten tämän vaikutusta kannattaa seurata ja yrittää eliminoida kokonaan. Ohjelmoinnin rajapinnan eli API:n viiva on paras saada sekunnin ympärille. Päästä päähän viiveen seuraaminen, josta yli 3-4 sekunnin tai ylitse menevät viiveet ajavat 60 prosenttia asiakkaista pois sovelluksen käytöstä. Seuraamalla sovelluksen rasisusta, näkee miten isoimmat rasisut vaikuttavat sovelluksen toimivuuteen, sovelluksen lataamiseen ja miten se on altis nopeille vaihteluille. (AppDynamics n.d., 3–4.)

3.2 Käyttäjäseuranta

Käyttäjät ovat tärkeitä mobiilianalytiikalle, joten heidän seuraamisensa tuo paljon seurattavaa hyötyä. Aktiiviset käyttäjät ja uudet käyttäjät keräävät näitä tietoja applikaation käytön perusteella maailmanlaajuisesti. Aktiiviseen käyttäjään lasketaan ainoastaan kerran ja käyttäjänä pidetään henkilöä, joka aukaisee applikaation kerran annettuna aikana. Näiden tyypillisenä mittarina toimivat kuukausittainen aktiivinen käyttäjä eli MAU tai päivittäinen aktiivinen käyttäjä eli DAU. He ovat tärkeitä tekijöitä applikaation kannalta heidän käyttäessään sovellusta ja on potentiaalia

tulla lojaaleiksi sekä lisätä tuottoa. (Beer 2016, 8.) Käyttöjärjestelmät sekä versiot kertovat minkälaisilla laitteilla asiakkaat käyttävät sovellusta. Kanssa käyttäjien maantieteellinen sijainti antaa viitteitä, missä menestytään ja missä ei. (AppDynamics n.d., 5.)

3.3 Sovelluksen toiminta

Sovelluksen toimintamittarilla mitataan istunnon pituutta, istuntojen aikaväliä ja kuinka usein kuluttajat palaavat sovelluksen käyttöön, mikä kertoo riippuvuudesta sovellukseen. Istunnon pituudella tarkoitetaan kuinka paljon aikaa käyttäjä käyttää yksittäisellä kerralla ja mitä enemmän sitoutuneita he ovat, niin sitä enemmän aikaa kertyy sovellukseen. Istuntojen aikavälissä haetaan käyttökertojen toistuvuutta ensimmäisestä kerrasta seuraavaan. Pitävyysaste kohdistaa prosenttimäärää asiakkaiden palaamisesta ensimmäisen käyttöpäivän jälkeen. Toiminnalliset ja arvokkaammat asiakkaat, rakentamalla paremmat kohdistus mahdollisuudet ja henkilökohtaistu sovellus kokemus korostuvat tällä mittarilla. (AppDynamics n.d., 6.) Säilyvyys suhteen vastakohtana pidetään vaihtuvuus suhdetta, jossa mitataan kuinka monta prosenttia käyttäjistä, olisi voinut lopettaa sovelluksen käyttämisen kokonaan tietyn ajanjakson aikana. Kehittäjän täytyy löytää syitä mikä voisi aiheuttaa käyttäjien noussevan vaihtuvuusasteen nousun ja korjata asia mahdollisimman nopeasti. Perusidea on pitää vaihtuvuusaste pienenä ja pitävyysaste suurena, joka kehittää käyttäjien pitävyyttä sovellusta kohtaan. (Beer 2016, 11.)

3.4 Liiketoiminta

Liiketoiminnan puolella keskitytään hankinnan kustannukseen, kaupan tuottoon ja elämän ajan arvoon. Kustannuksiin vaikuttavat miten asiakkaat löytävät sovelluksen. He ovat saattaneet löytää sovelluksen sovelluskaupasta kuten Google Play Store, muilta henkilöiltä saaneen kuulopuheen perusteella, maksetun kampanjan tiedolla tai sovelluksen sisäisellä viittauksella. Sovelluksen kaupankäynnin tuottoon haetaan arvoja sisäisesti tapahtuvista kaupankäynneistä, jotka käyttävät mCommercea eli mobiililaitteilla tapahtuvaa ostoa tai myyntiä. Tätä palvelua käyttävissä sovelluksissa voidaan seurata suoraan tätä, mutta myös ei palvelua käyttävistä voidaan saada suuntaa antavaa tietoa. Ensisijaisena tuottomittarina käytetään elämän ajan arvoa (LTV), joka esittää kuinka arvokkaita jokainen käyttäjä on heidän elämänajallaan. (AppDynamics n.d., 7.) LTV -mittarina jaetaan usein kuukausittaiseen keskiarvoon tai per asiakas, saadakseen jonkinlaista kehityssuuntaa tai jonkin tapaista katselmusta kokonaisvaltaiseen markkinointikampanjan luotuun rahalliseen tuottoon. (Beer 2016, 14.) Monesti sovelluksen sisäiset kaupankäynnit eivät toimi, joten sitä mittaillaan hylkäysasteella. Hylkäysasteeseen vaikuttaa esimerkiksi toimintakyky ja käyttäjäkokemus eivät olleet toiveiden mukaisia, sovellus kaatui tai käyttäjän mieli vaihtui. Näiden perusteella voidaan ymmärtää käyttäjän kokemusta ja ymmärtää miksi kaupankäynti

on hylätty. AppDynamics lopuksi nostattaa ”kaikkien mittareiden äidiksi”, sovelluksen tähtiarvostelun. Muualla ei anneta julkista kannanottoa sovelluksen arvostelulle kuin sovelluskaupassa. (AppDynamics n.d., 7.)

4 TOIMEKSIANTAJAN TAVOITTEET

4.1 Tämän hetkinen opetusmateriaali

Moduuli on englanninkielinen ja on tarkoitettu tietojenkäsittelyn kolmannen vuosikurssin opiskelijoille, jossa tutustutaan mobiilisovellusten tuotantoteknologiaan osiin. Mobiilianalytiikka tästä osin on päivän mittainen. Hetkellinen opetusmateriaali koostuu mobiilianalytiikan esittelystä, live demosta Pocket Monsters Cross Platformilla, tutustutaan HockeyApp -sovellusalustaan ja kolme erilaista tehtävää opiskelijoille. Ensimmäiset kaksi tehtävää on opiskelijan henkilökohtaisia, jotka valmistavat häntä viimeiseen ryhmätehtävään.

HockeyApp on mobiilianalytiikan sovellusalusta, joka tukee iOS, Android, OS X, ja Windowsin sovelluksia. HockeyApp tarjoaa auttakseen käyttäjälleen rakentaa tai ladata palveluun oman sovelluksen, jonka HockeyApp jakaa testaajilleen. Tämän lisäksi se tarjoaa kaatumisraportteja, käyttäjämittarin miten käyttäjät käyttävät sovellusta, palautteen testaajilta ja työnkulku integraation. (HockeyApp n.d.)

Pocket Monsters Cross Platform (PocketMonsters n.d.) on Hämeen ammattikorkeakoulun tuntiopettajan Turo Nylundin tekemä ohjelma kursseille, jonka toimintana on kerätä pokemoneja. Sovellus toimii Android, iOS ja Windows ympäristöissä.

Ensimmäisessä tehtävässä opiskelija tutustuu mobiilianalytiikkaan aiheena ja kirjoittaa siitä oman omin sanoin oman tekstin sekä liittyy linkin lähdetiedostoonsa. Toisessa tehtävässä etsitään kolmea analytiikan palvelua, joista kirjoitetaan, omin sanoin niiden jokaisen palvelut ja toiminnot, sisältäen linkin lähdetiedostoonsa. Kolmannessa tehtävässä tehdään ryhmässä, jonka aiheena on valita mobiilianalytiikan palvelu heidän mobiilisovellukseensa. Tehtävässä täytyy perustella miksi valitsi tietyn palvelun, mitä toimintoja käyttää kyseisestä palvelusta, palvelun hinta, mitä sovellusalustoja se tukee, kuinka palvelu integroidaan omaan sovellukseensa sekä esimerkkejä koodauksesta tarvittaessa ja mistä löytyy lisää tietoa palvelusta.

4.2 Koulutusohjelman tarpeet

Koulutusohjelman tarpeena on saada ja kehittää lisää tietämystä koulutuksen Mobile Programming moduulille. Lähetin pienen kyselyn Hämeen ammattikorkeakoulun tuntiopettajalle, mikä on hänen näkökulmansa mobiiliohjelmoinnista, mobiilianalytiikasta opettajana ja mitä aiempaa kokemusta hänellä on aihealueelta.

Mobiiliohjelmoinnin puolelta, hän on ollut tekemässä rentoutumissovellusta, matematiikkapeliä lapsille ja hetkellä on rakenteilla tasohyppely-peli. Kaikki näistä on tehty Corona SDK:lla, joka on enemmän kaksiulotteinen pelimoottori, mutta onnistuu tehdä myös Android kuin iOS version. Hän on myös koodannut kaksi www-sivustoa jQueryllä, jotka näyttävät ja tuntuvat mobiilisovellukselta. Kumpaakin sivustoa pääosin käytetään mobiililaitteilla.

Mobiilianalytiikasta hän on käyttänyt ainoastaan Flurryä ja vain teknisestä näkökulmasta. Flurry on hänen mielestään hyvä, esimerkiksi tapahtumien tekemiseen. Mobiilianalytiikan ymmärtäminen mobiilianalytiikan puolelta hän kertoo sen olevan ehdottoman tärkeää. Esimerkkinä, jos halutaan selvittää miksi pelaajat pelaavat pääosin vain kenttään kolme asti ja tästä voidaan tehdä jonkinlaisia johtopäätöksiä. Opetuksen näkökulmasta hän näkee.

- Hyvä tietää suurinpiirtein, miten analytiikkaa saadaan toimimaan sovelluksessa.
- Osa käyttää analytiikkapalvelun webbiliittymää tuloksia tarkasteltaessa.
- Ymmärtää eventit, funneloinnit sekä muut mobiilianalytiikan termit.
- Opiskelijat osaisivat analysoida ja tehdä johtopäätöksiä tuloksista
- Kuinka tärkeäksi näet mobiilianalytiikan ymmärtämisen ja osaamisen opetuksen näkökulmasta/työelämän näkökulmasta?

4.3 Tavoitteet - Opintomateriaalin kehitys

Opintomateriaalin kehitystä opinnäytetyön aikana on useiden linkkien takaa haettu tietoa ja tästä saatu tieto on tavoitteena tiivistää tähän opinnäytetyöhön. Tavoitteena on tutustua mobiilipeleihin sekä Enterprise Business-sovelluksiin ja pohtia eroavaisuuksia. Tavoitteena myös hankkia tarvittavaa yleistietoa kolmesta mobiilianalytiikan työkalusta ja perehtyä yhteen näistä, joka on Flurry. Flurryllä on tavoitteena rakentaa käyttötapaus. Flurryä varten rakennan Android sovelluksen, jossa seuraan tapahtumia eli eventtejä.

5 SOVELLUSTYYPPIEN VERTAILU

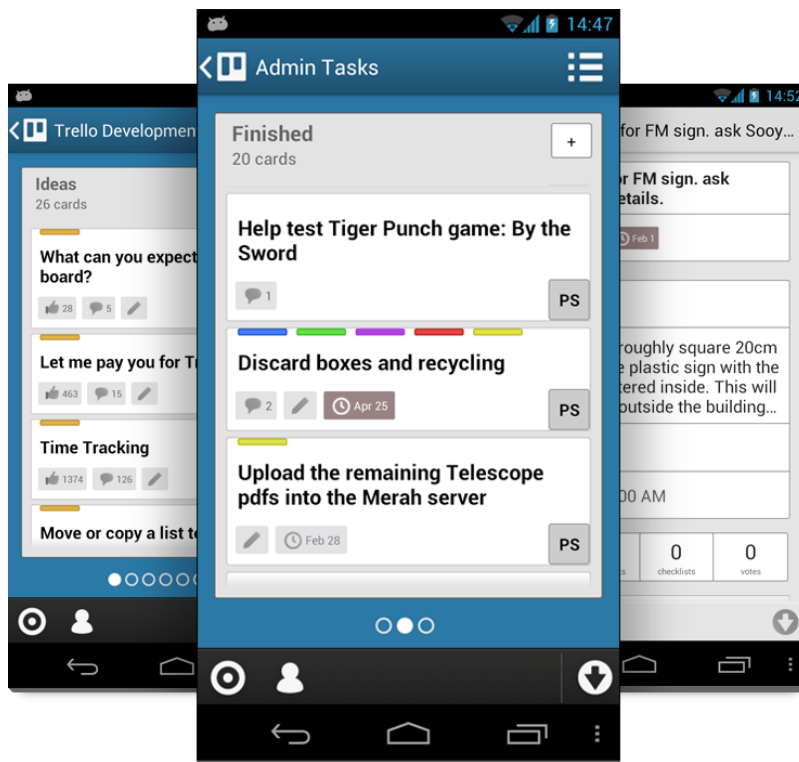
5.1 Sovellustyypit

Tässä osassa ei perehdytä tiettyihin sovelluksiin vain yleisesti mitä näillä sovellustyypeillä halutaan analysoida. Kanssa perehdytään miten niiden käyttö vaikuttaa asiakkaisiin ja minkälaista hyötykäyttöä näillä voidaan tehdä. Sovellukset voidaan jakaa yleisesti kahteen osin Mobiilipeleihin ja Enterprise Business-sovelluksiin. Näihin sovelluksiin liittyvät yrityskeskeiset sovellukset ja hyödylliset työkalut.

5.1.1 Enterprise Business-sovellus

Älypuhelinkehityksessä myös tietokoneilta tutut ohjelmat tai niiden muunnelmat ovat löytäneet itsensä sovelluskauppoihin. ”Global Mobile Enterprise Business Applications Revenue Forecast 2016-2022” raportin mukaan vuonna 2016 sovellusten tuotto nousi 74 miljardiin dollariin ja vuoteen 2022 mennessä ne nousivat 128 miljardiin dollariin. (Luk & Brown 2016.) Business to Employee (B2E) mobiilisovelluksilla analytiikka, voi nostaa huomattavasti työntekijöiden tuottavuutta ja parantaa organisaatioiden prosesseja. (EvokeTechnologies 2015.)

Sovellukset ovat auttaneet parantamaan ryhmätyöskentelyä ja kommunikointia työntekijöiden nähdessä viestit sekä viestittämään puhelimen avulla missä vain ja milloin vain. Näihin sovelluksiin lasketaan muun muassa keskusteluohjelmat kuten Slack ja Skype For Business, projektinhallintatyökalut kuten Trello (Kuva 1.) sekä Windows ympäristöstä tuttu Microsoft Office. Nämä enemmän painottuvat hyödykkäisiin työkaluihin, tai yrityskeskeisiin sovelluksiin. Sovellukset voivat olla maksullisia tai ne voivat olla tiettyyn pisteeseen saakka ilmaisia, minkä jälkeen niiden tarjoamat lisäosat saattavat olla maksullisia. Välttämättä Enterprise-sovelluksen tie ei ole ruusuinen. Voi olla vaikeaa saada implementoitua menestynyt sovellukseen ja vaikeampi saada mitattua suosiota tästä. (Popp 2015.)



Kuva 1. Näkymä projektinhallinta sovellus Trellostä. (Favoritemedium n.d.)

Enterprise-sovelluksen menestyksen mittaamiseen vaikuttavat tässäkin sovelluslataukset, joka kertoo yksinkertaisesti suosion määrän. Kanssa toisina samankaltaisena mittarina toimii sovelluksen käytön seuranta sekä sovelluskauppojen tähtiarvostelu menetelmä. Mobiililaitteiden seuranta antaa viittauksia minkä tyyppisillä laitteilla sovellusta käytetään ja minkälaiset ohjelmistoversiot sekä käyttöjärjestelmän ne sisältävät. Tällä myös nähdään ongelmatilanne jailbreakattujen laitteiden eli iOS-laitteilla, josta on poistettu estot, asentaa kolmannen osapuolen ohjelmistoa. Nämä jailbreakatut laitteet voivat olla tietoturvariskejä yhtiölle, että sovellukselle. Muita ongelmatilanteiden kannalta hyviä seurattavia Enterprise-sovellukselle on kaatumisseuranta, viiveen seuranta ja kuinka sovellus kestää isoa kuormitusta sekä miten se mukautuu sen ollessa kuormituksessa. ([x]Cube LABS n.d.)

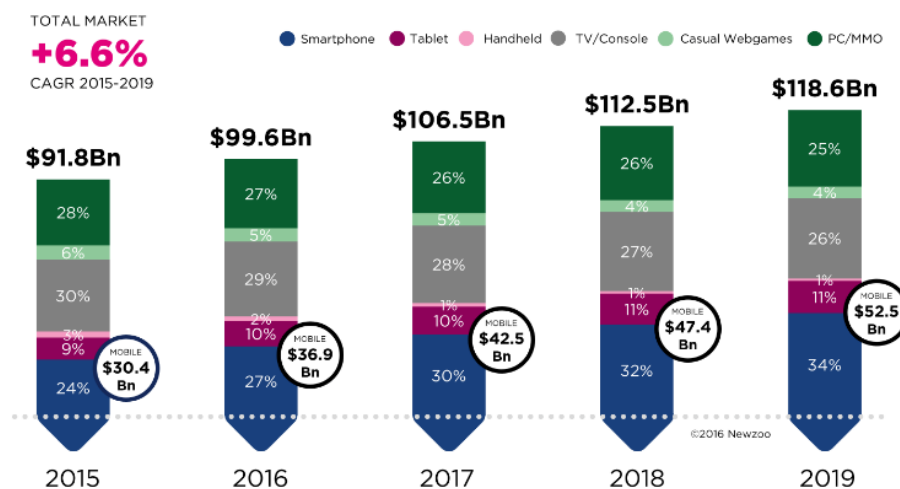
Sovellukselle uskollisuutta varten voidaan vertailla päivittäisten aktiivisten ja kuukausittain aktiivisten käyttäjien seuranta. Tällä saadaan tärkeää tietoa käyttäjän pitävyydestä ja sovelluksen käytöksestä. Käyttäjien säilyttäminen on suurin haaste sovelluksille, joten niitä varten täytyy seurata toimintojen toimivuutta sekä kuinka suosittu itse sovellus on. (Sharma 2016.)

Yrityksen rahallisen tuottavuuden kannalta monetisaatio on yksi isoimmista asioista käyttötarkoituksessa. Monetisaatiolla tarkoitetaan mekanismeja tai tapoja millä saadaan kuluttajia käyttämään rahaa tiettyyn tuotteeseen (Hiltunen, Latvala & Kaleva. 2013, 25.) Pääoman tuoton

hankkiminen analytiikan avulla, mobiilisovellus hallinta yhtiön App47 mukaan, mittareiden tuloksissa näkyvät kuinka usein sovellusta on käytetty sekä minkälaiset käyttäjät käyttävät. Näin ollen saadaan parempi mielikuva minkälaisen vaikutuksen siitä saa yhtiölle. Virhetilanteiden satuesssa kuten sovellusten kaatumiset vaikuttavat negatiivisesti pääoman tuoton laskemisessa, joka vie työntekijöiltä ylimääräistä aikaa korjauksiin. Tietäen kuinka luotettava tehty sovellus on, sitä varmemmin pääoman tuotto kasvaa. Tärkeää on kanssa nähdä, minkä tapaisia ostotapah- tumia tapahtuu sovelluksen sisällä. Lopuksi näistä saadaan pääteltyä, onko sovellus tarpeeksi tuottava yhtiölle. (App47 2014.)

5.1.2 Mobiilipeli

Pelien kehittäjät ovat pikkuhiljaa vuosien saatossa siirtyneet mobiililaitteiden pelien kehitykseen, sen kasvavan käyttäjäkunnan myötä. Tutkimusyhtiö Newzoon mukaan, vuonna 2016 mobiilipelit ohittavat ensimmäisen kerran tietokoneiden tuotot samalla nostaten edellisvuoden tuottoja 8,5 prosenttia. (Kuva 2.) Samalla he ennustavat mobiilipelien syövän tulevaisuudessa muiden laitteiden markkinoita. (Newzoo 2016.) Itse peleistä hankitaan pelaajien kautta tietoa, heidän käytöksen avulla, antaen näkemystä pelin kehitykseen.

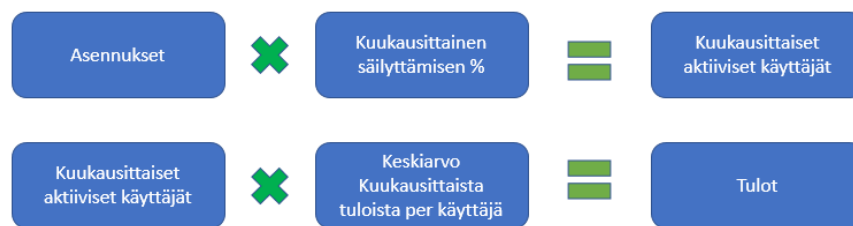


Kuva 2. Newzoon julkaisema ennuste vuosille 2015-2019. (Newzoo 2016.)

Nykyään kertaostoisista peleistä on siirrytty free-to-play (f2p) peleihin eli ilmaispeleihin. Ilmaispeilit ovat olleet nykyaikana tuotollisesti kannattavia, jotka voivat tuoda ajansaatossa isoja määriä rahallista tuottoa kehittäjille. Näiden pelien rahallinen tuotto saadaan ladatuista osista (DLC), mikro-ostoksiin pelin sisällä tai pelissä näkyvinä mainoksina. (Walz 2015a.)

Apptentiven Alex Walz tiivistää blogissaan viideksi tärkeimmäksi mobiilipelin mittareiksi keskiarvot tuloista käyttäjää kohden (ARPU), maksua

asennusta kohden (CPI) sekä lojaalia käyttäjää (CPLU), käyttäjien toiminnan seurausta, kuinka paljon käyttäjät pitävät pelistä ja kuinka paljon käyttäjistä pysyy pelaamassa peliä. Laskemalla keskiarvon tuloista käyttäjää kohden pystytään määrittämään kahden muun mittarin kanssa pelin suosiota. Maksua jokaiselle lojaalille käyttäjälle voidaan hyödyntää budjetoidessa markkinointi ja mainosmenoja. Yleinen sääntö on markkointiin kulutetut rahat ovat hyvin käytetty, niin kauan kuin keskiarvo tulot käyttäjää kohden ovat paremmat kuin menot jokaista lojaalia käyttäjää kohden. Toisena asiana pitävyyttä hyödynnetään sitä kertomalla enustetta elämänajan arvolle. Tulot voidaan laskea yksinkertaisesti (Kuva 3.) (Walz 2015b.)



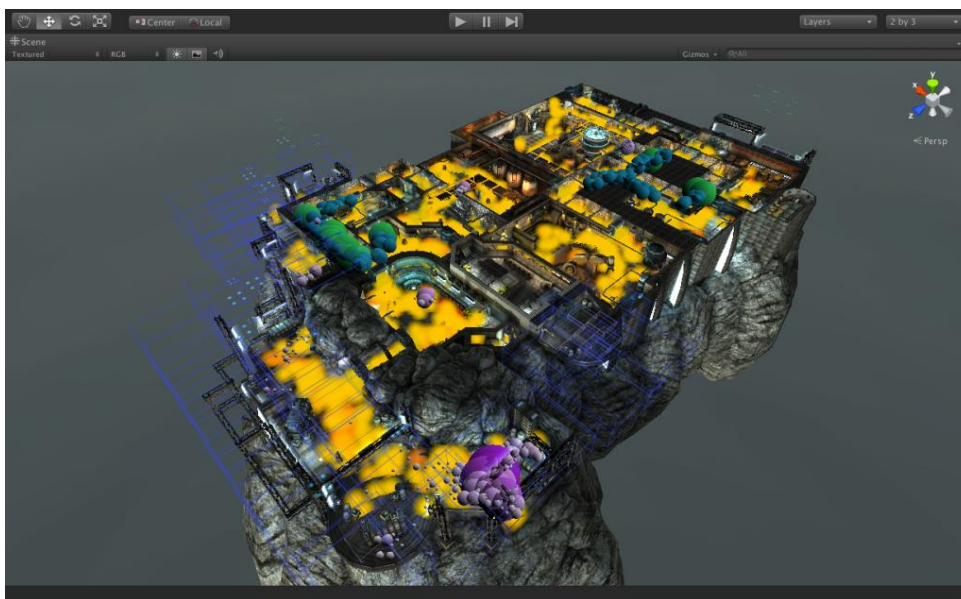
Kuva 3. Havainnointi sovelluksen tuloista. Muokattu alkuperäisestä. (Walz 2015b.)

CPI mittaa asiakkaiden hankintojen menoa, jotka ovat asentaneet sovelluksen ja CPLU mittari katsoo enemmän menoja, jokaisen aktiivisen käyttäjän kohdalle. ARPU:n kanssa saadaan laskettua pääoman tuottoa investoinnissa ja markkointia varten, ARPU:n täytyy olla arvoltaan parempi kuin CPLU. (Walz 2015b.)

Toiminnan seurauksessa käyttäjien istuntoa seurataan, kuinka kauan yhdellä istunnolla käytetään aikaa, miten paljon he aukaisevat sovelluksen osia ja kuinka useasti he aukaisevat sovelluksen. Käyttäjien pelin sisäisten suoritusten onnistumisprosenttiin, viestien ja kehotusten saannin määrään. Näin ollen kiintyneemmät asiakkaat ovat todennäköisiä antamaan suosituksia sovelluksesta muille henkilöille ja ovat todennäköisempiä lojaalimpia sekä tuottavampia pelille. (Walz 2015b.)

Kuinka paljon käyttäjät pitävät pelistä annetaan kyselyllä pelin sisällä ”Pidätkö tästä sovelluksesta”, joissa käyttäjillä on kolme valittavaa kohtaa. Kyllä, ei tai kehoituksen hylkääminen. Eniten tässä mitataan myönteisen vastauksen prosenttia kuin muita. Simppelinä ja tarkkana pidetty mittari kertoo yleisen asiakaskokemuksen sisältämällä pienen työn asiakkaalta. Tulosta voidaan verrata sovelluskaupan arvosanoihin ja päätellä tämän kertovan enemmän asiakaskokemuksesta kuin arvosanasta. (Walz 2015b.)

Näiden lisäksi on tärkeää seurata pelaajien kehitystä, kuten hahmojen tasoja, millä tasolla ovat ja kuinka paljon ovat keränneet tärkeitä esineitä esimerkiksi Donkey Kongista tuttuja banaaneja. Tasohyppelypelin kannalta on hyvä seurata mihinkä kohtiin pelaajat menettävät pelin elämäpisteitä tai jäävät jumiin. Näin ollen pelin tekijä ymmärtää kartan tasapainoisuutta. Tähän on ratkaisuna esimerkiksi GameAnalyticsin Heatmaps, joka seuraa näitä (Kuva 4). Myös hyvä seurata kriittistä tietoa mihin pelaajat käyttävät pelin sisäistä rahayksikköä kuten taitotasoja nostattavia esineitä ja tästä he saavat tietoa minkälaisia esineitä ostetaan. Niitä esineitä joita ei osteta, niin niille haetaan ymmärrystä miksei esineitä osteta, samalla yrittäen tasapainottaa peliä. (Drachen 2013.)



Kuva 4. Heatmaps. (Biterati 2016.)

Asiakkaiden pitävyydellä katsotaan miten asiakkaat ovat jääneet aktiivisiksi käyttäjiksi. Tähän arvoon vaikuttaa pelin sisäiset tekijät kuten rikkinäinen peli eikä palkitsettavuutta. Toisina kriittisinä ongelmina voidaan pitää, ettei peliä päivitetä ollenkaan tai ei riittävän usein, johon voi vaikuttaa huono pelin testaaminen. (Walz 2015b.)

Kanssa suurimpia negatiiviseen käyttäjäkunnan kasvun tekijöitä ovat liian vaikeasti saada pelaajat sisäistämään pelin, johonka voi vaikuttaa huono perehdytys peliin sekä tutoriaalinen liiallinen pituus estää pelaajaa liian kauan aloittamasta pelaamista. Lisäksi liian pitkät peli-istunnot merkitsevät paljon, sillä kaikilla pelaajilla ei ole saman verran aikaa kuin kokeneemmilla pelaajilla. Myöhemmässä vaiheessa pelaajan sisäistettyä pelin vaikeuksia voi tulla sisällön kanssa, kuten samojen asioiden koko aikainen tahkoaminen. Toistuvuus aiheuttaa tylsistymistä, joka vaatisi uutta ja mielenkiintoisuutta herättävää asiaa saadakseen pelaajaan työstämään lisää pelitunteja. Vaikeustason nopea vaihtuvuus vaikeaksi vaikuttaa myös erittäin negatiivisesti. Sosiaalisempien pelien eli moninpelien kannalta yhteisön kanssapelaajien negatiivinen asenne keskusteluikkunassa.

Esimerkiksi kokeneempi pelaaja haukkuu uudempaa pelaajaa, aiheuttaa uudemman pelaajakannan luopumista pelistä huonon kokemuksen takia. Lopuksi nousee kaksi isointa asiaa mitä kehittäjiä pitäisi ratkoa on tylsyys, turhautuminen ja siitä johtuva tuska, jonka kehittäjän täytyy itse miettiä miten saada tämä ratkottua. (Drachen 2013.)

5.2 Yhteenveto sovellustyypeistä

Kummankin sovellustyyppin pohjana on yrityksillä rahanteko sekä erilaisien mittareiden käyttäjäseuranta. Analytiikan puolella kummallakin on samankaltaisuuksia mittariseurannalla. Mobiilipelit eroavat lopulta sovellustyyppinä Enterprise Business-sovelluksista, niiden seuraamalla ei-käyttäjä riippuvaisia tekijöitä. Mobiilipeleissä seurataan käyttäjän kehitystä eri tavoin kuin Enterprise Business-sovelluksessa. Tähän liittyvillä pelin muiden tekijöiden, kuten kartassa tapahtuvien kuolemien ja ongelmatilanteiden sekä pelien sisäisen kaupan esineiden ostotapahtumien seurannalla, joiden kautta voidaan tasapainottaa pelejä.

6 FLURRY

Tässä osassa perehdytään mobiilianalytiikan työkaluun Flurryn. Valitsin Flurryn muista mobiilianalytiikkaan työkaluista, sen ollessa yksi vanhimista kehitysalustoista sekä myös yksi suosituimmista. Tässä luvussa perehdytään Flurryn yleistietoon, sen toimintaperiaatteista ja miten se voidaan käyttöönottaa. Tavoitteena käyttöönotossa on tutustua mitenkä saada se toimimaan sovelluksessa ja ymmärtämään, miten se kerää tietoa. Samalla tutkitaan miten tiedot esiintyvät kojelaudalla.

6.1 Yleistä Flurrystä

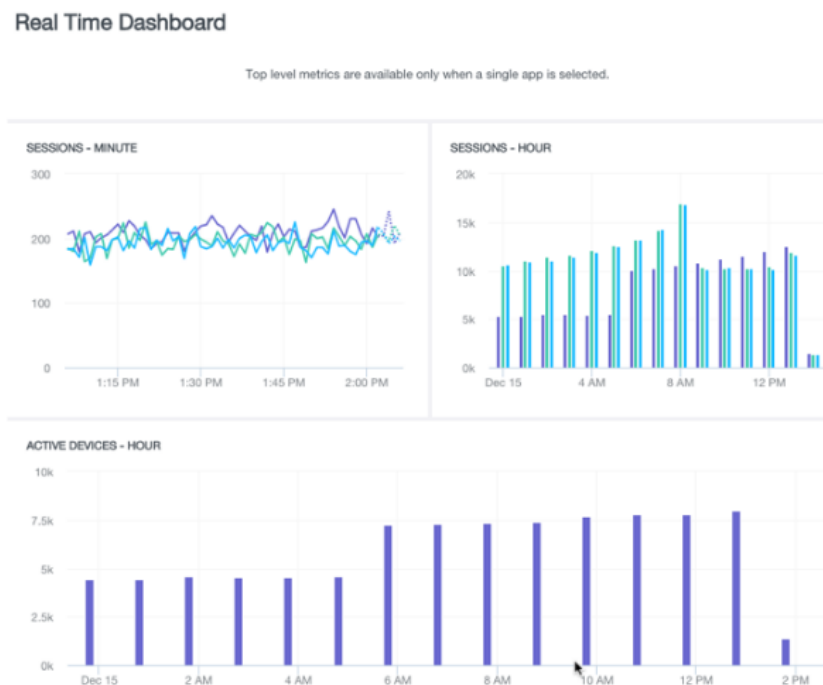
Yhtiön perustajan Searn Byrnesin haastattelussa Flurry aikoinaan perustettiin omana mobiilisovellus yrityksenä vuonna 2005. Silloin vielä sovelluksia asennettiin simpukkapuhelimiin mitkä sisälsivät huonot prosessorit ja näytön tarkkuudet. He huomasivat asennettuaan mainosalustan heidän sovelluksilleen, sen olevan isompi mahdollisuus tulevaisuutta varten. (Thalamus. 2015.) Vuoden 2014 heinäkuussa, Yahoo hankki yhtiön 200-300 miljoonalla dollarilla (Lunden 2014) ja on nykyään osa Yahoos Mobile Developer Suitea, joka keskittyy analytiikkapalveluihin. Flurry toimii ainoastaan Android ja Applen iOS alustalla sekä Applen kellon ja älytelevision kanssa. Flurry Analytics SDK tarjoaa käyttäjälle työkalut ja resurssit, joita tarvitaan saadakseen syvemmän tason tietoa käyttäjien käytöksestä sovelluksessa. (Flurry. n.d-a.)

Flurry tarjoaa ilmaisen palvelun samalla tavalla kuin Google, mutta ei ole data kattoa ja omistaa kaiken tiedon minkä se prosessoi. (McCobb 2012.) Flurry eroaa muista kilpailijoistaan korkealla tasolla kerätyn tiedon kokoon ja laatuun sekä SDK:n implementoinnin helppouteen. (Flurry n.d-c.)

6.2 Toimintaperiaate

Flurry Analytics SDK antaa työkalut ja resurssit syvempään ymmärtämiseen käyttäjien käyttäytymiseen sovelluksessa. Kehittynyt analyysi monimutkaisista tapahtumista mittaamisella, segmenteillä ja funneleilla, parempaan seurantaan käyttäjistä. Mittaamisella voidaan laskea "ad-hoc" mittaria sovelluksen tiedoista käyttämällä tehokasta keräystä, suodatusta ja ryhmittämiskykyä hyödykseen. Segmentillä tarkoitetaan käyttäjien ryhmittämistä monien vaikutusten pohjalta kuten asennuspäivästä, maantieteellisestä ja aktiivisuus tasoista. Funnelointi taas seuraa käyttäjien kehitystä, milloin he liikkuvat tiettyjen askelten mukaan. Tämä mahdollistaa visualisoimaan onnistumisprosenttia käyttäjien, jokaisen tapahtuman läpivientiä ja mikä askel tuottaa ongelmia asiakkaille. (Flurry n.d-f.)

Flurryn tarjoamalla reaaliaikaisella seurannalla (Kuva 5) mahdollistaa seurata asiakkaiden päivityksiä, käyttäjä hankintoja tai parannuksia. Maksimissaan sovelluksia pystyy seuraamaan viittä kerrallaan ja sessiot näkyvät minuutteina ja tunteina sekä nämä päivittyvät joka 15 sekunti. (Flurry n.d-a.)



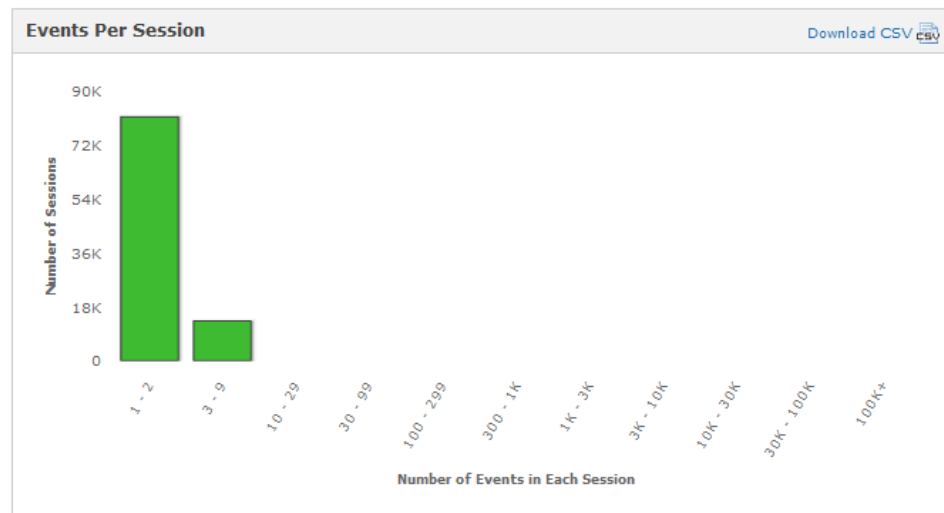
Kuva 5. Kojelaudan reaaliaikaisen seurannan esimerkki. Seurataan istuntojen minuutteja sekä tunteja. (Flurry n.d-a.)

Mahdollisuutena on vertailla monia mittareita samaan aikaan. Käyttäjien kohdalta Flurry tarjoaa auttamaan ymmärtämään käyttäjiä sekä heidän kiinnostuksien kohteista, tarkkoja arvioita käyttäjien iästä, sukupuolesta ja käyttäjien hankinnasta. Sovelluksen seurantaan on analysointia virheistä ja kaatumista sekä samanvertaisten sovellusten testaamista. Kehittäjät pystyvät keräämään verkotonta dataa käyttäjiltä seuraavalla kerralla, kun he ovat verkollisessa tilassa. Tiedot ovat säilöttynä laitteella paikallisesti ja tiedot lähetetään silloin keskuspalvelimelle. (Flurry n.d-a.)

6.2.1 Eventit

Itse räätälöidyt tapahtumat eli eventit ovat Flurryn tapa seurata tiettyjä sovelluksen sisäisiä käyttäjien toimia. Esimerkkinä milloin soitetaan musiikkikappaletta, otetaan kuvia tai suoritetaan pelitaso loppuun. Tapahtumien raportointi tapahtuu jokaiselle tapahtumalle, jonka on merkinnyt sovellukseen. Ensimmäiseksi annetaan tietoa, milloin tapahtuma on laukaistu ja kuinka monta kertaa tietyllä istunnolla. (Kuva 6.) Tämä auttaa ymmärtämään, kuinka useasti tapahtuma käynnistyy käyttäjien istun-

noissa. Esimerkkinä uutissovellus haluaa tietää miten monta kertaa jokaista istuntoa kohti ”artikkeli luettu” tapahtuma on käynnistetty. (Flurry n.d-e.)



Kuva 6. Havainto tapahtumista, jokaista istuntoa kohden. (Flurry n.d-e.)

Flurry automaattisesti tarjoaa yksityiskohtaisen käyttäjäpolun, jokaista luotua tapahtumaa kohden. Tapahtuma loki antaa yksityiskohtaisen näkökulman kaikista käyttäjäistunnosta, jotka sisältävät tapahtumia. Nämä sisältävät millaisella laitteella on vierailtu, aikaleiman, käyttäjätunnisteen numeron ja kuinka monta tapahtumaa on laukaistu. (Flurry n.d-e.)

6.2.2 Funnelointi

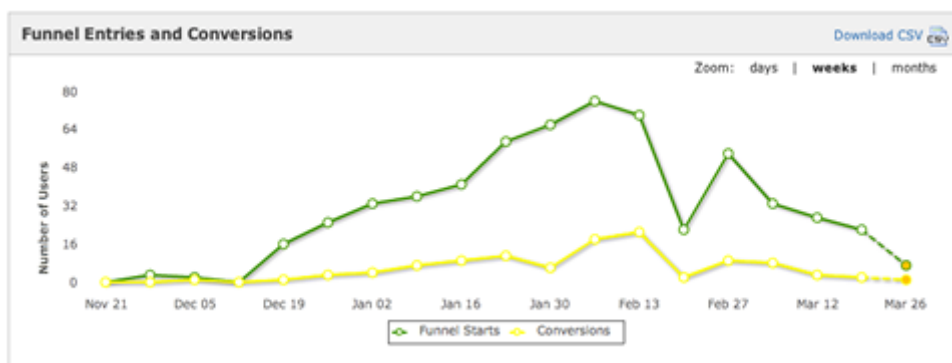
Funneloinnilla voidaan vastata kysymyksiin ”Kuinka moni tutoriaalın katsonneista pääsi lävitse tasot yksi, kaksi ja kolme, kolmessa päivässä”, ”Kuinka moni käyttäjä tekee sovelluksen sisäisen oston yhden tunnin, päivän tai viikon sisällä ensimmäisen sovelluksen käynnistämisen jälkeen”. Funnelointia on kahden tyyppistä Flurryn Analyticsissä, sovelluksen sisäistä ja monen sovelluksen välistä. (Flurry n.d-f.)

Uuden funnelin luominen sisältää parin askeleen konfigurointia, jokaiseen yksittäiseen tapahtumaan, jotka on sovelluksessa määritetty. Järjestys jokaisen askeleen kohdalla funneloinnissa on tärkeää, sillä jokainen funneli on laskettu tiettyyn järjestykseen, missä ne ovat konfiguroidut. Luotuaan funnelin, kestää noin 24 tuntia saada tietoa näistä. Flurry Analyticsin funneloinnissa on laskelmoitu koko sovellushistorian ja eikä ainoastaan tapahtumista, jotka syntyvät funnelin rakentamisen jälkeen. Etuna tässä ratkaisussa on, että niitä voi rakentaa jokaisen erilaisen kysymyksen noustessa sovelluskehittäjän mieleen. Funnel Vizuliazation sivu antaa näkymän funneleista. Askeleet jotka määrittävät funnelin, käyttäjien määrä sekä prosenttimäärä kuinka moni suoritti tietyn askeleen (Kuva 7). (Flurry n.d-f.)

Step	Users	Conversion %
Choose Car	592	
Check In	227	38.9%
View Check In	152	66.9%
View Stats	106	69.7%
		17.91%

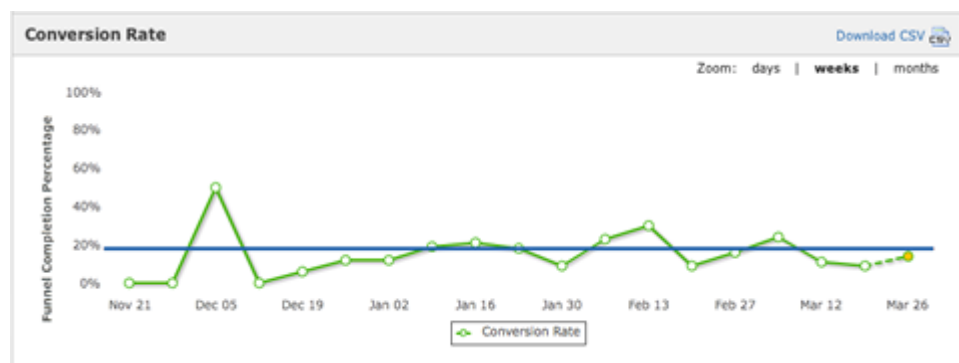
Kuva 7. Askeleet. (Flurry n.d-f.)

Kaavio, joka näyttää kuinka moni käyttäjä aloittivat askeleen tietyn ajanjakson aikana ja kuinka moni heistä lopulta suoritti viimeisen askeleen (Kuva 8). (Flurry n.d-f.)



Kuva 8. Kaavio. (Flurry n.d-f.)

Kaavio, joka näyttää muunnoksen määrän eli conversion raten. Nähdään kuinka iso prosentti käyttäjistä, jotka suorittivat viimeisen askeleen, jotka aloittivat ensimmäisen askeleen (Kuva 9). (Flurry n.d-f.)



Kuva 9. Kaavio muunnoksen määrästä. (Flurry n.d-f.)

Segmentit, versiot ja aika suodattimet ovat saatavilla funneleihin. Näillä suodattimille voidaan kehittää keskitystä funnelista ja vertailla muunnoksen määrää erilaisissa tilanteissa. Esimerkiksi käyttäen version suoda-

tinta, voidaan nähdä muutos ”Onko 4.6.15 versiota edeltäneet tutoriaaliin tarkoitetut parannukset vaikuttaneet käyttäjien suoritukseen ensimmäisellä ja toisella tasolla pelissä”. (Flurry n.d-f.)

6.3 Käyttöönotto

Flurryn käyttöönotossa on otettu heidän oma ohjeensa. Käyttöönotto alkaa Flurry Analyticsiin rekisteröitymisellä ja käyttäjätunnuksen aktivoimisella. Valitaan mihin käyttöjärjestelmään sovellus tulee, mihinkä kategoriaan sovellus kuuluu ja lopuksi nimetään (Kuva 10). Tämän vaiheen jälkeen saadaan oma uniikki API avain, joka lisätään Android Studiossa sovellukseen, jotta saadaan käyttöönotettua Flurry SDK. (Flurry n.d-c.)

Kuva 10. Havainto sovelluksen nimeämisestä ja sovelluksen kategoriatyypistä.

Flurryllä on kaksi tapaa, miten saada SDK omaan sovellukseen Androidissa. Suositelluin tapa on ladata jCenterin kautta tai jossain tapauksissa .jar-tiedostoina eli javan arkistotiedostoina, jonka täytyy itse manuaalisesti käydä lisäämässä Android Studiossa sovelluksen projektitiedostoihin. JCenterin kautta ladattaessa täytyy lisätä Build.Gradle tiedostoon koodi, jolla saadaan Flurry integroitua (Koodi 1). (Flurry n.d-d.)

```
dependencies {
    compile 'com.flurry.android:analytics:6.8.0'
}
```

Koodi 1. Tällä saadaan lisättyä integroituminen sovelluksen Build.Gradlen -konfigurointi tiedostoon. (Flurry n.d-e.)

AndroidManifest.xml -tiedosto tarvitsee myös konfigurointia, joka antaa tietoa Android järjestelmälle, mitä tarvitsee ennen kuin voi suorittaa mitään sovelluksen koodia. Muokattuna koodissa on lupa antaa tietojen menemisen Flurryn servereille. Koodissa on liitetty Flurry Analyticsille sekä Flurry Advertisingille (Koodi 2). (Flurry n.d-d.)

```

<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.flurry.sample"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />

    <application
        android:name=".esimerkkinimi "
        android:icon="@drawable/app_icon">

        </application>
</manifest>

```

Koodi 2. Esimerkki AndroidManifest.xml -tiedoston konfiguroinnista. (Flurry n.d-d.)

Saadaan kutsuttua Flurry SDK:n asennus käyttämällä FlurryAgent.Builderia asennuksen aloittamiseen ja täytyy sisällyttää sovelluksen oma API avain koodissa, jonka saa tehdessään sovelluksen Flurryn sivuilla (Koodi 3). (Flurry n.d-d.)

```

public class Sovellus extends Application {
    @Override
    public void onCreate() {
        super.onCreate();

        new FlurryAgent.Builder()
            .withLogEnabled(true)
            .build(this, "FLURRY_API_AVAIN");
    }
}

```

Koodi 3. Kutsutaan Flurryn SDK asennusta. (Flurry n.d-d.)

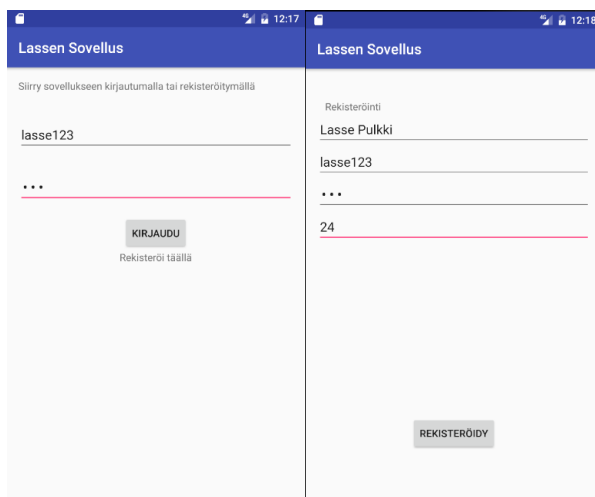
Tämän jälkeen voidaan testata Flurryn toimivuutta sovelluksessa ja katsoa syntyykö virheitä tai muuta sitä vastaavaa. Mikäli kaikki toimii kuten pitää, tapahtuma tallennetaan Flurryyn. (Flurry n.d-d.)

7 FLURRY ANDROID SOVELLUKSESSA

Tämän opinnäytetyön käytännönoosuus tehdään Android Studiolla tehdyllä sovelluksella, johon liitetään Flurry. Flurryn avulla seurataan sovelluksen käyttöä ja niiden pohjalta laaditaan käyttötapaus eli use case.

7.1 Sovellus

Sovellus on yksinkertaisesti tehty eventtien seuranta tarkkaileva sovellus. Osa applikaatiosta on tehty YouTube videotutoriaalin avulla. (ToniKammiTV 2016.) Sovellus sisältää rekisteröitymisen, joka on toteutettu Volley HTTP-kirjastolla, PHP-ohjelmistokielellä ja MySQL -relaatiotietokannalla. Rekisteröitymisessä kysytään koko nimeä, käyttäjätunnusta, salasanaa sekä käyttäjän ikää (Kuva 11). Rekisteröitymistiedot tallennetaan tietokantaan ja sieltä haetaan tietoa kirjautumista varten (Kuva 12). Sovelluksen koodiin on lisätty Flurry, jonka käyttöönotto näkyy tämän opinnäytetyön luvussa 6.



Kuva 11. Sovellukseen kirjautuminen ja rekisteröityminen.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	kayttaja_id	int(11)			No	None		AUTO_INCREMENT
2	nimi	varchar(16)	utf8_unicode_ci		No	None		
3	kayttajatunnus	varchar(16)	utf8_unicode_ci		No	None		
4	ika	tinyint(4)			No	None		
5	salasana	varchar(16)	utf8_unicode_ci		No	None		
6	tunnit	int(11)			No	None		

Kuva 12. MySQL näkymä.

MySQL ja PHP -tiedostojen verkkopalveluna käytin ilmaista 000webhostia. Loin tietokannan johon lisäsin käyttäjätunnisteen, nimen, salasanan, käyttäjän iän ja käyttäjänimen. Loin kaksi PHP-tiedostoa, joissa määritellään kirjautuminen ja käyttäjän luonti tietokantaan (Koodi 4).

```
<?php
    $con = mysqli_connect("localhost", "id816599_ikt", "salasana",
        "id816599_lassensovellus");

    $nimi = $_POST["nimi"];
    $ika = $_POST["ika"];
    $kayttajatunnus = $_POST["kayttajatunnus"];
    $salasana = $_POST["salasana"];

    $statement = mysqli_prepare($con, "INSERT INTO kayttaja (nimi,
        kayttajatunnus, ika, salasana) VALUES (?, ?, ?, ?)");
    mysqli_stmt_bind_param($statement, "ssis", $nimi, $kayttajatunnus,
        $ika, $salasana);
    mysqli_stmt_execute($statement);

    $response = array();
    $response["success"] = true;

    echo json_encode($response);
?>
```

Koodi 4. Esimerkkinä rekisteröinnin PHP-tiedosto. Sovelluksesta tallennetaan syötetyt tiedot tietokantaan, josta salasanaa ja käyttäjätunnusta voidaan todeta kirjautuessa.

Esimerkkinä rekisteröinnin PHP-tiedoston liittäminen koodiin tarvitsee luoda oma pyyntö luokka, missä se hakee verkkopalvelun osoitteesta PHP-tiedoston (Koodi 5). Lopuksi tämä liitetään rekisteröinti luokkaan, missä kaikki rekisteröintiin liittyvät asiat on määritetty (Liite 1).

```
public class RekisterointiPyynto extends StringRequest {

    private static final String REGISTER_REQUEST_URL = "http://univalent-
        goal.000webhostapp.com/Register.php";
    private Map<String, String> params;

    public RekisterointiPyynto(String nimi, String kayttajatunnus, int ika,
        String salasana, Response.Listener<String> listener) {
        super(Method.POST, REGISTER_REQUEST_URL, listener, null);
        params = new HashMap<>();
        params.put("nimi", nimi);
        params.put("kayttajatunnus", kayttajatunnus);
        params.put("salasana", salasana);
    }
}
```

```

        params.put("ika", ika + "");
    }

    public Map<String, String> getParams() {

        return params;

    }
}

```

Koodi 5. Rekisteröinnin pyyntö.

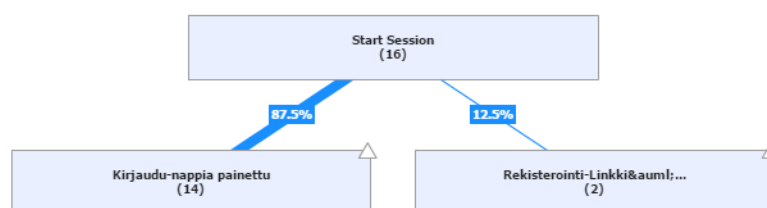
7.2 Käyttötapaus

Sovellusta käytetään Android studion avulla, johon on tehty Huaweiin Nexus 6 matkapuhelin, mikä käyttää Androidin Nougat 7.1 käyttöjärjestelmän versiota. Flurryn kanssa päädyin tekemään käyttötapauksen, jossa seurataan kaikkien nappien, sivujen ja linkkien tapahtumien seuranta. Jokaiseen nappiin ja linkkiin on liitetty koodi (Koodi 6). Loin muutaman käyttäjän sovellukseen ja testailin heillä erilaisia tehtäviä. Käyttötapauksessa seurataan mitä näkyy kojelaudalla, kun sovellusta on käytetty muutama päivä.

```
FlurryAgent.logEvent("Rekisteröinti-nappia painettu");
```

Koodi 6. Esimerkki koodista. Tässä tapauksessa seurataan rekisteröinti sivun painiketta, johon koodi on liitetty.

Sovelluksen käyttäjäpolkuun on automaattisesti tullut näkymä, kuinka monta kertaa kirjautumisen painiketta on painettu sekä rekisteröitymisen linkkiä (Kuva 13).



Kuva 13. Käyttäjäpolku.

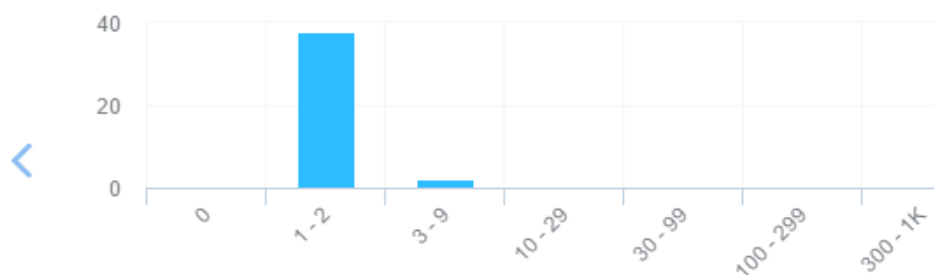
Seurattuna on myös mitä painiketta tai linkkejä on painettu, samalla seuraten päivittäisiä keskiarvoja eventille, joissa huomataan kirjautumisen painikkeen olevan painettuna eniten (Kuva 14).

Event Summary Statistics				
Event Name	Total Event Occurrences	Event Occurrences (Daily Avg)	Unique Event Users (Daily Avg)	Events per Session (Daily Avg)
Tunteja Lisätty	1	0.03	0.03	0.06
Kirjaututtu ulos-nappia painettu	5	0.16	0.03	0.31
Tutoriaali katsottu-nappia painettu	1	0.03	0.03	0.06
Kirjautu-nappia painettu	22	0.69	0.09	1.38
Rekisterointi-Linkkiä painettu	8	0.25	0.09	0.50
Lisää urheilutunteja pääsivulta-nappia painettu	3	0.09	0.03	0.19
Rekisteröinti-sivun eventti	6	0.19	0.09	0.38
Rekisteröinti-nappia painettu	8	0.25	0.09	0.50
Pääsivun-sivun eventti	15	0.47	0.09	0.94
Tutoriaali-sivun eventti	1	0.03	0.03	0.06
Tutoriaali-linkkiä painettu	1	0.03	0.03	0.06

Kuva 14. Kuinka monta kertaa eventit ovat lauenneet kokonaisuudessaan ja niiden keskiarvot.

Eventtien seurannassa nähdään, miten istunnon aikana on laukaistu eventtejä. Näissä istunnoissa nähdään 1-2 eventtien laukaisun olleen yleisempää kuin 3-9 laukaisua (Kuva 15).

Event Details Dashboard / Events Per Session



Kuva 15. Kuinka monta eventtiä on laukaistu per istunto.

Kaikki asiat esiintyvät automaattisesti kojelaudalla, kun kaikki on oikein tehty sovelluksessa. Näillä saa helposti eventeistä tietoa, mitä sovelluksessa laukaistaan useasti ja mitä ei laukaista usein. Tämä antaa heti tietoa, mitä asiaa voidaan parantaa, jotta moni käyttäjä suorittaisi askeleet.

Sekä tieto, kuinka monta eventtiä tapahtuu tietyn istunnon aikana, joka auttaa ymmärtämään, kuinka monta kertaa tietyt tapahtumat tapahtuvat.

8 YHTEENVETO

Opinnäytetyön aihe oli itselleni aivan tuntematon sekä haastava. Ohjelmointia en ole opetellut tai käyttänyt ensimmäisen lukuvuoden jälkeen ammattikorkeakoulussa ja se näkyi aika vahvasti tehdessäni työtä. Opinnäytetyöni teorian puoleinen osa onnistui mielestäni hyvin ja siitä sain melko laajasti kasaan tietoa sekä pystyin tiivistämään sen hyvän kokoiseksi paketiksi. Mobiilianalytiikka teorian osalta oli mielenkiintoista luettavaa sekä ymmärsin melkein heti, mitä niissä ajettiin takaa sovelluskehittäjien näkökulmasta. Kanssa opin tuntemaan itselleni tuntemattoman aihealueen teoriassa ja kasaamaan siitä opinnäytetyön. Lähteitä löytyi monta linkkiä ja sain paljon tietoa internetistä sekä E-kirjoista, mutta fyysistä kirjaa en saanut käsiini ollenkaan tähän opinnäytetyöhön, mikä olisi ollut erittäin hyvä lisäys lähdeluetteloon.

Käytäntö ei onnistunut sillä tavalla, mitä oletin eikä mikä oli opinnäytetyössäni tarkoitus. Opinnäytetyöhön annettu aika oli omasta mielestäni aika tiukka, vaikka päädyin yksinkertaiseen sovellukseen, niin se tuotti paljon ongelmia ja näin ollen ongelmakohtien ratkaisu koodissa vei itseltäni paljon aikaa. Se taas viivästytti kokonaisuuden kasaamista. Aluksi sovellustani aloittaessa etsin ohjetta, jonka pohjalta aloitin tekemään yksinkertaisen sovelluksen. Sovellus sisälsi kirjautumisen ja rekisteröinnin tietokantaan sekä sovelsin muita asioita siihen. Kaikki asiat tosin ei sovelluksessani toimineet, kuten pitäisi ja eivät tulleet käyttöön opinnäytetyössäni. Käytäntöön vielä piti tulla enemmän käyttäjien seurantaa, mutta siihen asti en kerennyt tekemään. Kanssa piti luoda tehtäviä oppilaille ja opetusmateriaali liitteenä, mutta aika otti siinäkin vastaan Työtä olisi voinut jatkaa enemmänkin, varsinkin käytännön puolelta, jossa aika-tila ja osaamistaitoni otti vastaan. Toisaalta teorian puolelta myös olisi pitänyt rajata vielä enemmän aiheita ja näin ollen lopulta käytäntö jäi suppeaksi. Käytin aikaa viimeisen viikon ajan kokonaan käytäntöön ennen opponointia ja väliseminaaria sekä kirjoitusasun korjaamiseen, jota oli syönyt liiallinen aika teoriaosuuden kasaamiseen. Hiihtoloma viikon aikana korjailin kirjoitusvirheitä sekä lähdeviittauksia.

Hyviä puolia mitä opin opinnäytetyön aikana, vastapainoksi koodaamistaitojeni puutteisiin oli se, että opettelin käyttämään Android Studiota, jota en ollut aiemmin käyttänyt ja myös en aiemmin ollut tehnyt mobiilisovellusta. Lopulta olen itse tyytyväinen opinnäytetyöhöni teorian osalta ja hieman pettynyt käytäntöön.

LÄHTEET

- Android. (n.d). *Create an Android Project*. Viitattu 20.2.2017. <https://developer.android.com/training/basics/firstapp/creating-project.html>
- Amplitude. (n.d). *Mobile Analytics: A Complete Guide to App Retention and Engagement*. Viitattu 2.2.2017. <https://amplitude.com/mobile-analytics>
- Appcelerator. (n.d). *Stop Guessing, Start Measuring: 5 Mobile Metrics for Great Apps*. E-kirja. Viitattu 26.1.2017. <http://appcelerator.uberflip.com/i/272485-stop-guessing-start-measuring-5-mobile-metrics-for-great-apps>
- AppDynamics. (2015). *16 metrics to ensure mobile app success*. <https://www.appdynamics.com/media/uploaded-files/1432066155/white-paper-16-metrics-every-mobile-team-should-monitor.pdf> Viitattu 4.2.2017
- Appsee. (n.d). *The Complete KPI Handbook for Mobile Apps: Measuring the Right metrics for Each Phase of Your App*. E-Kirja. Viitattu 26.1.2017
<https://www.appsee.com/ebooks/mobile-apps-kpi-handbook>
- App47. (2014). *Want to Measure the ROI of Your Enterprise Mobile App? Take a Good, Hard Look at Analytics*. Viitattu 10.2.2017
<http://www.app47.com/2014/03/06/want-to-measure-the-roi-of-your-enterprise-mobile-app-take-a-good-hard-look-at-analytics/>
- Beer, W. (2016). *Mobile App Analytics*. E-kirja. Viitattu 2.2.2017
<https://assets.dynatrace.com/content/dam/en/cs/Mobile-App-Analytics-Dynatrace-Ruxit.pdf>
- Biterati. (2016). *Game Analytics Offers "Inside Look" Into Gamer Habits*. Kirjoitus blogissa 23.2.2016. Viitattu 20.2.2017. <http://biterati.com/2016/02/game-analytics-offers-inside-look-into-gamer-habits/>
- Bourne, J. (2016). *Majority of firms using KPIs for mobile seeing positive benefits*. Kirjoitus verkkouutisessa 13.1.2016. Viitattu 10.2.2017. <http://www.appstechnews.com/news/2016/jan/13/majority-firms-using-kpis-mobile-seeing-positive-benefits/>

Luk, G. & Brown, A. (2016.) *The Mobile Enterprise Business Applications Market will Grow to \$128.2 billion by 2022*. Kirjoitus lehdistötiedotteessa 8.11.2016. Viitattu 10.2.2017. [https://www.strategyanalytics.com/strategy-analytics/news/strategy-analytics-press-releases/strategy-analytics-press-release/2016/11/08/the-mobile-enterprise-business-applications-market-will-grow-to-\\$128.2-billion-by-2022](https://www.strategyanalytics.com/strategy-analytics/news/strategy-analytics-press-releases/strategy-analytics-press-release/2016/11/08/the-mobile-enterprise-business-applications-market-will-grow-to-$128.2-billion-by-2022)

Darden, M. (2016). *Seven benefits of mobile analytics services for app developers*. Kirjoitus blogissa 3.6.2016. Viitattu 19.01.2017. <http://mobilebusinessinsights.com/2016/06/seven-benefits-of-mobile-analytics-services-for-app-developers/>

Drachen, A. (2013). *Balance and Flow Maps*. Kirjoitus blogissa 30.9.2013. Viitattu 3.2.2017. <http://www.gameanalytics.com/blog/balance-and-flow-maps.html>

Drell, L. (2013). *9 Mobile App KPIs to Know*. Kirjoitus blogissa 5.9.2013. Viitattu 4.2.2017. <http://mashable.com/2013/09/04/mobile-app-metrics/#t38jF83S.mqP>

EvokedTechnologies. (2015). *Getting the Most Out of Mobile App Analytics*. Viitattu 10.2.2017. <http://www.evokedtechnologies.com/blog/getting-most-of-mobile-app-analytics/>

Favoritemedium. (n.d). *Trello – Delivering Praise and Delight on Android*. Viitattu 10.2.2017. <http://www.favoritemedium.com/case-study/trello-android/>

Flurry (n.d-a). *Real-Time Metrics*. Viitattu 8.2.2017. <https://developer.yahoo.com/flurry/docs/analytics/lexicon/realtimemetrics/>

Flurry. (n.d-b). *Measures*. Viitattu 8.2.2017. <https://developer.yahoo.com/flurry/docs/analytics/explorer/measures/>

Flurry. (n.d-c). *Flurry Analytics*. Viitattu 19.01.2017. <https://developer.yahoo.com/flurry/docs/analytics/>

Flurry. (n.d-d). *Get Started With Flurry Analytics for Android*. Viitattu 2.2.2017. <https://developer.yahoo.com/flurry/docs/analytics/getting-started/android/>

Flurry. (n.d-e.). *Events for Flurry Analytics*. Viitattu 19.2.2017 <https://developer.yahoo.com/flurry/docs/analytics/lexicon/eventreporting/>

Flurry. (n.d-f.). *Funnel Analysis*. Viitattu 20.2.2017 <https://developer.yahoo.com/flurry/docs/analytics/lexicon/funnelanalysis/>

- Gauchet S. (n.d) Learn How People Use Your App – An App Analytics Tools Round-Up. Kirjoitus blogissa. Viitattu 10.2.2017. <https://www.apptamin.com/blog/app-analytics-tools/>
- Google. (n.d). *Google Analytics SDK for Android v3 (Legacy) – Getting Started*. Viitattu 15.2.2017. <https://developers.google.com/analytics/devguides/collection/android/v3/>
- Harty, J. & Aymer, A. (2016) *The Mobile Analytics Playbook – A Practical Guide to Better Mobile Testing*. E-kirja. Viitattu 19.1.2017. <https://www.hpe.com/h20195/V2/GetPDF.aspx/4AA6-4189ENW.pdf>
- Hiltunen, K., Latvala, S. & Kaleva, J-P. (2013) *Peliteollisuus – kehityspolku*. E-kirja. https://www.tekes.fi/globalassets/julkaisut/peliteollisuus_kehityspolku.pdf Viitattu 9.2.2017
- HockeyApp. (n.d). Viitattu. 19.01.2017. <https://www.hockeyapp.net/>
- Kwakyi, G. (2016). *Which KPI Should I Use For My Mobile App Marketing Plan*. Kirjoitus blogissa 24.1.2016. Viitattu 4.2.2017. <https://growth-bug.com/which-kpi-should-i-use-for-my-mobile-app-marketing-plan-a6408021cc4c#.b7tlzuxn4>
- Lunden, I. (2014). *Yahoo Buys Mobile Analytics Firm Flurry For North Of \$200M*. Kirjoitus 21.7.2014. Viitattu 5.2.2017. <https://techcrunch.com/2014/07/21/yahoo-is-buying-mobile-analytics-firm-flurry-for-north-of-200m/>
- Localytics. (n.d). *A Beginner’s Guide to App Analytics*. E-kirja. Viitattu 26.1.2017. http://pages.localytics.com/rs/650-CZQ-842/images/A_Beginners_Guide_to_App_Analytics_.pdf
- McCobb J. (2012). *Know your customers by applying Mobile Analytics to your app*. Kirjoitus blogissa 8.11.2012. Viitattu 2.2.2017. <http://www.techrepublic.com/blog/software-engineer/know-your-customers-by-applying-mobile-analytics-to-your-app/>
- Newzoo. (2016). *The Global Games Market Reaches \$99.6 Billion In 2016, Mobile Generating 37%*. Kirjoitus artikkelissa 21.4.2016. Viitattu 8.2.2017 <https://newzoo.com/insights/articles/global-games-market-reaches-99-6-billion-2016-mobile-generating-37/>
- Orange Analytics. (n.d). *Orange Analytics Development Process*. Viitattu 10.2.2017. <http://www.orangeanalytics.com.au/blog/orange-analytics-mobile-app-development-process>

Popp, M. (2015). *How To Measure Success From An Enterprise Mobile App*. Viitattu 11.2.2017 <https://chaione.com/blog/measure-success-enterprise-mobile-app/>

PocketMonsters. (n.d.). <https://bitbucket.org/vinyluntu/pocketmonsterscrossplatform>

Sharma, R. (2016). *How to Calculate Success of Your Enterprise Mobile Application*. Kirjoitus artikkelissa 5.4.2016. Viitattu 11.2.2017. <http://sandhill.com/article/how-to-calculate-success-of-your-enterprise-mobile-application/>

Statista. (n.d-a). *Number of apps available in leading app stores as of june 2016* .Viitattu 1.2.2017 <https://www.statista.com/statistics/276623/number-of-apps-available-in-leading-app-stores/>

Statista. (n.d-b). *Number of mobile app downloads worldwide from 2009 to 2017 (in millions)*. Viitattu 26.1.2017 <https://www.statista.com/statistics/266488/forecast-of-mobile-app-downloads/>

Thalamus. (2015) *Genesis Stories: An Interview with Flurry Founder Sean Byrnes*. Kirjoitus blogissa 30.7.2015. Viitattu 5.2.2017. <http://www.thalamus.co/blog/flurry-founder-sean-byrnes/>

ToniKamiTV. (2016) *New Login & Register Android App Tutorial*. Viitattu 19.2.2017. <https://www.youtube.com/playlist?list=PLE60o7ed8E-TztoF2K3y4VdDgT6APZ0ka>

Walz, A. (2015a) *The Data Behind Customer Acquisition and Retention For F2P Mobile Games*. Kirjoitus blogissa 9.4.2015. Viitattu 8.2.2017. <https://www.apptentive.com/blog/2015/04/09/the-data-behind-customer-acquisition-and-retention-for-f2p-mobile-games/>

Walz, A. (2015b). *The Five Mobile Marketing Metrics That Matter Most*. Kirjoitus blogissa 7.4. 2015. Viitattu 9.2.2017. <https://www.apptentive.com/blog/2015/04/07/the-five-mobile-marketing-metrics-that-matter-most/>

[x]Cube LABS. (2016) *How TO Measure The Success Of Enterprise Mobile Apps*. Kirjoitus blogissa 7.7.2016. Viitattu 11.2.2017. <https://www.xcubelabs.com/our-blog/enterprise-mobility/measure-success-enterprise-mobile-apps/>

Rekisteröinti luokan koodi.

```

package com.example.lassensovellus.lassensovellus;

import android.content.Intent;
import android.support.v7.app.AlertDialog;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;

import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.toolbox.Volley;
import com.flurry.android.FlurryAgent;

import org.json.JSONException;
import org.json.JSONObject;

public class Rekisterointi extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_rekisterointi);

        final EditText lsIka = (EditText) findViewById(R.id.lsIka);
        final EditText lsKokoNimi= (EditText) findViewById(R.id.lsKokoNimi);
        final EditText lsKayttajaTunnus= (EditText) findViewById(R.id.lsKayttajaTunnus);
        final EditText lsSalasana= (EditText) findViewById(R.id.lsSalasana);
        final Button lsRekBtn = (Button) findViewById(R.id.lsRekBtn);

        FlurryAgent.logEvent("Rekisteröinti-sivun eventti");

        lsRekBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                FlurryAgent.logEvent("Rekisteröinti-nappia painettu");

                final String nimi = lsKokoNimi.getText().toString();
                final String kayttajatunnus = lsKayttajaTunnus.getText().toString();
                final String salasana = lsSalasana.getText().toString();
                final int ika = Integer.parseInt(lsIka.getText().toString());
            }
        });
    }
}

```

```

    FlurryAgent.setAge();

    Response.Listener<String> responseListener = new Response.Listener<String>(){
        @Override
        public void onResponse(String response) {
            try {
                JSONObject jsonResponse = new JSONObject(response);
                boolean success = jsonResponse.getBoolean("success");
                if (success){
                    Intent intent = new Intent(Rekisterointi.this, Etusivu.class);
                    Rekisterointi.this.startActivity(intent);
                }else {
                    AlertDialog.Builder builder = new AlertDialog.Builder(Rekisterointi.this);

                    builder.setMessage("Rekisteröinti epäonnistui")
                        .setNegativeButton("Yritä uudelleen", null)
                        .create()
                        .show();
                }

            } catch (JSONException e) {
                e.printStackTrace();
            }
        }
    };

    RekisterointiPyynto rekisterointiPyynto = new RekisterointiPyynto(nimi,
    kayttajatunnus, ika, salasana, responseListener );
    RequestQueue queue = Volley.newRequestQueue(Rekisterointi.this);
    queue.add(rekisterointiPyynto);

}
});

}
}

```