

Samuel Aitamaa, Lotta Lampola, Teemu Luhtanen, Veeti
Pere ja Ilkka Savela

Projektin loppuraportti

The Grand Myllypuro - virtuaalikasino

Sisällys

1	Johdanto	1
2	Visio	1
3	Lyhyt kuvaus sovelluksesta	1
4	Sovelluksen käyttöohje	2
4.1	Aloituskäyttö	2
4.2	Pelinäkymä	3
4.3	Rekisteröityminen ja sisäänkirjautuminen	4
4.4	Pelihistorian tarkastelu	6
4.5	Peliasetukset	7
5	Mitä toteutettu ja mitä vielä tulossa	8
5.1	Toteutetut toiminnot	8
5.2	Jatkokehitysideoita	9
6	Kuvaus arkkitehtuurista	9
6.1	Arkkitehtuuri	9
6.2	Tietokanta	10
7	Teknologiat, ympäristöt, versionhallinta ja testaus	12
7.1	Teknologiat ja kehitysympäristöt	12
7.2	Testaus	12
7.3	Versionhallinta	14
7.4	15	
8	Yhteenveto	15

1 Johdanto

Kevään projekti oli jaettu kahteen osaan: OTP1 ja OTP2 joiden aikana oli yhteensä seitsemän sprint-jaksoa. Ensimmäisen jakson aikana tutustuimme koontityökaluihin sekä -palvelimiin, ja kehitimme ensimmäisen version virtuaalikasinosovelluksesta.

OTP2-projektin tavoitteena oli jatkaa kehitystyötä siitä mihin alkukevään jälkeen olimme jääneet. Projektiryhmän itse suunnittelemien toiminnallisuuden lisäksi tavoitteena oli suorittaa ohjelmiston kansainvälistäminen ja lokalisointi sekä viimeistellä kaikki projektiin liittyvä dokumentaatio. Toisessa jaksossa suoritimme lokalisaation Iso-Britanniaan ja viimeistelimme pelin toiminnallisuuden, sekä kehitimme asetuksia ja nostimme testikattavuutta.

2 Visio

Ohjelmistotuotantoprojektin tavoitteena oli Scrum-työtapoja noudattavan projektityöskentelyn tuloksena kehittää työpöytäsovellus. Ryhmän visiona oli luoda virtuaalikasino, joka tarjoaa käyttäjälle mahdollisuuden tutustua uhkapeleihin ilman riskejä. Kasinosovellus tarjoaa käyttäjilleen ilmaiseksi virtuaalirahaa, jota sovelluksessa käytetään pelaamiseen. Kasinosovelluksen kehitys aloitettiin Blackjack-pelistä. Tavoitteenamme oli suunnitella helppokäyttöinen ja riskitön virtuaalikasino, joka soveltuisi etenkin kokemattomille pelaajille.

3 Lyhyt kuvaus sovelluksesta

Ohjelmointiprojektin aikana kehitettiin työpöytäsovellus The Grand Myllypuro-virtuaalikasino, jossa käyttäjä voi pelata Blackjack-peliä. Jokaiselle uudelle käyttäjälle tarjotaan kiinteä summa – 2500 \$:a virtuaalirahaa, jota voi käyttää uhkapelaamiseen. Pelaamisen lisäksi käyttäjä voi halutessaan kerrata Blackjack-pelin sääntöjä sekä muokata pelin käyttökokemusta peliasetuksia muokkaamalla. Asetuksista voi säätää

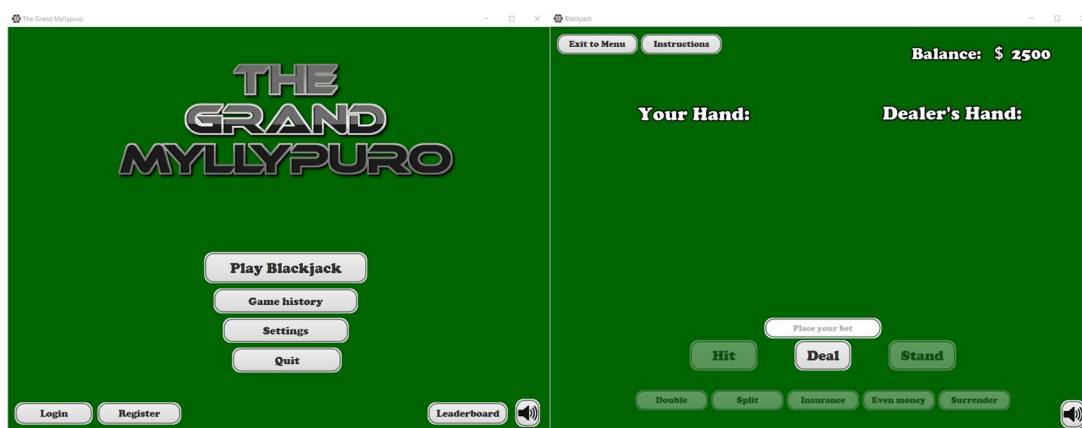
pelin äänenvoimakkuutta, halutessaan aktivoida pelin aikana saatavat pelivinkit ja vaihtaa kieltä suomen ja englannin välillä. Kilpailuhenkisille pelaajille sovellus tarjoaa myös mahdollisuuden tarkastella sovelluksen parhaimpien pelaajien pelimenestystä tulostaulusta.

Sovellus ei vaadi rekisteröitymistä, mutta halutessaan käyttäjä voi luoda tunnuksen ja pelata kirjautuneena käyttäjänä. Kirjautuneen käyttäjän pelitiedot tallennetaan ja täten käyttäjän rahasaldo pysyy tallessa seuraavaa pelikertaa varten. Rekisteröitynyt pelaaja voi myös halutessaan tarkastella pelihistoriaansa ja pystyy vaihtamaan salasanan.

4 Sovelluksen käyttöohje

4.1 Aloitusnäkymä

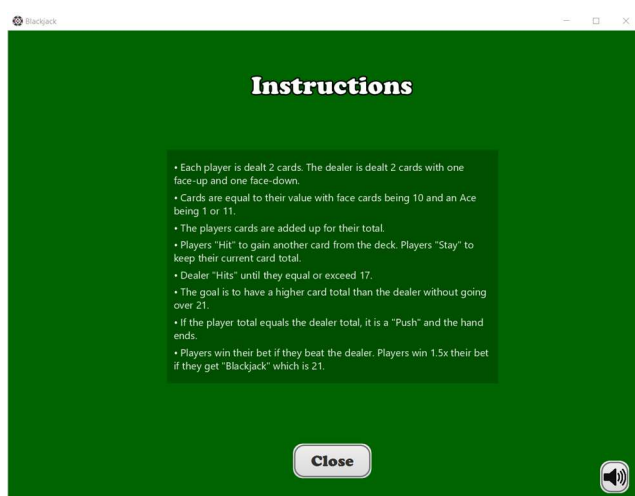
Sovelluksen aloitusnäkymän (kuva 1) kautta voi navigoida pelaamaan, tarkastelemaan pelihistoriaa, muuttamaan peliasetuksia sekä sulkea sovelluksen. Vasemman alalaidan kautta pääsee halutessaan joko rekisteröitymään tai kirjautumaan sisään. Oikean alalaidan kautta pääsee tulostaulukkoon sekä voi halutessaan mykistää äänet ja laittaa ne takaisin päälle.



Kuva 1. Sovelluksen aloitusnäkymä ja pelinäkymä aloittaessa

4.2 Pelinäkymä

Kuvassa 1 oikealla on esitetty pelinäkymä aluksi. Pelinäkymän vasemmasta ylälaidasta voidaan navigoida takaisin aloitusnäkymään. Exit to menu-valinta myös päättää mahdollisen käynnissä olevan pelin ja tämän vuoksi sovellus varmistaa käyttäjältä, haluaako hän varmasti poistua pelistä. Poistumisnapin vieressä oleva Instructions-toiminto ohjaa käyttäjän ohjesivulle. Ohjenäkymässä käynti ei keskeytä mahdollista käynnissä olevaa kierrosta. Kuvankaappaus ohjenäkymästä raportin kuvassa 2.



Kuva 2. Ohjenäkymä

Aloittaakseen pelin, käyttäjän tulee syöttää haluamansa panos ruudun Place your bet-kenttään. Panos tulee syöttää kokonaislukumuodossa ja olla positiivinen luku, eikä se saa ylittää pelaajan virtuaalirahasaldoa (balance). Virtuaalirahasaldon määrä on esitetty pelinäkymän oikeassa ylälaidassa. Mikäli käyttäjä ei syötä hyväksyttävää panosta, sovellus antaa käyttäjälle palautteen ja pyytää syöttämään hyväksyttävän panoksen.

Kuvassa 3 on esitetty tilanne, kun käyttäjä asettaa panokseksi 100 \$ ja tämän jälkeen käynnistää pelin. Peli käynnistetään painamalla Deal-nappia panoksen asettamisen jälkeen.



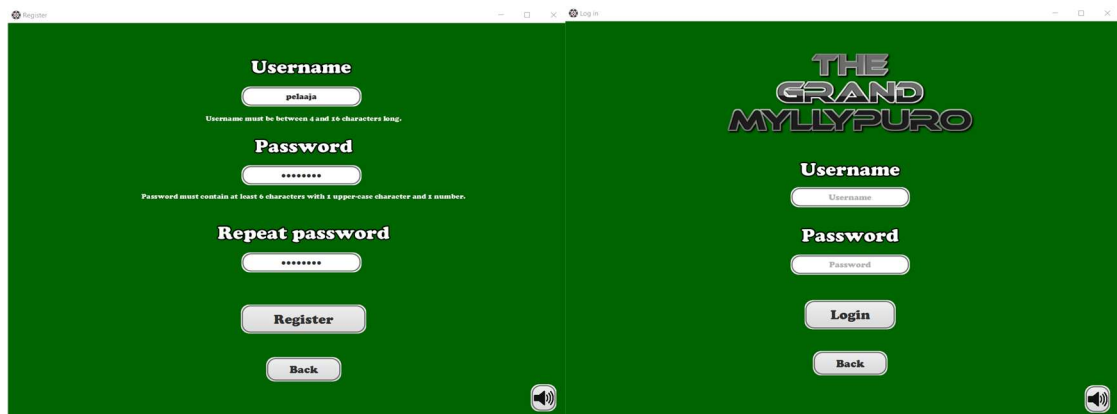
Kuva 3. Panoksen asettaminen ja näkymä pelin alettua

Pelin alkaessa pelaajan rahasalidon määrästä vähennetään hänen asettaman panoksen määrä ja pelaajalle sekä jakajalle jaetaan aloituskädet. Pelaajan käsi ja sen korttien yhteispistemäärä esitetään näkymän vasemmalla puolella.

Pelaajan käden korteista ja niiden pistemäärästä riippuen käyttöliittymässä esitetään käytettävissä olevat toiminnot ja pois päältä kytketyt toimintonapit esitetään himmennettynä (esim. Double tai Split-toiminto).

4.3 Rekisteröityminen ja sisäänkirjautuminen

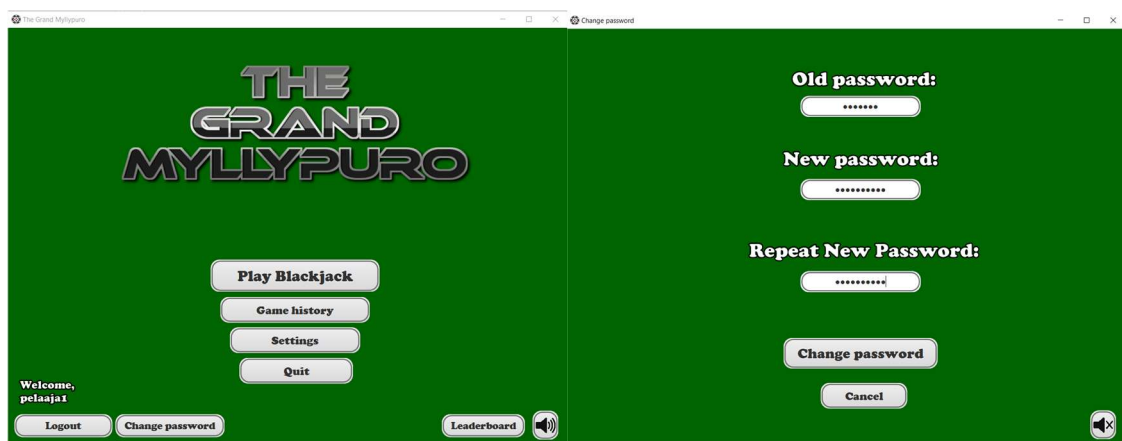
Kuvassa 4 vasemmalla esitetään rekisteröitymisnäkö, johon käyttäjä on syöttänyt haluamansa käyttäjänimen ja salasanan. Rekisteröityäkseen käyttäjän tulee syöttää sovelluksen muihin käyttäjiin nähden yksilöllinen käyttäjänimi, jonka pituus on 4-16 merkkiä. Käyttäjänimen lisäksi rekisteröityessä tulee asettaa salasana, joka on vähintään 6 merkkiä pitkä, sisältää pienen kirjaimen, sekä yhden ison kirjaimen että yhden numeromerkin. Varmistukseksi salasana tulee täyttää kahteen kertaan ennen rekisteröitymistä.



Kuva 4. Käyttäjäksi rekisteröityminen ja sisäänkirjautuminen

Register-nappia painamalla käyttäjä voi rekisteröityä. Mikäli jokin vaadittu tieto on täyttämättä tai tieto ei täytä rekisteröinnin validointiehtoja, käyttäjä saa tästä palautteen.

Kirjautuakseen sisään (kuva 4 oikea puoli) käyttäjän tulee syöttää käyttäjänimi sekä salasana. Log in-nappia painamalla kirjaudutaan sisään. Mikäli kirjautuminen ei onnistu, käyttäjä saa tästä palautteen ja kenttien tiedot nollautuvat. Mikäli kirjautuminen onnistuu, käyttäjä saa tästäkin palautteen ja käyttäjä siirretään aloitusnäkömään. Sisäänkirjautuneen käyttäjän aloitusnäkömää (kts. kuva 5) toivottaa pelaajan tervetulleeksi ja tarjoaa uloskirjautumismahdollisuuden. Lisäksi kirjautuneelle käyttäjälle tarjotaan mahdollisuus salasanan vaihtamiseksi.



Kuva 5. Sisäänkirjautuneen käyttäjän aloitusnäkömää ja salasanan vaihtaminen

4.4 Pelihistorian tarkastelu

Virtuaalikasinonsovellus tarjoaa käyttäjilleen mahdollisuuden tarkastella aiemmin pelattujen pelien tietoja. Mikäli käyttäjä ei ole rekisteröitynyt, pelihistoriatietoja ei tallenneta. Rekisteröityneiden käyttäjien tiedot taas tallennetaan ja he voivat tarkastella kaikkia pelitietojaan. Kuvassa 6 esitetään rekisteröityneen ja sisäänkirjautuneen pelihistorianäkymä.

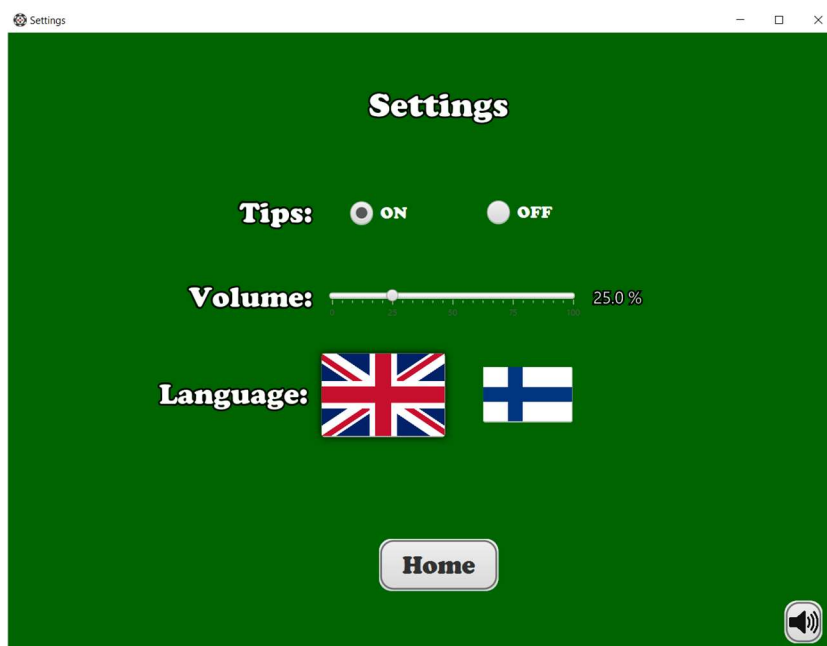
[illegible]

Kuva 6. Sisäänkirjautuneen käyttäjän pelihistorianäkymä

Pelihistoria esitetään taulukkona. Taulukosta voidaan nähdä pelikerran numerotunnus, pelitulos, syy pelitulokseen (esim. Blackjack, normaalivoitto, ylimeno), panoksen määrä, virtuaalirahasaldon määrä pelin päätyttyä sekä pelin aloitusajankohdan aikaleima.

4.5 Peliasetukset

Peliasetuksista käyttäjä voi säätää äänenvoimakkuutta, ottaa käyttöön tai pois käytöstä reaaliaikaiset pelivinkit sekä vaihtaa käyttöliittymän kieltä. Kielivaihtoehtoina on suomi sekä englantia. Asetusnäky on esitetty kuvassa 7.

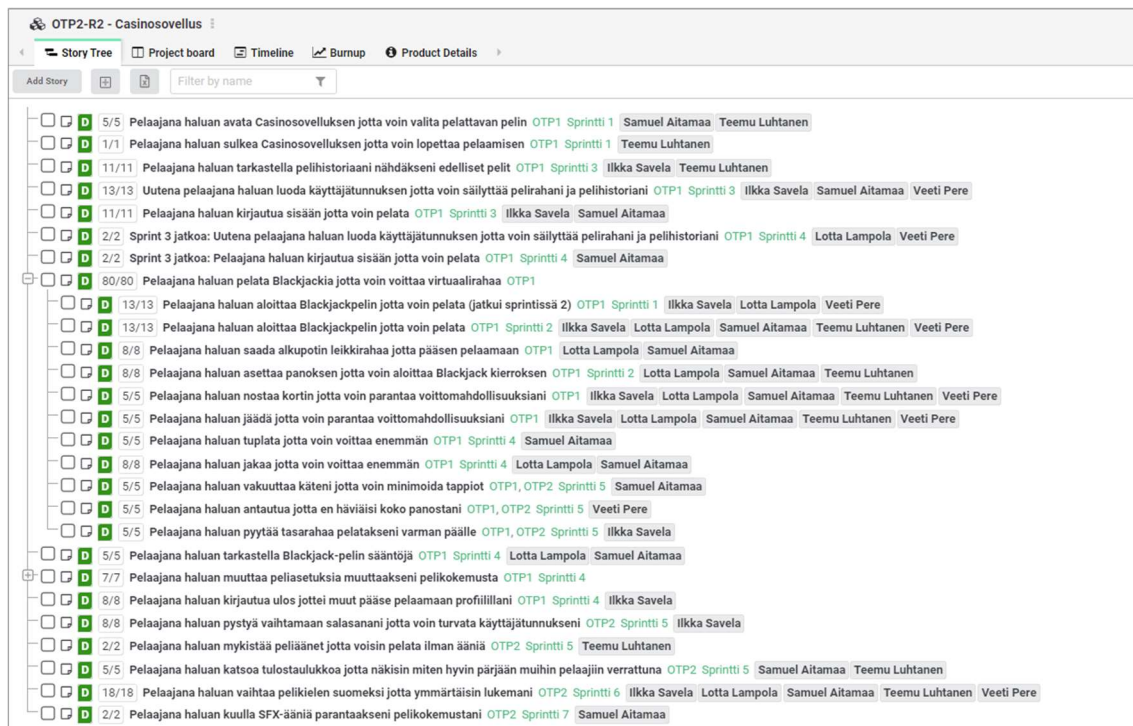


Kuva 7. Peliasetukset

Pelivinkit toimivat tooltipsien avulla. Pelinäkyssä pelaajalle näytetään tooltip, kun osoitinta pidetään painikkeen päällä. Vinkit toimivat melko yksinkertaisesti ja ne neuvovat pelaajaa aina samassa tilanteessa samalla tavalla. Esimerkiksi, jos korttien summa on alle 17, vinkit neuvovat nostamaan uuden kortin tai jos tuplaaminen on mahdollista, neuvoo vinkit tuplaamaan. Oletuksena pelivinkit ovat päällä. Volumeslider yksinkertaisesti säätää pelimusiikin äänenvoimakkuutta sen arvon mukaisesti. Oletuksena se on asetettu arvoon 25 %.

5 Mitä toteutettu ja mitä vielä tulossa

5.1 Toteutetut toiminnot



Kuva 8. Projektin käyttäjätarinoiden toteutustilanne Nektionissa 03.05.2021

Ohjelmistotuotantoprojekti 2:n aikana onnistuttiin viimeistelemään toimiva Blackjack-peli. Käyttäjätarinoiden tilanne projektin lopussa on nähtävissä kuvasta 8. Kaikki suunnitellut Blackjack-pelin erikoistilanteet saatiin toteutettua kevään kolmen viimeisen sprintin aikana. Sovelluksen käyttöliittymässä on mahdollista jäädä, nostaa uusi kortti, tuplata panos, jakaa käsi, vakuuttaa, ottaa tasaraha sekä antautua.

Visiomme oli toteuttaa virtuaalikasino, joka soveltuisi myös aloittelevien ja kokemattomien pelaajien käytettäväksi. Kevään aikana toteutimme ohjenäkymän sekä reaaliaikaiset pelivinkit, jotka tukevat uutta pelaajaa.

Koko sovelluksen käyttöliittymä saatiin myös kansainvälistettyä ja lokalisoitua kahdelle eri alueelle - Suomi ja Englanti (Iso-Britannia). Viimeistelimme englanninkielisen oletusversiomme ja sen lisäksi toteutimme käännöstuen sekä

suomenkieliselle että Iso-Britanniaan sopivalle englanninkieliselle versiolle. Kielenvaihto onnistuu käyttöliittymässä.

5.2 Jatkokehitysideoita

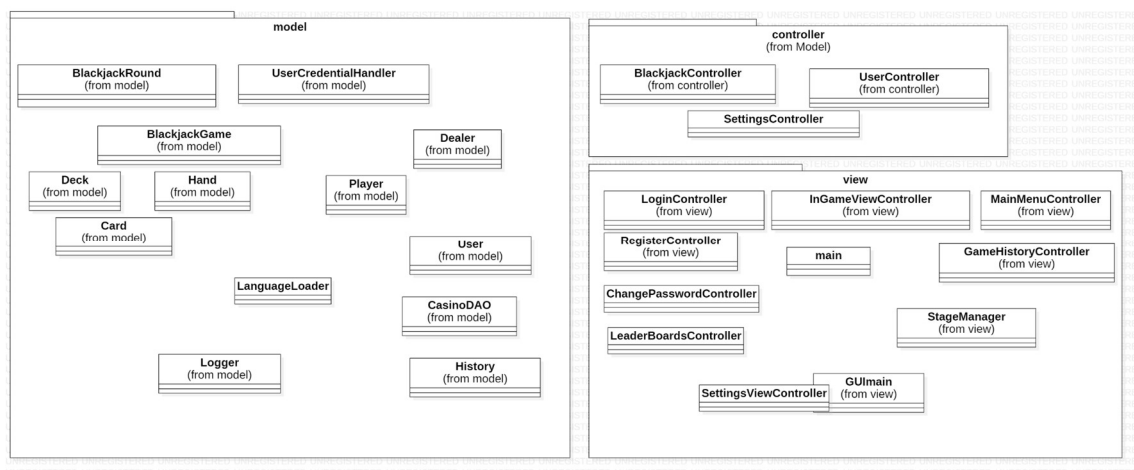
Kaikki Blackjack-peliin liittyvät toiminnallisuudet onnistuttiin toteuttamaan projektin aikana suunnitellusti. Projektiryhmällä on kuitenkin liuta ideoita sovelluksen jatkokehitystä varten.

Uusia toiminnallisuuksia voisi olla esimerkiksi unohtuneen salasanan vaihtaminen sähköpostin avulla, edistyneemmät pelivinkit, pelihistorian visualisointi ja täysin uudet kasinopelit (esim. ruletti tai pokeri). Saavutettavuutta ajatellen yksi kehitysidea voisi myös olla käyttöliittymän värien ja kontrastien säätämismahdollisuus.

6 Kuvaus arkkitehtuurista

6.1 Arkkitehtuuri

Ohjelmistomme arkkitehtuuri on toteutettu MVC-mallin mukaisesti. Lähdekoodi on jaettu kolmeen pakkaukseen; model, controller ja view. Sovellus koostuu reilusta kahdestakymmenestä luokasta. Luokkakaavio, jossa esitettynä pakkaukset ja luokat on kuvassa 9.



Kuva 9. Yksinkertaistettu luokkakaavio.

Kuvan selkeyttämiseksi luokkakaaviosta on jätetty kuvaamatta luokkien välinen kommunikaatio sekä metodit.

Mallitason pakkauksessa on toteutettu ohjelmiston sisäinen logiikka. Mallitasolla on niin korttipelin komponentteja kuvaavia kuin käyttäjähallintaan liittyviäkin luokkia. Model-pakkaukseen on myös sisällytetty tietokannan käsittelyyn tarkoitettu DAO-luokka. Tällä hetkellä kaikki tiedon tallentamiseen liittyvä toiminnallisuus on toteutettu yhdessä DAO-luokassa eli kaikille luokkaolioille ei ole siis tehty omaa vastaavaa DAO-oliota.

Controller-pakkauksessa on toteutettu kontrollerit, joilla on vastuu sovelluksen ohjauslogiikasta. Kontrollereita on kolme; yksi itse pelilogiikan ohjausta varten, toinen käyttäjähallintaa varten ja kolmas asetusten ohjausta varten.


View-pakkauksen luokat ovat myös ns. kontrollereita. Virtuaalikasinon käyttöliittymä on toteutettu JavaFX Scene Builderin avulla. Lähdekoodin luokissa on toteutettu eri näkymien sisältämät toiminnallisuudet.

6.2 Tietokanta

Käyttäjän tiedot, sekä pelihistoria tallentuvat Metropolian educloud-palvelimella sijaitsevaan MySQL tietokantaan. Tietokannan hallintatyökaluna käytimme


HeidiSQL:ää. Sovelluksen tietokannan käyttö on toteutettu Hibernate-kirjastoa hyödyntämällä. Yhteys tietokantaan on luotu JDBC-rajapinnan avulla.

Käyttäjän tiedot tallentuvat users-tauluun (kuva 10) käyttäjän rekisteröityessä. Tauluun tallennetaan käyttäjän id, pelisaldo, käyttäjänimi, sekä salasana PBKDF2 algoritmilla salattuna tietoturvan takaamiseksi.

 userID	balance	password	username
1	0	1000:d21bba34c8071f6ef34489c7eb924b7e:cc...	123
2	0	1000:7671ee238c62e1c01126fe19957de63b:0...	projekti
3	0	1000:9d405c4929142a26cce62c2b3fde203f:9f...	33
4	0	1000:ffae3cee97364acaa6f68322655d0477:5c...	testikayttaja

kuva 10. users taulu

Pelihistoria tallentuu history tauluun (kuva 11), jonne kirjataan pelinumero, käyttäjän pelin jälkeinen saldo, asetettu panos, aikaleima, metodi jolla peli päättyi, boolean arvona voittiko pelaaja, sekä pelanneen käyttäjän id. Mikäli pelaaja ei ollut kirjautunut millään käyttäjällä, asetetaan käyttäjän id:ksi 0.

 gameNumber	balance	bet	date	method	result	userID
1	21 000	1 000	2021-03-06 17:17:04	Blackjack	0	7
2	23 000	1 000	2021-03-06 17:17:08	Basic	0	7
3	21 000	1 000	2021-03-06 17:18:44	Blackjack	1	7
4	21 000	1 000	2021-03-06 17:18:48	Busted	1	7
5	2 500	100	2021-03-06 17:21:23	Blackjack	0	0
6	2 700	100	2021-03-06 17:21:27	Blackjack	0	0
7	2 700	100	2021-03-06 17:21:52	Basic	0	0
8	2 900	100	2021-03-06 17:21:56	Basic	0	0
9	2 800	100	2021-03-06 17:22:05	Busted	1	0
10	2 600	100	2021-03-06 17:22:24	Busted	1	0
11	2 800	100	2021-03-06 17:22:28	Basic	0	0
12	2 800	100	2021-03-06 17:22:44	Blackjack	0	0
13	2 800	100	2021-03-06 17:22:49	Busted	1	0
14	2 600	100	2021-03-06 17:23:04	Busted	1	0
15	2 600	100	2021-03-06 17:23:08	Busted	1	0

kuva 11. History taulu

7 Teknologiat, ympäristöt, versionhallinta ja testaus

7.1 Teknologiat ja kehitysympäristöt

Kaikki projektin jäsenet käyttivät kehitysympäristönään JetBrainsin IntelliJ IDEA-ympäristöä. Itse sovellus on kehitetty käyttäen Javaa ja käyttöliittymä on rakennettu JavaFX-kirjaston avulla.

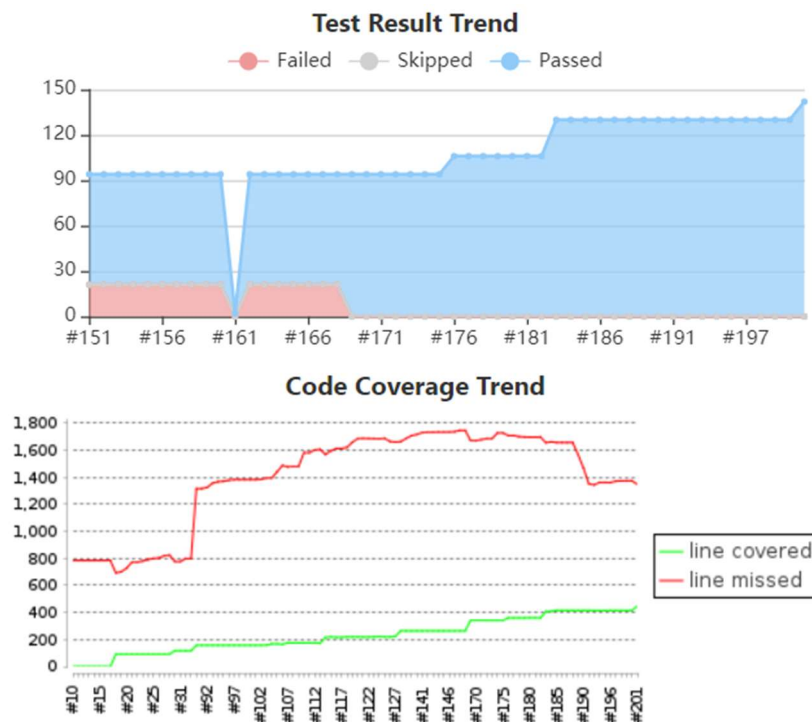
Kehitystyömme koontityökaluna käytimme Mavenia ja koonti- ja automaatiopalvelimena käytimme Jenkinsiä. Koska sovellusta ei todellisuudessa julkaista missään, oli Jenkins projektissamme lähinnä harjoitusmielessä.

Tutustuimme Jenkinsiin ja Maveniin yhdessä sekä suoritimme käyttöönoton yhteispuhelussa, jotta kaikki pääsisivät tutustumaan näihin uusiin työkaluihin.

7.2 Testaus

Projekti aloitettiin mallitason pelilogiikan kehityksestä. Emme käyttäneet TDD-mallia kehitystyössämme, ja täten osa testeistä tehtiin vasta kun itse toiminnallisuus oltiin jo toteutettu.

Muutama viikko OTP1-projektin aloittamisen jälkeen ryhdyimme tekemään yksikkötestejä, ja JUnit-testejä kertyikin lopulta n. 150. Kuvasta 12 voidaan nähdä ylemmästä grafiikasta yksikkötestien määrä ja alemmasta grafiikasta testikattavuus. Loppuviikkojen refaktoroinnin myötä saimme poistettua duplikaattikoodia UI-tasolta joka paransi testikattavuutta.



Kuva 12. Testien lukumäärä ja testikattavuus.

Mallitason luokille pyrittiin tekemään yksikkötestit, joiden kattavuus oli täydellinen. Kuvasta 13 voidaan mallitason testikattavuus luokittain.



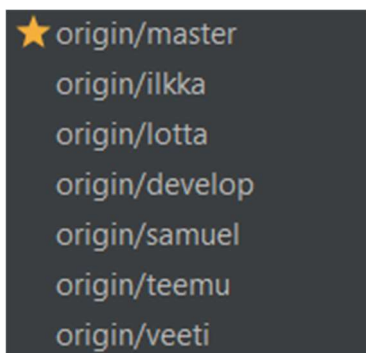
Kuva 13. Testikattavuus mallitasolla luokittain

DAO-luokkien (CasinoDAO ja History) testausta ei koettu kriittiseksi, joten näihin testeihin ei käytetty suurempia resursseja. Pelilogiikan kokoavien luokkien (BlackJackGame ja BlackJackRound) osalta huomattiin, ettemme olleet noudattaneet MVC-mallia täydellisesti. Kontrollerien käyttäytyminen hankaloitti yksikkötestien tekemistä ja täten näiden kokoavien luokkien testikattavuus jäi myös suunniteltua vajaammaksi. Luokat kuitenkin käyttävät suurimmin osin muiden luokkien metodeja joten niiden toiminta on osittain varmistettu muiden luokkien testauksen myötä.

Käyttöliittymää ja siihen suoraan liittyvää koodia testasimme sinnikkäästi käyttöliittymää testaamalla. Mietimme testitapauksia ja yritimme saada aikaan virheitä, jotta huomioisimme käyttöliittymää rakentaessa kaikki mahdolliset käyttäjän tekemät virheet ja niiden aiheuttamat vikatilanteet.

7.3 Versionhallinta

Projektin versionhallintaan käytettiin Git-järjestelmää. Versionhallinnan haarat on esitetty kuvassa 11.



Kuva 11. Versionhallinnan haarat

Jokaisella Scrum-työryhmän jäsenellä oli oma haara, jossa kehitystyötä tehtiin. Toimivaksi todetut toiminnallisuudet vietiin aina yhteiseen develop-haaraan. Projektin aikana ei tarkemmin sovittu yhtenäistä kommitointitapaa. Jokainen ryhmän jäsen vei tuotoksensa yhteiseen kehityshaaraan parhaaksi näkemissään

palasissa. Master-haaraan kommitoinnit tehtiin yleensä aina sprintin päätteeksi koko ryhmän yhteisellä päätöksellä.

8 Yhteenveto

Kahdeksan viikon aikana toteutimme Scrum työtapoja noudattaen virtuaalikasinon, jossa on mahdollista pelata Blackjackia. Toteutimme sovelluksen viiden hengen ryhmässä Javalla ja sen kirjastolla JavaFX:llä MVC-arkkitehtuurin mukaisesti. Yksikkötesteissä hyödynsimme JUnit ohjelmistokehystä, käännösaubomaatio- ja koontityökaluina Jenkinsiä ja Mavenia. Sovelluksen ideana on tarjota aloittelijaystävällinen ja helppokäyttöinen Blackjack kokemus ilman riskejä. Projektin tavoitteena oli tutustua Scrum työtapoihin, uusiin kehitystyökaluihin ja lokalisointiin, sekä kehittää yksilöllistä ohjelmointiosaamista ja ryhmätyöskentelytaitoja.

Sovelluksen rakenne koostuu yhdeksästä näkymästä: rekisteröityminen, sisäänkirjautuminen, salasanan vaihto, päävalikko, pelinäkömä, ohjeet, pelihistoria, tulostaulu ja asetukset. Sovellukseen on mahdollista rekisteröityä, jotta pelivaluutta ja -historia tallentuvat tarkastelua varten. Tietojen tallennus on toteutettu Metropolian palvelimella sijaitsevalle SQL tietokannalla ja sen kanssa kommunikointi Hibernatella. Pelaajalla on myös mahdollisuus kytkeä päälle pelivinkit, jotka antavat pelistrategisia neuvoja tilanteesta riippuen. Myös kieli on mahdollista vaihtaa englannin ja suomen välillä, joka kattaa koko sovelluksen. Ääniasetuksia on muutettavissa sekä asetuksissa sijaitsevasta slideristä että jokaisessa näkymässä olevasta Mute-napista.

Kevään projektityö oli jaettu kahteen osaan: OTP1 sekä OTP2, jotka kummatkin vielä neljään ja kolmeen noin kahden viikon pituiseen sprinttiin. Ensimmäisen jakson aikana otimme kehitystyökalut käyttöömme ja rakensimme ohjelman rungon ja veimme sen lähes valmiin tuotteen tasolle. Toisessa jaksossa suoritimme lokalisaation Iso-Britanniaan ja viimeistelimme pelin toiminnallisuuden, sekä kehitimme asetuksia ja nostimme testikattavuutta.