



Golf kierroskortti sovellus

Harjoitustyö

Jani Ervalahti AC7686, TTV22SM

Harjoitustyö

Full stack -ohjelmointi, Pasi Manninen

6.12.2022

Tieto- ja viestintätekniikka, Informaatio teknologia

Sisältö

1	Tehtävän kuvaus.....	1
2	Sovelluksen esittely	1
3	Käytännön toteutus	2
4	Ajankäyttö.....	6
5	Kehityskohteet	6
6	Itsearviointi	6
	Lähteet	7

1 Tehtävän kuvaus

Harjoitustyössä rakennetaan opintojakson aihealueeseen liittyvä Full Stack -sovellus. Harjoitustyöprosessissa opiskelija oppii, soveltaa ja osoittaa opintojakson aihealueen hallintaa laajemmin ja syvemmin kuin pelkästään harjoitustehtävien avulla. Parhaimmillaan tämä näkyy sellaisten ratkaisujen soveltamisessa, jotka eivät ole suoraan nähtävissä opintojaksolla tarjotuista materiaaleista. Harjoitustyön suunnittelu- ja tekoprosessi sekä tuotos dokumentoidaan HTML- tai PDF-muotoisella harjoitustyödokumentaatiolla. Huom. myös mahdollisessa git- repositoriossasi tarjolla oleva ja markdownilla tuotettu dokumentaatio on ok. Harjoitustyö tehdään optimissaan **1-3** hengen ryhmässä. Poikkeava ryhmäkoko on hyväksyttävä ennen työn aloitusta. Jokaisen työntekijän tekemä työ on dokumentoitava selvästi. Harjoitustyön aihe voidaan valita melko vapaasti, mutta aihe kannattaa tarkistuttaa opintojakson opettajalla ennen aloitusta. Aihe on helpointa tarkistuttaa laatimalla alustava suunnitelma (**) ja esittämällä se opettajalle. Työ esitellään **harjoitustyöseminaarissa** (=harjoitustöiden esittely) erikseen sovittavan aikataulun mukaan. Esityksen voi pitää myös videoesittelynä. Verkko- ja monimuotototeutuksissa esittely suoritetaan aina videoesittelynä. Videoesittelyn pituus on 3-10min. Esitys ei ole pakollinen, mutta se vaikuttaa arviointiin. Harjoitustyötä työestetään omalla ajalla. Lähiopetusjaksoja sisältävissä toteutuksissa ohjausta voi saada myös lähiopetuksen yhteydessä. Työn lähdekoodi ei voi olla salaista, vaan se pitää toimittaa työn loputtua dokumentaation mukana. Pyydetessä opiskelijan tulee pystyä kertomaan harjoitustyössä käytettyjen ratkaisujen keskeisestä teknisestä toimintalogiikasta esim. (video)puhelun välityksellä. Harjoitustyö kaikkineen tulee olla saatavilla siihen asti, kunnes opintojakson suoritus on virallisesti hyväksytty ja kirjattu opintotietojärjestelmään.

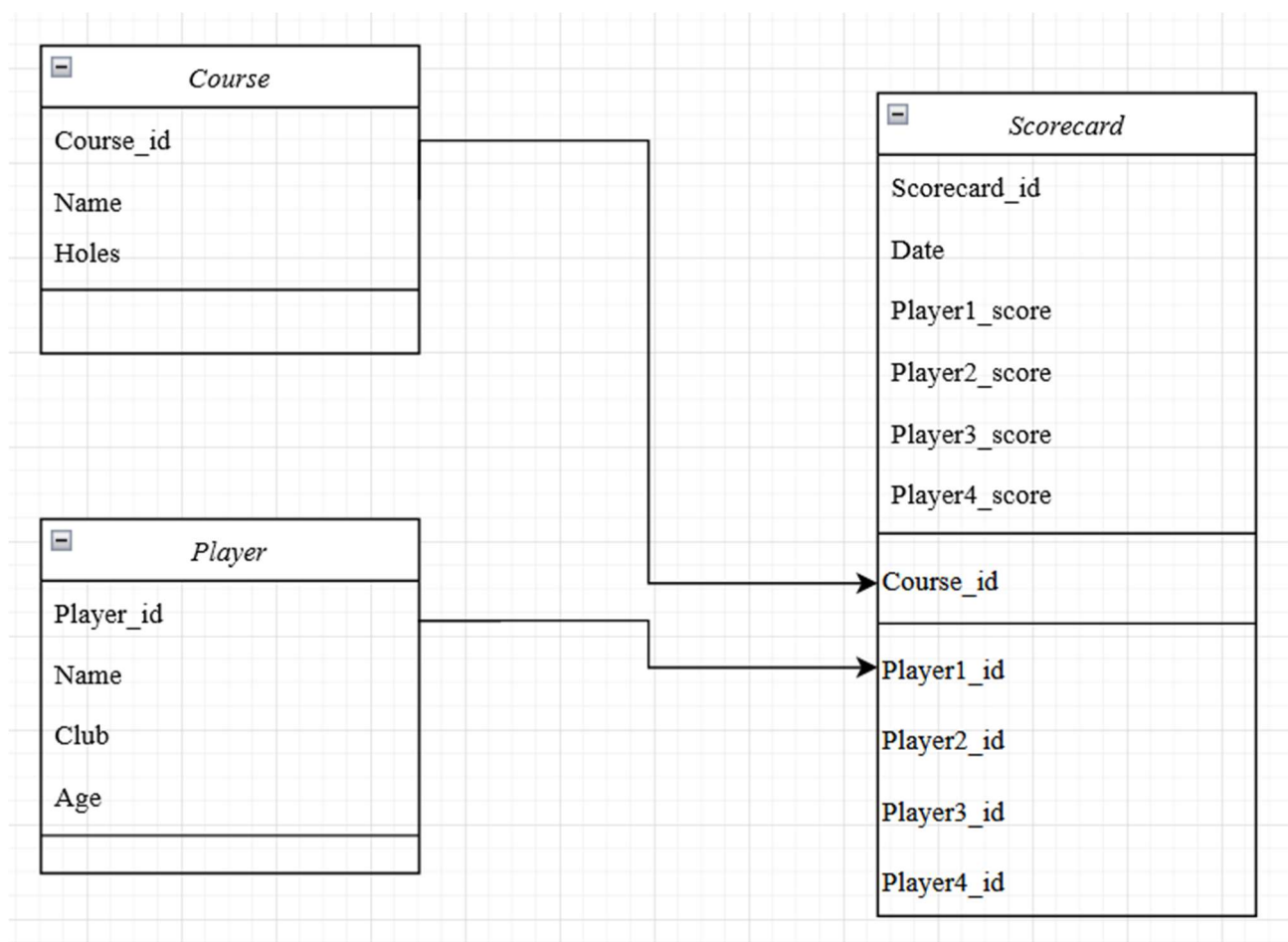
2 Sovelluksen esittely

Työksi valitsin golf kierroskortti sovelluksen. Sovelluksen perusidea on seuraavanlainen. Käyttäjä pystyy lisäämään, poistamaan ja muokkaamaan pelaajia. Pelaajat haetaan tietokannasta ja edellä mainitut toiminnallisuudet muokkaavat tietokannan dataa. Samat toiminnot pätevät ratoihin. Ratat sijaitsevat tietokannassa omassa kokoelmassaan. Kun käyttäjä on lisännyt pelaajia ja ratoja voi hän näitä nyt hyödyntää tulokortin luonnissa. Tulokortille käyttäjä valitsee radan, jonka haluaa pelata. Tämän jälkeen käyttäjä lisää kortille pelaajat, jotka pelaavat kierroksen. Kunkin pelaajan

tulososioon kirjataan tulokset. Kierroksen päätyttyä painetaan tallenna kierroskortti -painiketta ja kierroskortti tallentuu tietokantaan.

3 Käytännön toteutus

Työ lähti liikkeelle suunnittelusta. Suunnittelussa liikkeelle lähdettiin tietokannan luokkarakenteen suunnittelusta (Kuvio 1).



Kuvio 1. Tietokannan luokkarakenne

Luokkarakenteen suunnittelun jälkeen, suunnittelin layoutin sovellukselle. Alkuperäinen layout (Kuvio 2) ei valitettavasti tällä aikataululla ollut toteutettavissa, joten se muuttui lopulliseen versioon (Kuvio 3) nähden aika paljon.

Logo ja aplikaation nimi	Käyttäjän tiedot
Uusi kortti, jota tehdessä luodaan tuloskortille väylien määrä ja pelaajat	
Viimeisin tuloskortti	X
Toisiksi viimeisin	X
	X
	X
	X
Tuloskortit jatkuvat tästä alaspäin scrollilla, mikäli kyseisellä käyttäjällä on pelattuna tarpeeksi tuloskortteja	

PELATTAVA RATA	Käyttäjän tiedot
Väylä PAR lukema (muokattavissa)	Pelaajien tulokset
Väylä PAR lukema (muokattavissa)	Pelaajien tulokset
Väylä PAR lukema (muokattavissa)	Pelaajien tulokset
Väylä PAR lukema (muokattavissa)	Pelaajien tulokset
Väylä PAR lukema (muokattavissa)	Pelaajien tulokset
Väylä PAR lukema (muokattavissa)	Pelaajien tulokset
Väylä PAR lukema (muokattavissa)	Pelaajien tulokset
Väylä PAR lukema (muokattavissa)	Pelaajien tulokset
Väylä PAR lukema (muokattavissa)	Pelaajien tulokset

Kuvio 2. Sovelluksen alkuperäinen layout-suunnitelma

Pelaajat

Nimi:

Seura:

Ikä:

Lisää

Ladataan pelaajia... ▾

Muokkaa

Poista

Radat

Lisää

Ladataan ratoja... ▾

Muokkaa

Poista

Väylä	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
PAR																		

Uusi kierroskortti

Valitse kierroskortti ▾

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

PAR																		
Valitse Pelaaja ▾																		
Valitse Pelaaja ▾																		
Valitse Pelaaja ▾																		
Valitse Pelaaja ▾																		

Tallenna kortti

Ladataan kierroskortteja... ▾

Muokkaa

Poista

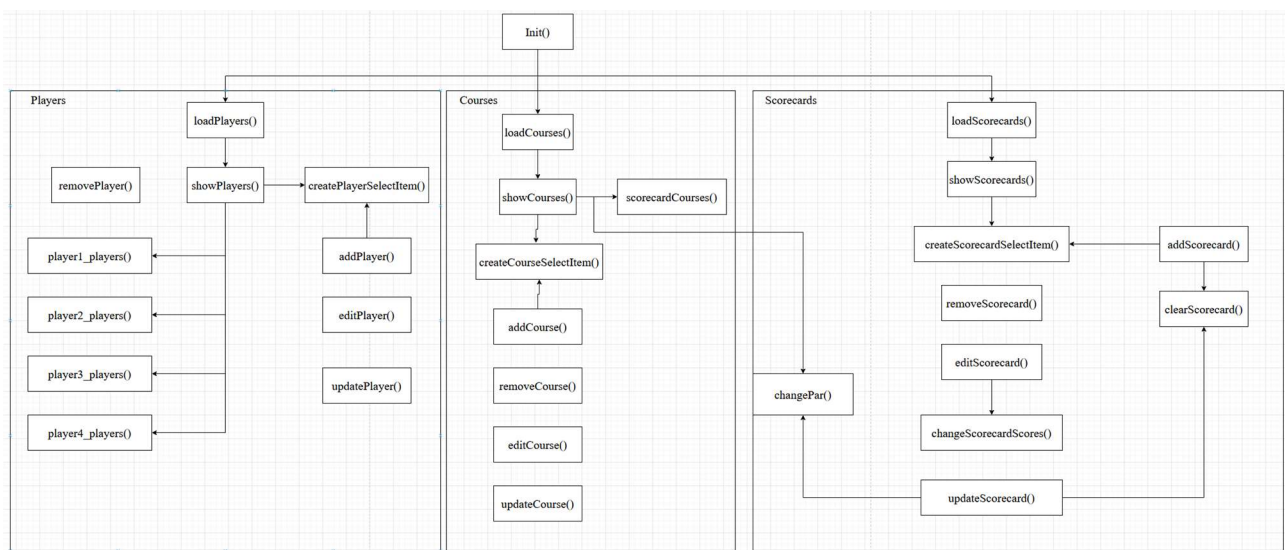
Kuvio 3. Lopullinen layout

Layoutin suunnittelun jälkeen lähdettiin sitten rakentamaan backendille ensiksi endpointit Pelaajat osiolle. Lisää-painike kutsuu funktiota `addPlayer`, joka nimensä mukaisesti lisää pelaajan POST metodilla käyttöliittymältä serverille ja serveri välittää tiedon tietokantaan. Tietokannasta haetaan pelaajat select elementtiin GET metodilla. Muokkaa-painike hakee select-elementissä olevan pelaajan tiedot input-kenttiin ja lukitsee select-elementin arvon. Myös Lisää-painike, muuttuu Tallenna-painikkeeksi ja taustaväriiltään keltaiseksi. Käyttäjän muutettua haluamansa muutokset ja

Tallenna-painiketta painettuaan tiedot viedään UPDATE metodilla tietokantaan kyseiselle pelaajalle. Poista painike poistaa select-elementin valitun pelaajan DELETE metodilla. Ratojen osalata logiikka on täysin sama.

Pelaajakortille tuodaan yllä oleva data valitavaksi. Pelaaja valitsee ensimmäisenä radan jota haluaa pelata, valitsee pelaajat jotka kierrokselle haluaa. Kierroksen edetessä pelaajille annetaan kunkin väylän mukainen tulos löyntien perusteella. Kun kierros on valmis ja kaikki väylät pelattu painetaan ”Tallenna kortti” -painiketta jolloin tieto viedään kantaan POST metodilla. Kierroskortit listataan GET metodilla haettuna select-elementille. Muokkaa painike tuo valitun tulokortin arvot muokattavaksi tulokortille. Muutettu tulokortti tallennetaan kun käyttäjä painaa ”Tallenna kortti” painiketta. Poista-painike poistaa tulokortin tietokannasta DELETE metodilla.

Tarkemmat funktioiden suhteet kuvataan Kuviossa 4.



Kuvio 4. Funktioiden suhteet

4 Ajankäyttö

Aikaa työhön meni odotettua enemmän. Aikaa yhteensä harjoitustyöhön kului n. 50 tuntia. Mikäli vain aikaa olisi ollut itsellä enemmän niin työhön olisi saanut kulumaan vieläkin enemmän aikaa, se jopa hieman vei mennessään.

5 Kehityskohteet

Kehityskohteita rupesi satelemaan itselle ovista ja ikkunoista, mitä pidemmälle työtä sai tehtyä. Tietokannan luokkarakenne voisi olla sivistyneempi. Lähdekoodissa toistuvia hyvin samankaltaisia funktioita olisi voinut uudelleen käyttää järkevämmällä parametroinilla. Syötteiden tarkistus puuttuu. Käyttöliittymän layout ei loppujen lopuksi ollut kovinkaan suunniteltu, joten siihen olisi voinut kiinnittää huomiota uudelleen matkan varrella. Käyttöliittymä on muutenkin varsin pelkistetty ja käyttäjäkokemuksen puolesta ei varsin ihmeellinen. Kehityskohteita siis löytyy varsin paljon jokaiselta osa-alueelta sovellusta. Näiden esille tuominen mahdollistaa myöhemmin sovelluskehityksessä jo suunnitteluvaiheessa niiden huomioon ottamisen ja paremman lopputuloksen.

6 Itsearviointi

Vaikka kehityskohteita löytyykin paljon, ei työ kuitenkaan huono mielestäni ole. Työ on laaja, mutta sen verran kriittinen olen, että antaisin itselleni 1,75/2 pistettä.

Lähteet

Github linkki sovelluksen lähdekoodiin ja dokumentaatioon. <https://github.com/j3rvis/golf>