

MOBIILISOVELLUS JÄÄKIEKON PELIVIRRRAN SEURANTAAN



Ammattikorkeakoulututkinnon opinnäytetyö

Visamäki, Tietojenkäsittely

hyväksymislukukausi, 2016

Juuso Saarinen

Tietojenkäsittely
Visamäki

Tekijä Juuso Saarinen **Vuosi** 2016

Työn nimi Mobiilisovellus jääkiekon pelivirran seurantaan

TIIVISTELMÄ

Opinnäytetyön tavoitteena oli luoda nopea ja helposti ymmärrettävä käyttöliittymä jääkiekon pelivirran seurantaan tarkoitettuun sovellukseen. Tämä tehtiin Xamarin-sovelluskehitysalustalla Android-käyttöjärjestelmälle. Käyttöliittymällä piti olla mahdollista tallentaa oman joukkueen pelitapahtumat ja ketjujen kokoonpanot sekä vastustajan pelistä sen pelitapahtumat pienemmällä tarkkuudella.

Tarve sovellukselle oli jääkiekkovaikuttajien alkuperäinen tapa kerätä tietoa pelistä Excelin avulla pelien jälkeen. Sovelluksella sama tehtävä hoituu pelin aikana, ja tiedon analysointi tulee tehtyä nopeammin. Toimeksiantajana toimi Hämeen ammattikorkeakoulun Älykkäät palvelut -tutkimusyksikkö, jolle projekti oli tullut kahdelta paikalliselta jääkiekkovaikuttajalta.

Opinnäytetyössä avuksi olivat aiemmat ohjelmointitaidot ja käyttöliittymäsuunnitteluun keskittyneet kurssit opintojen aikana. Sen lisäksi tekijä oli lukenut käyttöliittymäsuunnitteluun liittyvää kirjallisuutta. Androidiin liittyen tekijä oli lukenut ohjelmointioppaita ja tehnyt hieman omia kokeiluita, mutta muuten Androidissa oli paljon uutta.

Opinnäytetyö on jaettu kahteen osaan. Ensimmäinen osa koostuu teoriaosasta, jossa käsitellään Androidia, Xamarin-sovelluskehitysalustaa ja käyttöliittymäsuunnittelua ja tiedon keräämiseen liittyviä tapoja Androidissa. Toisessa osassa käsitellään sovelluksen näkökulmasta mahdollisia tiedonkeräämisen ja esittämisen tapoja sekä toimeksiantajan omia tavoitteita sovellukseen ja sen käyttöliittymään.

Opinnäytetyön lopputuloksena on sovellukseen sopivan käyttöliittymän esittäminen sekä sen käyttöliittymään kuuluvien toimintojen esittelyä.

Avainsanat Android, Xamarin, Mobiilisovellukset, Käyttöliittymä

Sivut 28 sivua

Business Information Technology

Visamäki

Author Juuso Saarinen **Year** 2016

Subject Mobile application for following hockey game flow

ABSTRACT

The goal of the thesis was to create a fast and easily understandable user interface for an application to be used in following the game flow of a hockey game. This was done with the Xamarin mobile application development platform for the Android operating system. The user interface would make it possible to save the detailed game events of the user's own team as well as the game events of the opponent's team but with less accuracy.

The need for the application comes from the original way of collecting data by watching the match from video after the game and saving the events to an Excel sheet. With the application, the same can be done more quickly during the game. The thesis was commissioned by the Smart Services Research Unit at Häme University of Applied Sciences and the project was originally given by two local influential hockey persons.

The previous programming skills and courses focusing on interface design helped working on the thesis. Besides that, literature about interface design was used as reference material. For Android, programming books and the author's own projects helped the process which gave a lot of new information.

The thesis is divided into two parts. The first part deals with the theory on Android, Xamarin mobile development platform and user interface. In the second part, the thesis discusses the collecting and displaying of the data from the application's perspective and the client's own goals and the state of the application and its user interface in the end of the thesis.

The product of the thesis presents a fitting user interface for the application and introduces the parts of the user interface.

Keywords Android, Xamarin, Mobile applications, User Interface

Pages 28 pages

SISÄLLYS

1	JOHDANTO.....	1
2	ANDROID.....	3
2.1	Versiohistoria	3
2.2	Androidin käyttöliittymä	3
2.3	Resurssit	4
3	XAMARIN	5
3.1	Ohjelmointiympäristö	5
3.2	Käyttöliittymätyökalut Xamarinissa	5
4	KÄYTTÖLIITTYMÄSUUNNITTELUN HUOMIOITA	7
5	TIEDONKERÄÄMISEN TAPOJA	8
5.1	Erilaiset toiminnot.....	8
5.2	Eleiden käyttäminen.....	8
6	TOIMEKSIANTAJAN TAVOITTEET SOVELLUKSEEN JA KÄYTTÖLIITTYMÄÄN.....	10
6.1	Alkuperäinen tapa kerätä tietoa	10
6.2	Sovelluksen tavoite	10
6.3	Toiminnallisuuden huomioon ottaminen	10
7	SOVELLUKSEEN SOPIVAT TOIMINNOT JA ELEET.....	12
7.1	Tiedon keräämiseen sopivat toiminnot ja eleet	12
7.2	Tiedon esittämiseen sopivia toimintoja ja eleitä	14
8	KÄYTTÖLIITTYMÄN TOTEUTUS	15
8.1	Käyttöliittymän ensimmäinen versio	15
8.2	Käyttöliittymän toinen versio.....	18
8.3	Käyttöliittymän kolmas versio.....	21
9	SOVELLUKSEN LOPPUTILANNE JA JATKOKEHITYS	24
9.1	Tiedon kerääminen	24
9.2	Tiedon esittäminen	24
9.3	Jatkokehitys.....	25
10	YHTEENVETO	26
	LÄHTEET	27

1 JOHDANTO

Opinnäytetyön aihe tuli Hämeen ammattikorkeakoulun Älykkäät palvelut - tutkimusyksiköltä, joka puolestaan sai aiheen kahdelta paikalliselta jääkiekkovaikuttajalta, jotka toimivat asian parissa yksityishenkilöinä. Opinnäytetyötä teen sekä omalla ajalla että samalla kun olen työharjoittelussa Älykkäät palvelut -tutkimusyksikössä. Aiheena on mobiilisovellus jääkiekon pelivirran seurantaan Androidille, ja opinnäytetyössä keskityn sovelluksen käyttöliittymään. Sovelluksella itsellään on tarkoitus kerätä pelin aikana tietoa pelitapahtumista ja pystyä esittämään oleellista tietoa jo pelin aikana, mutta myös jälkepäin.

Opinnäytetyön tarkoituksena on selvittää mitä toimintoja ja ominaisuuksia Androidista löytyy, ja miten niitä voidaan käyttää tiedon tehokkaaseen keräämiseen ja esittämiseen sekä miten niillä voidaan luoda sopiva käyttöliittymä. Käytän sovelluksen ja käyttöliittymän tekoon Xamarin-sovelluskehitysalustan Android-sovelluskehitystyökalua. Xamarin sisältää Designer työkalun, jonka avulla käyttöliittymän teko on nopeaa ja sen avulla näkee muutokset ulkoasussa välittömästi.

Aiempi kokemus Androidista on lyhyesti aiemmilla kursseilla käydyt asiat ja sen jälkeen itsenäisesti opiskellut asiat. Kurssien aikana Androidia käsiteltiin Java-ohjelmointikielellä, kun taas opinnäytetyössä käytetty Xamarin käyttää C#-ohjelmointikieltä kehitykseen. C#-ohjelmointikieltä on opiskelujen aikana käytetty laajimmin, joten se on paremmin itsellä hallussa. Käyttöliittymäsuunnittelustakin on ollut opiskelujen aikana kurssi, mutta enemmän asiasta olen oppinut lukemalla teoriaa ja käyttämällä lukemattomia sovelluksia nähden hyviä ja huonoja ratkaisuja eri sovellusten välillä. Tulevaisuudessa itseäni kiinnostaa erityisesti mobiiliohjelmointi, mutta koska Xamarinissa käytetään C#-kieltä, pystyn käyttämään työssä oppimaani muutenkin kuin vain mobiiliohjelmoinnissa. Työn käyttöliittymäosuudesta on myös hyötyä jatkossa muillakin alustoilla kuin vain mobiilissa, vaikkakin esimerkiksi työpöytäsovellukset mahdollistavat laajemman työskentelyalan.

Opinnäytetyön teoriaosuudessa käydään läpi käyttöjärjestelmänä toimiva Android, josta käydään läpi yleistä asiaa, mutta lisäksi käyttöliittymään kuuluvia asioita. Sen jälkeen käydään läpi Xamarin-sovelluskehitysalusta, ja miten sitä voidaan käyttää sovelluksen kehittämisen lisäksi käyttöliittymän luomiseen. Tämän osuuden jälkeen käydään läpi yleistä käyttöliittymäsuunnittelusta ja Androidissa käytettävissä olevia tiedonkeräämisen tapoja.

Opinnäytetyö vastaa seuraaviin kysymyksiin:

1. Millainen on hyvä käyttöliittymä tavoiteltuun sovellukseen?
2. Mitä erilaisia tapoja on nopeaan tiedon keräämiseen?

2 ANDROID

Tämä luku käsittelee Androidia yleisellä teoriatasolla, eikä käy yksittäistä asiaa kovinkaan yksityiskohtaisesti läpi. Luvussa käydään läpi Androidin versiohistoriaa, käyttöliittymää ja ulkoasuun liittyvää resurssirakennetta.

2.1 Versiohistoria

Android on käyttöjärjestelmänä moneen kertaan muuttunut ja kehittynyt, ja päivityksiä tulee jatkuvalla tahdilla. Ensimmäinen versio julkaistiin syyskuussa 2008 ja tällä hetkellä uusin versionumero 7.0 julkaistiin elokuussa 2016. Versiosta 1.5 alkaen julkaisut on nimetty leivonnaisten mukaan. Isompia päivityksiä on tullut vähintään yksi vuosittain julkaisusta lähtien. (Android 2016a.)

Uudet päivitykset ovat tuoneet mukanaan uusia toimintoja ja parannellut vanhoja. Isoimpien päivitysten välillä näkyvimmat muutokset ovat olleet ulkoasussa, mutta ne ovat myös sisältäneet paljon sisäisiä muutoksia. (Android 2016a.)

2.2 Androidin käyttöliittymä

Androidin käyttöliittymä on ulkoisesti muuttunut huomattavasti alkutilanteesta. Tähän on osasyynä suuret ulkoasumuutokset suurien versiomuutosten välillä. Uudet päivitykset ovat tuoneet huomattavasti lisää toimintoja ja tapoja, joilla tietoa voidaan esittää. Uusien ominaisuuksien lisäksi, vanhoja ominaisuuksia on korvattu ja kehitetty paremmiksi sekä ulkoisesti että koodillisesti. (Android 2016b.)

Näytön koko tarjoaa erilaisia mahdollisuuksia luoda käyttöliittymiä. Esimerkiksi suurille näytöille voidaan tuoda näkyviin enemmän toimintoja yhtä aikaa verrattuna siihen, että pienemmillä näytöillä täytyy osa toiminoista piilottaa alivalikoihin. Erikokoiset näytöt luovat myös haasteensa sovellusten käyttöliittymien kehittämiseen. Android-käyttöjärjestelmä on käytössä usean eri valmistajan laitteessa, ja laitteita voi olla samalla valmistajalla kymmeniä erilaisia varustettuina eri kokoisilla näytöillä. Tämän seurauksena sovelluksen saaminen näyttämään samalta kaikissa laitteissa voi olla haastavaa. Tähän vaikuttaa huomattavasti sovellus itsessään, ja mikä sovelluksen tarkoitus on. Yksinkertainen sovellus, mihin syötetään muutamaan kenttään tietoa, on helpompi tehdä, kuin sovellus missä on useita painikkeita ja kuvia. Helposti käy niin, että sovellus näyttää hyvältä tietyissä laitteissa, mutta joissain laitteissa painikkeet ja kuvat menevät päällekkäin. (Android 2016c.)

2.3 Resurssit

Androidin resurssitiedostot jaetaan kansioihin niiden tyyppin mukaan. Tämä mahdollistaa sovelluksen helpomman ylläpidettävyyden ja esimerkiksi näin saadaan eri kieliversioille ja resoluutioille luotua omat kansiot ja tiedostot. (Taulukko 1 Android 2016d.)

Taulukko 1. Kansiorakenne resurssitiedostoille (Android 2016j.)

Kansio	Resurssin tyyppi
animator/	XML-tiedostot, jotka määrittelevät animaatiot ja niiden ominaisuudet, kuten pituuden.
anim/	XML-tiedostot, jotka määrittelevät tween animaatiot ja niiden ominaisuudet.
color/	XML tiedostot, jotka määrittelevät värit.
drawable/	Bitmap tai XML-tiedostot, joista sitten tulee sovelluksen kuvat.
mipmap/	Kuvat sovelluksen ikonille.
layout/	XML-tiedostot, jotka määrittävät sovelluksen ulkoasun asettelun.
menu/	XML-tiedostot, jotka määrittelevät valikoiden sisällön.
raw/	Muualle sopimattomia tiedostoja.
values/	XML-tiedostot, jotka sisältävät yksinkertaisia arvoja, kuten string ja int.
xml/	XML-tiedostot, kuten esimerkiksi hakuehdotusten käyttöönotto.

Resurssitiedostojen laajempi kansiorakenne, joka sisältää esimerkiksi eri resoluutioiden tiedostot lisää kansioiden määrää. Tämä myös mahdollistaa sen, että sovellus voidaan kehittää monelle eri laitteelle mahdollisimman tarkasti. Kansioiden nimillä voidaan tarkentaa, mitkä laitteet käyttävät mitään tiedostoja. Esimerkiksi kansio `drawable-en-port/` sisältää piirrettävän tiedoston englanninkieliselle, pystyssä olevalle Android-laitteelle. (Android 2016c.)

3 XAMARIN

Tässä luvussa käydään läpi Xamarinia teoriatasolla. Ensin käydään läpi Xamarin ohjelmointiympäristön osalta, ja sen jälkeen enemmän opinnäyte-työhön liittyvien käyttöliittymätyökalujen osalta.

3.1 Ohjelmointiympäristö

Xamarin on Microsoftin omistama mobiilikehitystyökalu. Xamarinia voidaan käyttää iOS-, Android- ja Windows Phone -käyttöjärjestelmien sovellusten kehittämiseen. Sovelluksia voidaan tehdä joko jokaiselle käyttöjärjestelmälle niiden omia ominaisuuksia ja työkaluja käyttäen, tai sitten voidaan luoda yhteinen koodi, joka toimii jokaisella laitteella. Sovelluksen tarkoituksesta riippuen voi olla, että yksi koodi riittää, mutta jos sovelluksesta tulee monimutkaisempi, on hyvä käyttää eri käyttöjärjestelmien omia koodipohjia ja toimintoja. Osa koodista toimii jokaisessa käyttöjärjestelmässä, joten aivan kaikkea ei tarvitse järjestelmäkohtaisesti luoda. (Xamarin 2016a.)

Xamarinia voidaan käyttää Xamarinin omalla Xamarin Studio-ohjelmalla, joka on saatavilla Windows- ja Mac-käyttöjärjestelmille. Xamarin Studiolla voidaan kehittää Android- ja iOS-sovelluksia. Xamarin Studioon saa käyttöönsä emulaattorit, joita voidaan käyttää sovellusten testaamiseen nopeasti. (Xamarin 2016b.)

Visual Studioon on saatavilla Xamarin-lisäosa, joka mahdollistaa sovellusten kehittämisen Windows Phone-, Android- ja iOS-laitteille. Näin ollen ei tarvita eri kehitystyökaluja eri järjestelmille. Visual Studioon on saatavilla myös emulaattorit jokaiselle edellä mainitulle käyttöjärjestelmälle, joita voidaan käyttää nopeaan sovelluksen testaukseen. Tämä antaa suuntaa siitä, miltä sovellus tulee näyttämään oikealla laitteella. Emulaattoreihin on mahdollista luoda erikokoisia ja eri järjestelmäversion omaavia virtuaalisia laitteita. (Xamarin 2016c.)

Ohjelmointi Visual Studion Xamarin-lisäosassa ja Xamarin Studiossa tapahtuu C#-ohjelmointikielellä Androidille, iOS:lle ja Windows Phonelle. Tämä mahdollistaa sen, että ohjelmoijan tulee osata vain yhtä ohjelmointikieltä, verrattuna siihen, että tulisi osata käyttöjärjestelmien natiiveja ohjelmointikieliä. (Xamarin 2016d.)

3.2 Käyttöliittymätyökalut Xamarinissa

Xamarinin Visual Studio-lisäosa sisältää Designer nimisen toiminnon, jonka avulla kehittäjä voi luoda sovelluksen ulkoasun käytännössä ilman että tarvitsee koskea ulkoasun koodiin ollenkaan. Designer näyttää käyttöjärjestelmissä oletuksena olevat ulkoasutoiminnot tekstikentistä kuvapohjiin.

Tämän lisäksi Designerillä voidaan määrittää toimintojen ja ulkoasun välit, marginaalit ja koot suoraan samassa ruudussa mihin toiminnot asetetaan. Ulkoasu voidaan luoda ilman että sovellusta käytetään välillä emulaattorissa, sillä Designer näyttää koko ajan sovelluksen todellisen ulkoasun eri kokoisilla näytöillä. (Xamarin 2016e.)

Xamarinissa Androidin ulkoasutiedostot ovat axml-päätteisiä toisin kuin Androidin natiiviohjelmointikielellä, jossa ne ovat xml-päätteisiä. Kaikki mitä voidaan lisätä Designerin kautta, voidaan lisätä myös koodin kautta suoraan axml-tiedostoon. Mutta axml-tiedostoon voidaan lisätä myös erilisiä paketteja, kuten vanhempien versioiden tukipaketteja ja käyttäjän muokkaamia toimintoja. Näitä voidaan myös asetella ja säätää kohdalleen koodin kautta, mutta Designerin kautta näin ei voida tehdä. (Xamarin 2016e.)

4 KÄYTTÖLIITTYMÄSUUNNITTELUN HUOMIOITA

Käyttöliittymä on muutakin kuin kaikkien toimintojen asettaminen ruudulle näkyviin ja toimimaan saaminen. Käyttöliittymän tulee sopia sovelluksen tarkoitukseen ja oletettuun kohderyhmään. Käyttöliittymää ei pidä myöskään suunnitella sen mukaan, mikä kehittäjästä itsestään näyttää parhaimmalta. Esimerkkinä tästä on fonttien valinta. Fontit voivat olla kehittäjän mielestä hienoja, mutta tosiasiaassa ne voivat olla erittäin vaikealukuisia ja aiheuttavat vain turhaa sotkua näytölle. Liian paljon kuvia voi aiheuttaa sen, että käyttäjän keskittyminen kulkeutuu muualle kuin siihen missä se tulisi olla. (Getting Real 2016.)

Käyttöliittymästä on helppo tehdä liian laaja ja sotkuinen. Tämä mahdollistaa sen, että sovelluksen käyttäjä ei tiedä mistä lähtisi liikkeelle ja on erittäin todennäköistä, että sovellusta ei tulla käyttämään. Tämän takia on parempi, jos sovelluksessa ei ole liikaa toimintoja tai graafisia ominaisuuksia. Tämä pätee erityisesti silloin, jos sovellusta tehdään pienille laitteille. Silloin jos lisätään paljon kuvia ja toimintoja, iso osa jää näkymättömiin, tai käyttäjän tulisi liikutella ruutua, jotta voisi nähdä loputkin toiminnot. Tämä ei ole isommilla näytöillä niin merkittävä ongelma, mutta isommillakin laitteilla on mahdollista, että toimintoja lisätään liikaa yhdellä kertaa näkyviin. (Getting Real 2016.)

Käyttöliittymää kehittäessä olisi parempi, jos samankaltaisia toimintoja liittäisiin esimerkiksi alivalikoihin ja pudotusvalikoihin. Näille olisi sitten hyvä lisätä sopivat kuvat tai otsikot, jotta käyttäjä löytäisi ne helposti. Tässäkin on myös mahdollista mennä liian pitkälle, ja olettaa että käyttäjä löytää ja osaa käyttää kaikkia painikkeita ja valikoita. (Getting Real 2016.)

5 TIEDONKERÄÄMISEN TAPOJA

Luvussa käydään läpi tiedonkeräämiseen liittyviä toimintoja ja eleitä teoriasolla. Mainitut toiminnot ja eleet käydään läpi siten, että jokaisesta annetaan esimerkki mahdolliseen käyttöön liittyen.

5.1 Erilaiset toiminnot

Androidissa on useita erilaisia toimintoja, joilla tietoa voidaan kerätä. Tavat vaihtelevat yksinkertaisista painalluksista raahaamiseen. Eri toimintoilla on erilaiset käyttötarkoitukset ja kaikki toiminnot eivät välttämättä sovi sovelluksen toimintaan. Osa toimintoista on nopeita käyttää, mutta osa taas on hitaampia. Esimerkkinä nopeasta toiminnosta on yksinkertainen painallus ja hitaampi toiminto olisi esimerkiksi pitkä painallus. Toisaalta, eri tavoilla samoille painikkeille saadaan eri toimintoja, kuten alivalikon avaus pitkällä painalluksella ja lyhyellä painalluksella valinta. (Android 2016e.)

Myös muuta tietoa tulee olla kerättävissä ja sitä voidaan kerätä esimerkiksi Spinner- toiminnolla, tekstikentillä ja Radio buttoneilla. Spinner-valikkoon saadaan lyhyitä tekstejä, joita voidaan käyttää valintoina. Radio buttonit voidaan asettaa ryhmiin, jolloin yhdestä ryhmästä voidaan valita vain yksi vaihtoehto. Radio buttoneiden taakse voidaan asettaa muita toimintoja, kuten vaikkapa useampia Spinner-toimintoja. Esimerkiksi Spinner-toimintoihin voidaan asettaa arvot, joista valinnan mukaan sovellusta voidaan hallita. (Android 2016i.)

5.2 Eleiden käyttäminen

Toimintojen lisäksi Androidista löytyy erilaisia eletoimintoja. Tavanomaisiin eleisiin lukeutuu muun muassa kaksoispainallus. Tavanomaisten eleiden lisäksi löytyy tarkempia ja monipuolisempia eleitä, kuten erilaiset kuviot ja usean sormen toiminnot. Eletoimintoja on erilaisia, mutta pitää sovelluskohtaisesti miettiä, mitkä sopisivat mihinkin sovellukseen. (Android 2016e.)

Tavanomaiset eleet eivät vaadi ohjelman koodin erityistä muokkaamista vaan ne sisältyvät Androidin GestureDetector luokkaan. Luokkaan sisältyy rullaaminen, kaksoispainallus, painallus ja ”heittäminen”. Näillä voidaan kerätä monenlaista tietoa. Osaa näistä saattaa olla vaikea oppia käyttämään sovelluksessa, koska niille saattaa löytyä Androidista jokin käyttökohde, joka voi olla käyttökohteiltaan huomattavasti erilainen kuin se sovelluksessa tulisi olemaan. (Android 2016f.)

Monipuolisempia eleitä Androidissa on liikkeen seuraaminen ja monen sormen toiminnot. Nämä toiminnot tulee lisätä sovellukseen itse ja niille on mahdollista lisätä monenlaisia toimintoja.

Liikkeen seuraaminen mahdollistaa eleiden luonnin esimerkiksi siten, että seurataan mistä pisteestä sormea on lähdetty vetämään ja mihin se on pysäytetty, tai vaikkapa erilaisten kuvioiden luominen. Tämän lisäksi on mahdollista seurata sormen nopeutta näytöllä. Yleisesti ottaen nopeutta seurataan sen vuoksi, että selvitetään, tekikö käyttäjä eleen, vai liikuttiko käyttäjä sormea vahingossa näytöllä. (Android 2016g.)

Monen sormen toimintoja voidaan käyttää esimerkiksi "zoomaamiseen". Monen sormen toiminnot otetaan huomioon sitä varten, jos käyttäjä vahingossa laskee toisen sormen näytölle, vaikkapa raahatessaan sovelluksessa olevaa kohdetta toiseen sijaintiin. Tällä voidaan varmistaa, että kohde ei siirry toisen sormen kohdalle, vaan se voidaan säilyttää oikean sormen kohdalla. (Android 2016h.)

6 TOIMEKSIANTAJAN TAVOITTEET SOVELLUKSEEN JA KÄYTTÖLIITTYMÄÄN

Luvussa käydään läpi sovelluksen tavoitteita toimeksiantajan näkökulmasta. Näihin sisältyy se miten tiedon kerääminen on tehty aiemmin, sovelluksen tavoite ja toiminnallisuus.

6.1 Alkuperäinen tapa kerätä tietoa

Alun perin toimeksiantajan tiedonkeruu tapahtui pelin jälkeen, ottelu videolta katsottaessa. Tiedot lisättiin Excel-tiedostoon, joka sisältää monta sivua ja monia taulukoita. Yhdessä nämä johtavat siihen, että muutosten tekeminen ja tilastojen tarkasteleminen vie todella paljon aikaa. Tilastoja ei voida tarkastella esimerkiksi erätauoilla, jolloin voitaisiin tehdä muutoksia taktiikoihin. Excel-tiedosto sisältää päätaulun, johon kerätään pelitiedot, josta Excel sitten laskee ja luo tarkempia tietoja.

6.2 Sovelluksen tavoite

Toimeksiantajan tavoite oli selkeä: mobiilisovellus, jolla voidaan kerätä tietoa ja näyttää tilastoja reaaliajassa sekä tallentaa kaikki kerätty tieto, joka voidaan sitten viedä sovelluksen ulkopuolelle. Tallennettu tieto olisi hyvä olla Excelin ymmärrettävissä, kuten vaikkapa csv-tiedosto. Sovelluksessa näytettävät tilastot liittyisivät pelin sen hetkiseen tilanteeseen ja niiden avulla voitaisiin tehdä muutoksia ketjuihin ja taktiikoihin. Sovelluksen avulla tiedon keräämisen pitäisi tapahtua nopeasti. Tilastojen näyttäminen sovelluksessa pitäisi olla sujuvaa eikä sovellus saisi näyttää ylimääräistä tietoa.

6.3 Toiminnallisuuden huomioon ottaminen

Tieto tullaan keräämään sovelluksessa yksi syöttö tai kiekon kuljetus kerrallaan, joten näkyviä painikkeita on useita. Tässä pitää huomioida se, ettei painikkeita tule olemaan niin montaa, että virhepainalluksien määrä kasvaa, tai että käyttäjä sekoittaa kahden painikkeen toiminnot keskenään. Tietoa pitää myös pystyä keräämään nopeasti, kuitenkin sekoittamatta käyttäjää liian monimutkaisilla toiminnoilla.

Pelin toiminnoille tulee myös lisätä tarkentavat valinnat, jotka mahdollistavat erikoistilanteiden kirjaamisen. Näitä voidaan kerätä esimerkiksi eleillä ja alavalikoilla riippuen tarvittavasta tiedosta ja siitä mikä olisi luonteva vaihtoehto kyseiselle toiminnolle. Sovelluksessa tulee myös olla mahdollista muokata pelissä olevia ketjuja tilastojen perusteella.

Sovelluksen sisäisesti tulee myös olla mahdollista esittää oleellista tietoa kaikesta kerätystä tiedosta. Vaikka tietoa kerätään huomattava määrä,

kaikkea ei tarvita pelin aikana, vaan niitä voidaan käyttää myöhemmin tarkempien analyysien tekemiseen. Se tieto mikä on oleellista, tullaan päättämään yhdessä toimeksiantajan kanssa.

7 SOVELLUKSEEN SOPIVAT TOIMINNOT JA ELEET

Tässä luvussa käydään läpi sovellukseen mahdollisesti sopivia toimintoja ja eleitä. Näistä käydään tarkemmin läpi mihin niitä voitaisiin sovelluksessa käyttää, ja mahdollisesti niihin liittyviä ongelmia.

7.1 Tiedon keräämiseen sopivat toiminnot ja eleet

Alla olevaan taulukkoon on kerätty toimintoja ja eleitä, joita voitaisiin käyttää sovelluksen tiedonkeruun apuna. Taulukosta löytyy lisäksi mahdollisia käyttökohteita jokaiselle toiminnolle ja eleelle. (Taulukko 2.)

Taulukko 2. Käytettäviä toimintoja ja eleitä

Toiminto/Ele	Mahdollinen käyttökohte
Lyhyt painallus	Yleisimmät toiminnot, kuten tavallisen syötön merkitseminen.
Pitkä painallus	Lisävalikoiden ja tarkentavien toimintojen avaaminen.
Kaksoispainallus	Lisävalikoiden avaaminen.
Raahaaminen	Syötön merkitseminen, tai pelaajaan siirtyminen toiseen pisteeseen.
Kuvion piirtäminen	Tarkentavien toimintojen tekeminen kuvioilla alivalikoiden sijaan.
Pyyhkäisy	Virheellisen merkinnän kirjoittaminen kerättyyn tietoon.
Painike	Tavalliseen painikkeeseen voidaan lisätä vaikkapa pelaajan numero, tai aloituspiste.
Spinner	Spinnerillä voidaan kerätä vaikkapa ketjukokoonpanot.
Pudotusvalikko	Pudotusvalikoihin voidaan lisätä erikoistapahtumat ja muut tarkentavat tiedot.
Radio button	Radio buttonia voidaan käyttää ketjujen valintaan rajoittamalla yhden ketjun kerrallaan valituksi, tai erän valintaan.

Lyhyellä painalluksella saadaan kerättyä nopeimmat sovelluksessa käytetyt toiminnot. Näihin sisältyy valikoiden käyttäminen, mutta myös nopeiden pelitapahtumien kerääminen. Tavallinen syöttö kuuluu näihin tapahtumiin. Tapahtumien lisäksi lyhyillä painalluksilla tehdään valintoja pelikokoonpanojen kohdalla sekä pudotusvalikoiden erikoistoimintojen ja pelaajanumeroiden valintoja.

Pitkällä painalluksella saadaan avattua pelaajapainikkeille pudotusvalikot, mutta myös yhdistettynä raahaamiseen, sillä voidaan kerätä syötöt. Osa erikoistapahtumista voidaan myös merkitä raahaamalla. Pitkää painallusta ja raahaamista voidaan käyttää samaan aikaan eri painikkeilla, mutta käyttäjälle saattaisi alkuun olla haastavaa oppia ero, kumpaa toimintoa tulisi käyttää. Näin toimiessa pitäisi tarkkaan miettiä minkälaisille pelitapahtumille ja painikkeille toiminnot lisätään.

Kaksoispainallus voisi toimia samoissa toiminnoissa kuin pitkä painallus, mutta kaksoispainallusta käyttäessä virhepainallusten määrä voisi kasvaa huomattavasti, ja tällöin jopa hidastaa sovelluksen käyttöä.

Kuvion piirtäminen voisi tuoda käyttäjälle nopean ratkaisun erikoistoimintojen kirjaamiseen, verrattuna alivalikoiden käyttämiseen. Tällöin sovelluksen käyttäjä voisi esimerkiksi suoraan syötön jälkeen piirtää kyseiseen tilanteeseen kuuluvan kuvion. Tässä vaihtoehdossa käyttäjän tulisi muistaa kaikki kuviot, joita sovelluksessa tarvitsee, ja uutena käyttäjänä kuvioden opettelu veisi paljon aikaa. Koska kuvioita tulisi olemaan merkittävä määrä, voisi kuvioden piirtämisessä tulla sekaannuksia. Esimerkiksi väärin piirretyt kuviot eivät tekisi mitään, tai käyttäjä voisi piirtää väärän kuvion. Tässä ratkaisussa virheiden määrä voisi kasvaa ja tietojen korjaaminen pelin jälkeen veisi aikaa.

Pyyhkäisyllä käyttäjä voisi luoda merkinnän kerättyyn tietoon, josta käyttäjä huomaa, että kyseisessä kohtaa on tapahtunut virhepainallus. On mahdollista, että käyttäjä saattaa vahingossa pyyhkäistä johonkin suuntaan, jolloin tietoihin tulisi väärä merkintä virheestä. Tätä vastaan voidaan luoda varmistuksia, jolloin käyttäjän tulisi esimerkiksi vetää tarpeeksi pitkä veto, että ele rekisteröityisi. Liian pitkän vedon vaativa ele taas voisi hidastaa sovelluksen käyttämistä, joten pitäisi löytää oikea pituus kokeilujen kautta.

Painikkeilla voidaan kerätä kaikenlaista nopeaa toimintaa vaativia tietoja yhdistettynä lyhyeen painallukseen. Painikkeisiin voidaan lisätä pelaajat, tai sitten painikkeita voidaan käyttää esimerkiksi pelisuunnan valitsemiseen.

Spinner toimintoa voidaan käyttää ketjujen muodostamiseen ja niiden muokkaamiseen, kun tilanne niin vaatii. Spinnereitä voidaan käyttää myös kaikkien pelaajien listaamiseen, tai esimerkiksi erän valintaan.

Pudotusvalikosta voidaan kerätä tietoa erikoistoiminnoista tai sillä voidaan valita, kuka pelaajista on esimerkiksi ollut aloituksessa. Pudotusvalikon aukaiseminen voidaan asettaa esimerkiksi jokaisen painikkeen kohdalle. Valikko voidaan avata painikkeesta riippuen esimerkiksi yksinkertaisella painalluksella tai pitkällä painalluksella.

Radio button toiminnoilla voidaan esimerkiksi valita käynnissä oleva erä, tai ketju mikä sillä hetkellä pelikentällä on. Radio buttoneita voidaan käyttää myös yhdistämään useampi toiminto saman valinnan taakse.

7.2 Tiedon esittämiseen sopivia toimintoja ja eleitä

Alla olevaan taulukkoon on kerätty tiedon esittämiseen soveltuvia toimintoja ja eleitä, ja niille on kerätty mahdollisia käyttökohteita. Samalle toiminnolle tai eleelle voi olla useita erilaisia käyttökohteita, ja niitä selvennetään taulukon jälkeen. (Taulukko 3.)

Taulukko 3. Käytettäviä toimintoja ja eleitä

Toiminto/Ele	Mahdollinen käyttökohde
Tekstikenttä	Pelaajan numeron, tai tapahtuneen toiminnon esittäminen.
Pudotusvalikko	Pudotusvalikkoon voidaan lisätä käytetyt pelaajat, tai erikoistoiminnot, joita on käytetty.
Spinner	Spinner-toimintoon voidaan lisätä vaikkapa pelaajanumerot.
Radio Button	Radio buttoneilla voidaan tehdä valinnat esimerkiksi erästä ja kentällä olevista ketjuista.
Swipe	Swipe eli pyyhkäisyllä voidaan siirtyä esimerkiksi tilastoja katsellessa toiseen näkymään.

Tekstikentällä voidaan esittää monenlaista tietoa. Sovelluksissa ei ole rajoitusta siitä, kuinka monta tekstikenttää samalla kertaa näytöllä voi olla. Ainoa rajoittava tekijä on näytön koko. Tekstikenttiin voidaan lisätä pelaajien tarkemmat tiedot, kuten vaikkapa nimet ja pelipaikat.

Pudotusvalikoihin voidaan lisätä pelin aikana käytettyjä pelaajia. Valitsemalla pelaaja, voidaan esittää yksityiskohtaiset pelitiedot kyseiselle pelaajalle.

Spinnereillä voidaan esittää pelaajat, jotka ovat käytettävissä, tai voidaan valita haluttu tilastonäkymä. Liian monen Spinnerin tiheä asettelu voi olla sovelluksessa haasteellista, jos niille ei aseteta ulkoasussa selkeitä rajoja.

Radio buttoneilla voidaan esittää mahdolliset ketjut, tai käynnissä oleva erä. Radio buttonit voidaan asettaa ryhmiin, jolloin niistä voidaan valita vain yksi kerrallaan.

Pyyhkäisyllä voidaan siirtyä tilastoja katsellessa toiseen näkymään, jossa esimerkiksi voidaan esittää tietoja eri tarkkuudella tai vaikkapa vastustajajoukkueen maalitilanteet.

8 KÄYTTÖLIITTYMÄN TOTEUTUS

Luvussa käydään läpi sovelluksen käyttöliittymän toteutus vaiheittain. Vaiheet vastaavat sitä mitä toimeksiantajalle on esitetty. Eri versioiden jälkeen on pyydetty kommentteja ja kehityskohteita, joita on sitten lähdetty muokkaamaan.

8.1 Käyttöliittymän ensimmäinen versio

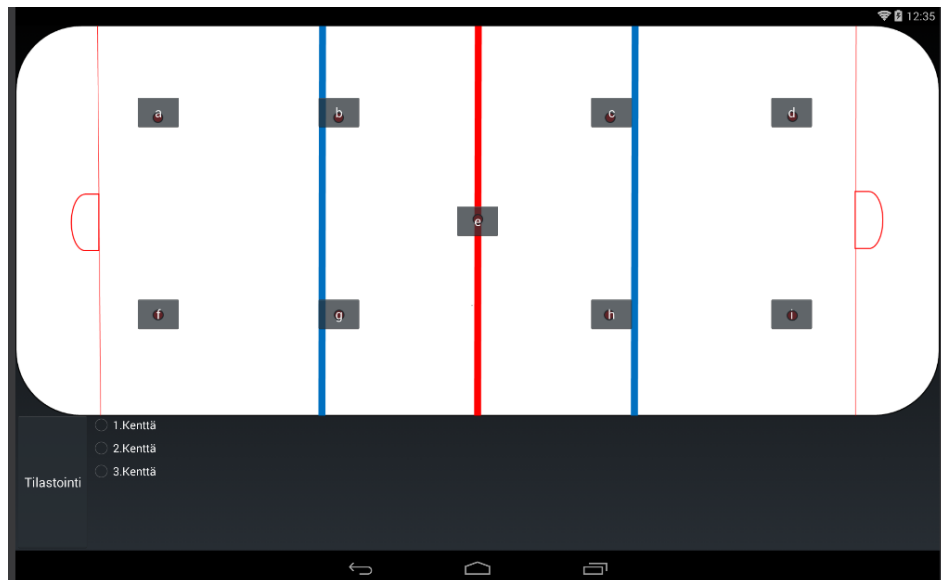
Näihin valintoihin päädyttiin ensimmäisen toimeksiantajan palaverin jälkeen. Sovellukseen lisättiin tiedonkerääminen eri tavoilla eri ikkunoihin. Tämän version jälkeen otettiin uudelleen yhteyttä toimeksiantajaan ja esiteltiin sen hetkinen tilanne. Tästä pyydettiin palautetta ja kehityskohteita, joiden perusteella tehtiin lisäpäätöksiä ja valintoja.

Sovelluksen tähän versioon lisättiin mahdollisia toimintoja, kuten eleitä, joista esimerkkeinä raahaaminen, painallukset lyhyenä ja pitkänä sekä erilaisia valikoita. Näiden avulla saatiin luotua tarpeeksi erilaiset tavat kerätä tietoa. Näin voitiin kokeilla, mitkä toiminnot sopisivat parhaiten, ja mistä voitaisiin luopua.

Sovellus koostui ensimmäisessä versiossa viidestä eri ikkunasta, joista kaksi ikkunaa toimi pelikenttänä eri pelisuuntiin. Näiden lisäksi sovelluksessa oli ikkuna aloituspisteille, tilastoille ja sovelluksen päävalikolle. Eri pelisuunnat oli toteutettu eri tavoin, jotta saatiin kokeiltua, mitkä ominaisuudet sopisivat parhaiten sovelluksen käyttöliittymään.

Päävalikko sisälsi vain painikkeen, jolla siirryttiin aloituspisteet sisältävään ikkunaan. Muita toimintoja tai eleitä päävalikkoon ei oltu lisätty tässä vaiheessa.

Aloituspisteet-ikkuna sisältää painikkeet aloituspisteille, tilastot-ikkunaan siirtymisen ja kentällisen valinta Radio button painikkeilla. Radio buttonit on asetettu yhteen ryhmään, jolloin niistä voidaan valita vain yksi kerrallaan. Pelikenttä painikkeiden takana on vain taustakuva. (Kuva 1.)

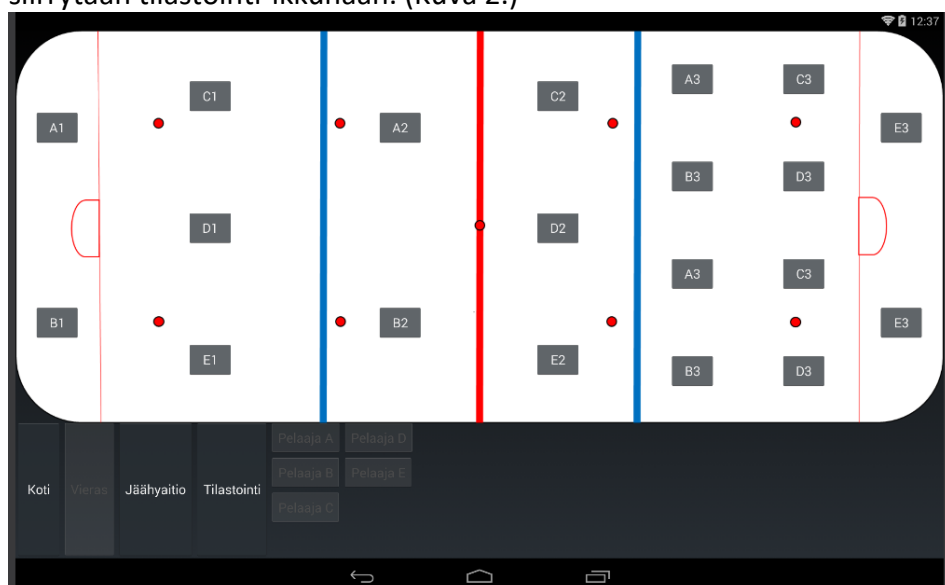


Kuva 1. Sovelluksen aloituspisteiden ikkuna

Aloituspisteet sisältävät vain yksittäisen painalluksen toiminnon. Pistettä painamalla avautuu painetun pisteen viereen popup menu, joka tässä vaiheessa sisälsi vaihtoehdot sille, että voittiko aloituksen koti- vai vierasjoukkue. Kun oli valittu voittanut joukkue, avautui uusi popup menu, josta valittiin aloituksen voittanut pelaaja.

Varsinaiset pelikenttäikkunat on toteutettu kahdella eri tapaa. Niissä on käytetty erilaisia toimintoja ja eleitä. Ulkoasultaan ikkunat ovat samankaltaiset, mutta toiseen ikkunaan on lisätty enemmän painikkeita pelaajille. Ikkunoissa pelisuunnat ovat päinvastaiset.

Molemmista ikkunoista löytyy painikkeet jokaiselle pelaajalle, toiseen ikkunaan siirtyminen, jäähyt ja siihen liittyvät painikkeet sekä painike jolla siirrytään tilastointi-ikkunaan. (Kuva 2.)



Kuva 2. Pelikenttä ikkunan ulkoasu

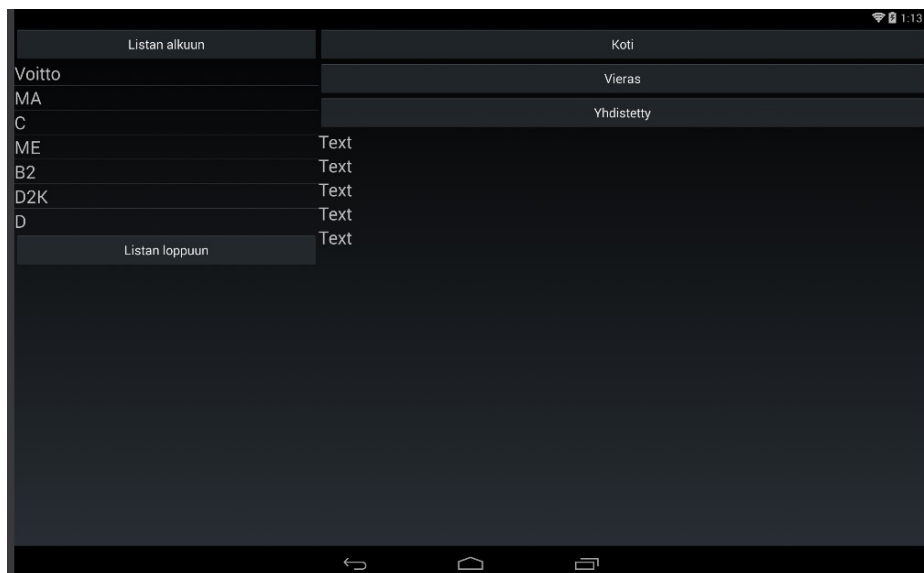
Vierasjoukkueen ikkunassa käytettyjä toimintoja ovat lyhyet ja pitkät painallukset, popup menut ja raahaaminen. Syötöt tässä ratkaisussa tehdään vetämällä pelaajan painike kohtaan, johon syöttö meni. Painiketta ei tarvitse pudottaa täysin toisen pelaajan päälle, vaan jokaisen pudotettavan alueen ympärille tulee näkyvät rajaviivat, jonka sisään painike on pudotettava. Raahaaminen aloitetaan pitkällä painalluksella, joten erikoistoiminnot avautuvat lyhyellä painalluksella. Lisävalinnoista valitaan tavallisella painalluksella haluttu tapahtuma, ja riippuen valinnasta, siitä avautuu tarkentava toiminto esimerkiksi pelaajan valinnalle. Tätä voidaan käyttää esimerkiksi silloin kun pelaaja ei ole ollut omalla paikallaan.

Muut painikkeet kuten jäähyjen merkintä tai tilastointi-ikkunaan siirtyminen tapahtuu lyhyellä painalluksella. Esimerkiksi jäähy-painiketta painamalla avautuu popup menu, josta löytyy kentällä olevat pelaajat. Vaihtoehtoja valitaan se pelaaja, kenelle annetaan jäähy. Pelaajan ollessa jäähyllä, on siihen pelaajaan linkitetyt painikkeet poistettu käytöstä, mutta näkyvissä. Lisäksi pelaajien jäähy-painikkeet ovat näkyvissä koko ajan, mutta poistettu käytöstä.

Koti-ikkunassa ei ole käytetty raahaamista, mutta muut toiminnot kuten lyhyet ja pitkät painallukset ja popup menut löytyvät tästä ratkaisusta. Syötöt koti-ikkunassa tehdään yksinkertaisesti painamalla eri pelaajien painikkeita lyhyellä painalluksella ja pelaajien lisätoiminnot tulevat esiin painamalla pelaajan painiketta pitkään. Lisävalikkojen valinnat tehdään tässäkin mallissa lyhyellä painalluksella.

Koti-ikkunan ulkoasu on hieman erilainen kuin vierasjoukkueen ikkunassa. Se ei sisällä päätympelin painikkeita molemmilla puolin maalia, vaan vain toisella puolella. Jäähyllä olevien pelaajien painikkeita ei näytetä, ellei pelaajia ole jäähyllä.

Tilastoinnista löytyy tässä ratkaisussa vain painikkeita ja tekstikenttiä. Ikkunan vasemmassa reunassa on pelitietoja mitä sovelluksella on kerätty. Ikkunan oikeassa reunassa on painikkeet, joista voidaan valita tarkasteltavat tiedot. Vaihtoehtoina on kotijoukkueen, vierasjoukkueen ja yhdistettyjen tietojen tarkastelu. Painikkeiden alla on tekstikenttiä, joihin tulee tekstiä riippuen siitä, mitä tietoa on valittu vasemmalla olevasta listasta. Tässä vaiheessa tilastoinnista ei ollut erityisemmin sovittu mitään, joten tämä ratkaisu on vain yksi vaihtoehto. Tämän perusteella voitaisiin lähteä kehittämään tilastointia eteenpäin. (Kuva 3.)



Kuva 3. Tilastointi-ikkuna

Tämä käyttöliittymän versio lähetettiin jääkiekkovaikuttajille tutkittavaksi. He tekisivät päätökset siitä, mitkä ominaisuudet olisivat heidän mielestään parhaiten toimivia. Sovelluksen tästä versiosta puuttuva ominaisuus on ketjujen muokkaaminen, joka varmasti lisätään sovelluksen seuraavaan versioon.

8.2 Käyttöliittymän toinen versio

Sovelluksen toinen versio esiteltiin toimeksiantajalle ja jääkiekkovaikuttajille palaverissa, jossa käytiin läpi sovelluksen tilaa sillä hetkellä. Sovellukseen oltiin lisätty ketjujen muokkaaminen, pelikello ja pelaajien tietojen syöttäminen ja muokkaaminen. Näiden lisäksi sovelluksesta oltiin poistettu ominaisuuksia, joita ei tarvita. Lisäksi oltiin tehty muutoksia sen mukaan, millaista palautetta oli edellisestä versiosta saatu.

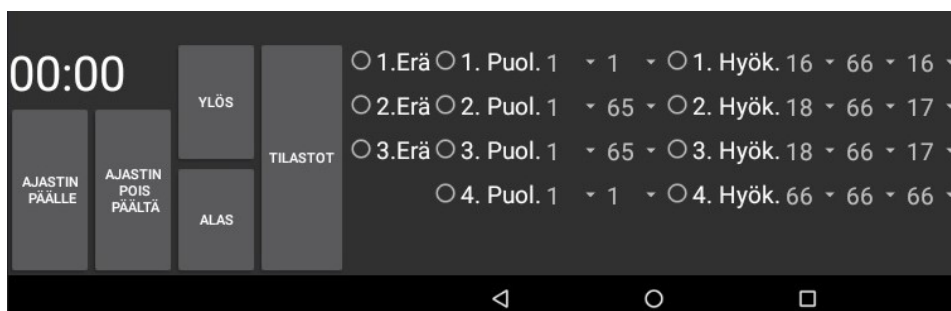
Sovellukseen lisättiin pelaajien tietojen tallentaminen. Pelaajille voidaan syöttää pelinumero, pelipaikka sekä etu- ja sukunimi, joista vain pelaajan numero ja pelipaikka ovat pakollisia. Pelinumero-kenttään voi syöttää vain numeroita ja pelipaikkaan jokin annetuista vaihtoehdoista. Nimikentissä ei ole rajoituksia syötettävässä tiedossa. Pelipaikka-kentän voisi korvata Spinner-toiminnolla, jolloin käyttäjä voisi valita varmasti oikean vaihtoehdon. Tämä poistaisi kirjoitusvirheiden mahdollisuuden.

Pelaajien tiedot tallentuvat paikalliseen tietokantaan, josta niitä voidaan käyttää muualla sovelluksessa. Ikkunan oikeassa reunassa oleva lista sisältää kaikki pelaajat, jotka tietokannasta löytyvät. Listaa voi liikuttaa ylös tai alas painamalla listan ylä- ja alapuolella olevia painikkeita. Tämän lisäksi listaa voi selata vetämällä joko ylös- tai alaspäin listan kohdalla. (Kuva 4.)

TALLENNA PELAAJA	LISTAN ALKUUN
Pelaajan numero: Pakollinen	1
Pelaajan paikka: Pakollinen, Syötä arvo VP, OP, OL, VL tai KH	op
Etinimi	2
Sukunimi	vp
MUOKKAA	
POISTA	33

Kuva 4. Pelaajat sisältävän tietokannan muokkaaminen

Aloituspisteet-ikkunan pelialue ei ole muuttunut edellisestä versiosta, mutta pelialueen alapuolella sijaitsevat toiminnot ovat muuttuneet merkittävästi. Alueen sisältö on samanlainen aloitusikkunassa, kuin myös varsinaisessa peli-ikkunassa. Alueelle on lisätty pelikello ja sen säätämisen mahdollisuus. Uutena ominaisuutena on lisätty peliketjujen valinta ja muokkaaminen pelin aikana. Valinnat sisältävät puolustajaparit ja hyökkäysketjut. Näistä on samaan aikaan valittava yksi puolustuspari ja yksi hyökkäysketju. (Kuva 5.)

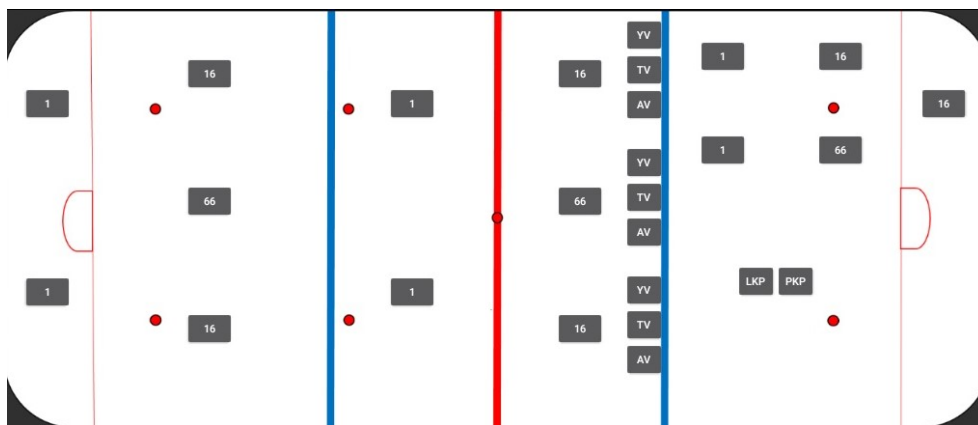


Kuva 5. Alapalkin sisältö

Ketjujen kokoonpanojen muokkaaminen on toteutettu Spinner-toiminnoilla, joihin sisältö haetaan tietokannasta. Näytettävien pelaajien rajausta erittelee vain hyökkäysketjujen ja puolustusketjujen pelaajat keskenään.

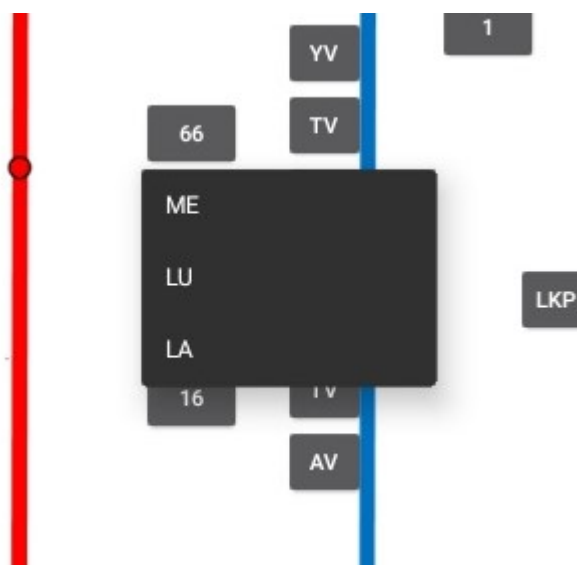
Pelikentän ulkoasu on muuttunut hieman ja siihen on lisätty uusia painikkeita. Ikkunassa käytettävä alapalkki on lähestulkoon samanlainen kuin aloituspisteet sisältävässä ikkunassa.

Pelikenttään on lisätty painikkeet ylivoimalle, alivoimalle ja tasavoimalle sekä painikkeet lyhyelle ja pitkälle kulmapelille. Edellä mainitut painikkeet ovat lyhyellä painalluksella toimivia, eivätkä sisällä muita toimintoja. Yli-, ali- ja tasavoiman painikkeet on sijoitettu kolmeen kohtaan, joilla merkitään tieto, mistä kohtaa kenttää on edetty maalialueelle. Pelaajien painikkeiden nimet haetaan aktiivisina olevista ketjuista, jotka määritetään kentän alapuolella olevista toiminnoista. (Kuva 6.)



Kuva 6. Pelikentän ulkoasu

Pelaajien painikkeista pitkällä painalluksella avautuvat lisävalinnat riippuvat painikkeen sijainnista kentällä. Esimerkiksi puolustuspään pelaajilla on erilaiset valinnat, kuin maalialueen pelaajilla. Lisävalinnat itsessään voivat vielä sisältää valintoja, kuten vaikkapa valinnan sille, kuka pelaajista todellisuudessa teki maalin. Lisävalintojen sisältö ei ollut muuttunut sen jälkeen, kun ne on sovellukseen lisätty. (Kuva 7.)



Kuva 7. Pelaajien lisävalinnat

Alapalkin toiminnot ovat muuten samanlaisia kuin aloituspisteet sisältävän ikkunan, mutta erona on painike "VIRHE". Sillä merkitään tallennettaviin tietoihin, että tässä kohtaa on tapahtunut virhepainallus. Tämän avulla käyttäjä voi myöhemmin muokata virheellisiä tietoja ilman, että tarvitsisi käydä koko peliä läpi. (Kuva 8.)



Kuva 8. Pelikentän alapalkki

Pelikenttä-ikkunasta löytyy painike, jolla merkitään vastustajan maali. Painike tulee näkyviin, kun sovelluksessa on merkitty kiekon menetys.

Sovelluksen sisäisestä tilastojen tarkastelusta päätettiin ennen tapaa- mista, että sitä käydään läpi myöhemmin. Tässä sovellusversiossa tilas- tointi-ikkunasta poistettiin kaikki toiminnot. Tilastointi-ikkunaan siirtyessä sovellus luo siihen asti tallennetuista tiedoista csv-tiedoston, jota voidaan katsella ja käsitellä sovelluksen ulkopuolella.

8.3 Käyttöliittymän kolmas versio

Kolmannessa versiossa on palaverista saadun palautteen perusteella tehty pienempiä ja suurempia muutoksia. Sovellus on käyttökelpoinen kokeilta- vaksi oikean pelin aikana. Muutoksia, mitä kolmanteen versioon on tehty, ovat esimerkiksi pelisuunnan valinta, vastustajan tietojen tallentaminen ja ilmoituksen lisääminen näytölle, kun käyttäjä on painanut painiketta.

Kolmannen version suurin muutos on vastustajan pelin seuraaminen ja tal- lentaminen. Vastustajan pelistä ei tallenneta tietoja samalla tarkkuudella kuin oman joukkueen pelistä. Osa vastustajan pelitiedoista on oleellista oman joukkueen toiminnan kannalta, mutta esimerkiksi vastustajan ketju- jen seuraaminen on epäolennaista.

Painetun painikkeen luoma ilmoitus on toteutettu usealla eri tavalla. Tämä on sen vuoksi että eri laitteissa ei välttämättä ole samoja mahdollisuuksia kuten vaikkapa värinätoimintoa. Pelisuunta on määritettävissä sovelluksen alussa ja pelisuunta valitaan oman joukkueen mukaan.

Pelaajien lisäys-ikkunaan on lisätty sovelluksen nopeamman käytettävyy- den kannalta ketjujen muokkaaminen ja tallentaminen. Nämä tiedot pysy- vät pääpiirteittäin samana, joten ne tallennetaan pysyvästi laitteen muis- tiin. Tietoja voi edelleen muokata myös pelin aikana, ja niissä tehdyt muu- tokset tallentuvat myös laitteen muistiin. (Kuva 9.)

Ketjut esitetään Spinner-toimintojen avulla, joihin haetaan tiedot tietokan- nasta. Myös pelin aikana näkyviin Spinner-toimintoihin haetaan tiedot sa- masta tietokannasta. Spinner-toimintoon haetaan näkyviin vain pelaajan numero, joka on uniikki. Tämä varmistaa sen, että kahdella eri henkilöllä ei

voi olla samaa pelinumeroa. Sama pelaaja voi olla useammassa ketjussa, joten sille ei ole asetettu rajoitteita.

TALLENNA PELAAJA	LISTAN ALKUUN
Pelaajan numero: Pakollinen	1
Pelaajan paikka: Pakollinen, Syötä arvo VP, OP, OL, VL tai KH	op
Etunimi	
Sukunimi	2
	vp
MUOKKAA	
POISTA	
	33
1. Puol. Ketju 1 ▾ 1 ▾	op
2. Puol. Ketju 1 ▾ 65 ▾	
3. Puol. Ketju 1 ▾ 65 ▾	22
4. Puol. Ketju 1 ▾ 1 ▾	op
1. Hyök. Ketju 16 ▾ 66 ▾ 16 ▾	
2. Hyök. Ketju 18 ▾ 66 ▾ 17 ▾	15
3. Hyök. Ketju 18 ▾ 66 ▾ 17 ▾	op
4. Hyök. Ketju 66 ▾ 66 ▾ 66 ▾	44
TALLENNA KETJUT	vp

Kuva 9. Pelaajien lisäys ikkunan ketjujen kokoonpanot

Aloituspisteet sisältävää ikkunaa on muutettu hieman edellisestä versiosta. Ikkunaan on lisätty painike, jolla voidaan valita pelisuunta oman joukkueen näkökulmasta katsoen. Painike avaa valintailmoituksen, josta valitaan haluttu pelisuunta. Pelisuunnan valinta oltaisiin voitu toteuttaa myös esimerkiksi Radio buttoneilla, mutta erillisellä painikkeella avattava ruutu estää virheelliset valinnat. (Kuva 10.)

Valitse joukkueen pelisuunta

PERUUTA
OIKEALTA VASEMMALLE
VASEMMALTA OIKEALLE

1. Erä 1. Puol. 1 ▾ 1 ▾ 1. Hyök. 16 ▾ 66 ▾ 16 ▾ PELISUUNTA

2. Erä 2. Puol. 1 ▾ 65 ▾ 2. Hyök. 18 ▾ 66 ▾ 17 ▾

Kuva 10. Pelisuunnan valinta

Pelikenttä-ikkunan suurin muutos on pelisuunnan näkyminen kentän painikkeiden sijainnissa. Toinen suuri muutos on vastustajan pelin seuraaminen, ei kuitenkaan samalla tarkkuudella kuin oman joukkueen seuranta. Vastustajan painikkeista on karsittu lisävalintoja. Vastustajan tiedot tallen-

netaan vain pelaajan paikan tiedolla, koska vastustajan ketjujen ja pelaajien seuraaminen oman joukkueen lisäksi on haastavaa. Tämän vuoksi vastustajan pelitietojen keräämisestä tärkeimpänä ajatuksena on erityisesti ne tilanteet, joista vastustaja on saanut tehtyä maalin.

Koska pelikentälle on lisätty pelisuunta, on pelikentän ikkunasta poistettu vastustajan maali-painike sen ollessa tarpeeton. Kiekon ollessa vastustajalla, pelikenttä näkyy oman joukkueen pelisuunnasta katsottuna käänteisenä. Säilytetyillä lisävalinnoilla vastustajan pelistä merkitään esimerkiksi maalit ja kiekon menetykset. Kiekon menetyksestä pelikenttä kääntyy oman joukkueen näkymään ja lisävalinnat näyttävät laajat tallentamismahdollisuudet.

9 SOVELLUKSEN LOPPUTILANNE JA JATKOKEHITYS

Tässä luvussa käydään läpi sovelluksen tilanne opinnäytetyön lopulla ja sen jatkokehittäminen. Luvun alussa käydään läpi toiminnot ja eleet, joita käytetään tiedon keräämiseen ja esittämiseen. Lopuksi käydään läpi, mitä jatkokehittämiseen sisältyy.

9.1 Tiedon kerääminen

Tiedon keräämiselle on määritetty useita eri toimintoja. Osalle toiminnoista olisi mahdollista käyttää vaihtoehtoisia menetelmiä. Sovelluksessa oleviin toimintoihin on päädytty, koska ne eivät vaadi käyttäjältä erikoisempaa opettelua, eikä niitä ole mahdollista sekoittaa keskenään.

Lyhyt painallus on sovelluksen käytetyin toiminto. Se on nopein vaihtoehto verrattuna niihin ratkaisuihin, joita sen tilalla on kokeiltu. Tällä merkitään syötöt pelaajien välillä, valitaan tarkentavat toiminnot kuten vaikkapa kiekkojen menetykset, ja lisäksi valikoissa liikkuminen.

Pitkällä painalluksella avataan lisävalinnat sisältävät valikot, sillä yleensä valikoihin listatut tapahtumat eivät ole niin nopeita peliliikkeitä. Tämän takia sovelluksen käyttäjä ehtii valita haluamansa toiminnon. Valinnat tulevat popup menuihin, joihin on mahdollista lisätä eri määrä vaihtoehtoja. Näitä lisätoimintoja on huomattava määrä, ja ne eivät ole samanlaisia koko pelikentän alueella.

Spinner-toiminnoilla kerätään tiedot siitä, ketkä pelaajista ovat olleet kentällä pelin aikana. Pelaajista kerätään myös tiedot siitä, missä ketjuissa pelaajat ovat olleet. Ketjut mistä tiedot tallennetaan, valitaan Radio button-painikkeilla. Näillä painikkeilla tallennetaan myös tieto siitä, mitä missäkin erässä on tapahtunut.

Tekstikentillä tallennetaan pelaajien tiedot tietokantaan, joista niitä voidaan käyttää sovelluksen sisäisesti. Tekstikentillä tallennetaan pelaajanumerot, pelipaikat sekä pelaajan nimi.

Käyttäjän omasta toiminnasta riippumatonta tietoa on pelikellon aika ja sen tallentaminen. Sovellus tallentaa kellossa olevan ajan jokaisen alkavan pelivirtauksen mukana.

9.2 Tiedon esittäminen

Tiedon esittäminen tapahtuu osittain samoilla toiminnoilla kuin tallentaminenkin. Sovelluksen lopullisesta versiosta tilastojen tarkasteleminen jätettiin pois. Tilastojen tarkastelu on päätetty hoitaa jatkokehityskohteena.

Tekstikenttiä käytetään sovelluksessa erityisesti tietokannan hallinnassa. Tekstikentät näyttävät kokonaisuudessaan mitä tietoja pelaajalle on syötetty.

Spinner-toiminnot näyttävät pelaajista pelinumerot, ja näiden perusteella kentällä näytetään valittuna oleviin ketjuihin kuuluvat pelaajat. Spinnerit ovat jaettu kahteen ryhmään, joista toisessa näytetään puolustavat pelaajat ja toisessa näytetään hyökkäävät pelaajat.

Radio button toiminnoilla näytetään käytettävissä olevat ja aktiiviset ketjut sekä käynnissä oleva erä. Näihin toimintoihin on yhdistetty edellä mainitut Spinner-toiminnot.

9.3 Jatkokehitys

Sovellus on valmis käytettäväksi ja kokeiltavaksi oikean pelin aikana. Pidempään kokeilemalla tullaan näkemään, onko jokainen sovellukseen päätynyt ominaisuus paras mahdollinen vaihtoehto. Lisäksi on mahdollista, että sovelluksesta puuttuu ominaisuuksia, joita ei olla osattu kehitysvaiheessa ajatella.

Varsinaisten ominaisuuksien lisäksi kokeilemalla selviää sovelluksen ulkoasun kehittämistarpeet. Voi olla, että osa värivalinnoista, tai painikkeiden käyttäjäväste vaatii kehittämistä. Lisäksi sovellusta tullaan käyttämään viileässä ilmassa pidempään, jolloin nähdään, onko sillä vaikutusta näytön käytettävyyteen. Tämän perusteella voidaan tehdä muutoksia ulkoasuun esimerkiksi tiheään asetettujen toimintojen välien kasvattamista.

Sovelluksesta toistaiseksi pois jätetty ominaisuus, kerätyn tiedon tarkastelu sovelluksessa, on jatkokehityksen kannalta tärkein ominaisuus. Pelin aikana kerätystä tiedosta tarvitsee eritellä oleellinen tieto ja sen lisäksi pitää kehittää sopiva tapa sen esittämiseen. Oleellista tietoa kerääntyy pelin aikana merkittävä määrä, joten ikkunasta tulee helposti sekava.

10 YHTEENVETO

Opinnäytetyön tarkoituksena oli kehittää nopea ja helppokäyttöinen käyttöliittymä sovellukseen, jonka tarkoituksena on tallentaa tietoa jääkiekon pelivirrasta. Alkuperäinen tapa kerätä tietoa oli pelin jälkeen käydä esimerkiksi videolta läpi jokainen tilanne ja sitten tallentaa tiedot Excel-tiedostoon. Tämä tapa oli liian hidas. Sovellus oli luonnollinen kehityssuunta sen tuoman merkittävän nopeuden vuoksi. Yhteen pelivirtaan kuuluvat tilanteet siltä väliltä, kun joukkue saa kiekon ja kiekko joko menetetään tai saadaan maali. Sovelluksella oli tarkoitus kerätä oman joukkueen tiedot tarkasti, mutta myös vastustajan pelitietoja hieman pienemmällä tarkkuudella. Sovellus oli tarkoitus tehdä Android-käyttöjärjestelmälle käyttäen Xamarin-sovelluskehitysalustaa ja sen käyttämää C#-ohjelmointikieltä.

Työn tuloksena on käyttökelpoinen sovellus, sen tarkoitukseen sopivalla käyttöliittymällä. Käyttöliittymän osalta kokeiltiin erilaisia ratkaisuja tiedon keräämiseen ja sen esittämiseen. Sovelluksen lopullisessa versiossa on käytetty sopivimpia ja helppokäyttöisimpiä ratkaisuita. Työssä on käsitelty myös muita mahdollisia ratkaisuita, joilla tietoa voitaisiin esittää ja kerätä, mutta nämä ratkaisut eivät ole olleet sopivia sovelluksen käyttöön. Niistä on luovuttu niiden toiminnan, tai sitten toimeksiantajan palautteen perusteella. Alkuperäiseen tiedonkeräämistapaan nähden sovellus on merkittävästi parempi tapa hoitaa samat asiat. Lisäksi sovelluksella saadaan automaattisesti kerättyä tietoa, mihin ennen olisi tarvinnut manuaalisesti kiinnittää huomiota.

Uutena tietona opinnäytetyöprosessissa on tullut erityisesti Xamarin-sovelluskehitysalustasta, mutta myös Androidista käyttöjärjestelmänä. Sen lisäksi uutta tietoa on tullut merkittävästi siitä, miten Xamarin-alustalla on toteutettu Android-sovellusten kehittäminen, verrattuna Androidin natiiveihin ratkaisuihin. Teknisen osaamisen lisäksi, uutta tietoa on tullut erityisesti käyttöliittymäsuunnittelusta, ja siitä, kuinka moneen asiaan tulee kiinnittää huomiota.

Käyttöliittymän osalta on monta eri tapaa ratkaista sama asia. Yksi ratkaisu saattaa sovelluskehittäjän näkökulmasta olla paras mahdollinen, mutta sovelluksen käyttäjän näkökulmasta, jokin toinen ratkaisu olisi luonnollisempi ja parempi tapa ratkaista asia. Sovelluksessa oltaisiin voitu käyttää monia erilaisia eleitä, mutta sovellusta kehittäessä nähtiin, että valitut ratkaisut olisivat käyttäjän kannalta yksinkertaisempia ja luonnollisempia. Tähän tulokseen päädyttiin myös yhdessä toimeksiantajan kanssa. Sovelluksen jatkokehityksen osalta tulen itse toimimaan kehittäjänä erityisesti virheitä korjaten. Muuta jatkokehittävää sovelluksessa on tietojen tarkastelu sovelluksen sisäisesti.

LÄHTEET

Android 2016a Viitattu 28.10.2016

<https://www.android.com/history/>

Android 2016b Viitattu 28.10.2016

<https://developer.android.com/design/index.html>

Android 2016c Viitattu 28.10.2016

<https://developer.android.com/training/multiscreen/index.html>

Android 2016d Viitattu 28.10.2016

<https://developer.android.com/guide/topics/resources/providing-resources.html>

Android 2016e Viitattu 28.10.2016

<https://developer.android.com/guide/topics/ui/ui-events.html>

Android 2016f Viitattu 28.10.2016

<https://developer.android.com/training/gestures/detector.html>

Android 2016g Viitattu 28.10.2016

<https://developer.android.com/training/gestures/movement.html>

Android 2016h Viitattu 28.10.2016

<https://developer.android.com/training/gestures/multi.html>

Android 2016i Viitattu 28.10.2016

<https://developer.android.com/guide/topics/ui/controls.html>

Android 2016j Viitattu 20.11.2016

<https://developer.android.com/guide/topics/resources/providing-resources.html>

Getting Real 2016

<https://gettingreal.37signals.com/>

Xamarin 2016a Viitattu 28.10.2016

<https://blogs.microsoft.com/blog/2016/02/24/microsoft-to-acquire-xamarin-and-empower-more-developers-to-build-apps-on-any-device/>

Xamarin 2016b Viitattu 28.10.2016

<https://www.xamarin.com/studio#designers>

Xamarin 2016c Viitattu 28.10.2016

<https://www.xamarin.com/visual-studio#designers>

Xamarin 2016d Viitattu 28.10.2016

<https://www.xamarin.com/platform>

Xamarin 2016e Viitattu 28.10.2016

https://developer.xamarin.com/guides/android/user_interface/designer_overview/