Arttu-Pekka Konttila, Kristofer Maripuu, Riku Lind, Kelil Ghaith

Ohjelmistotuotanto
F1 Tilastopalvelu

CENTRIA-AMMATTIKORKEAKOULU
Marraskuu 2024

SISÄLLYS

1 JOHDANTO

Trello otettiin käyttöön, jossa listasimme roolituksen kuten Riskianalyytikon, Projektipäällikön, Tuotepäällikön ja Suunnittelijan. Käytämme yhteyden pitoon discordissa tehtyä ryhmää. Tavoitteena on saada projekti valmiiksi vuoden sisällä. Tavoitteena on tehdä Formula 1 tulospalvelu, josta pystytään näkemään F1 Kisaviikonloppujen eri sessioiden tulokset. Tehtävinä on saada erikseen Kisan, Aika-ajojen ja harjoituksien tulokset omille tauluilleen kyseisessä palvelussa. Käytämme työtavaksi SCRUM työtapaa eli neljän viikon työjaksoissa, jonka jälkeen tarkistetaan siihen asti tehty työ. Jos työssä huomataan virheitä, se korjataan tai tehdään uudestaan. Lisäksi tulospalvelu tarjoaa käyttäjille mahdollisuuden katsoa kisojen tuloksia helposti ja verrata kilpailijoiden suorituksia reaaliajassa. Palvelussa voi myös olla erillisiä näkymiä eri käyttäjäryhmille, kuten medialle, tiimeille ja faneille, jolloin jokainen ryhmä saa itselleen tärkeä tietoa.

2 PROJEKTISUUNNITELMA JA ROOLITUS

Trello

Otimme käyttöön Trellon, joka on todella kätevä suunnittelussa ja tehtävien jaossa. Sen avulla näemme myös mitä muut tiimin jäsenet ovat tehneet ja missä vaiheessa he ovat. Projektin alussa kävimme läpi Trellon ominaisuudet ja sen, miten sitä käytetään. Joka viikko käymme läpi jokaisen projektin työtehtävät ja varmistamme, että kaikki ymmärtävät oman vastuunsa. Aina kun lisäämme jotain Trelloon, ilmoitamme siitä muille ryhmän jäsenille ja kerromme mahdollisista ongelmista tai haasteista
https://trello.com/b/oKcb9pMV/it00ak37-3003-ohjelmistotuotanto

Roolitus

Valitsimme roolit sen perusteella, kuka koki olevansa pätevä tietyllä alueella. Keskustelimme kuka haluaa minkäkin roolin. Jos useampi henkilö halusi saman roolin, teimme kompromisseja, jotta kaikki voivat osallistua ja kehittyä omilla alueilla. Roolien ja vastuiden jakaminen pyrittiin tekemään reiluksi. Jos jollain on helpompi tehtävä, hän auttaa toista, jolla on vaikeampi tehtävä. Tämä ei poista vastuuta, vaan jokaisella on edelleen oma roolinsa ja vastuunsa. Apua voi pyytää, mutta lopulliset päätökset tekee henkilö, jolle tehtävä annettiin. Jos rooleissa tapahtuu muutoksia, asiasta ilmoitetaan ryhmälle ja yhdessä mietitään vaihtoehtoisia tapoja. On tärkeää olla empaattinen ja auttaa toisia jos he kohtaavat ongelmia. Rooleiksi valittiin Arttu-Pekka Konttila projektipäälliköksi, Ghaith Kelil vaatimusmäärittelijäksi, Riku Lind riskianalyytikoksi ja Kristofer Maripuu suunnittelijaksi.

Prosessimalli

Valitsimme prosessimalliksi Scrumin, koska sitä käytetään laajasti projektinhallinnassa. Samalla tutustumme siihen, miten Scrumi toimii. Aluksi suunnittelemme tehtävät suunnittelupalaverissa ja tehtävät jaemme tiimin kesken, joita pyrimme saavuttamaan ne sprintin aikana. Scrumin vahvuuksia ovat lyhyet, 15 – 30 minuutin mittaiset palaverit, jotka pidämme joka toinen päivä. Näin pysymme ajan tasalla projektin edistymisestä. Projektissamme käytämme kahden viikon sprinttejä, jotka mahdollistavat tehokkaan työn. Kun työt on saatu valmiiksi, tarkistamme että lopputulos vastaa tehtävänantoa. Kun annettu tehtävä on valmis, merkitsemme sen Trelloon “done” kohtaan osoittaaksemme, että se on suoritettu.

Tavoitteet

Projektin päätavoite on saada kurssin loppuun mennessä demoversion valmiiksi. Aina kun tehtävä jaetaan, sovitaan kuka sen tekee ja milloin sen tulisi olla valmis. Henkilö, jolle tehtävä on annettu, ilmoittaa kun se on valmis. Silloin saamme käsityksen projektin etenemisestä.

Tehtävät ja aikataulut

Tehtävät jaetaan yleensä pienempiin osiin, jotta ne eivät tuntuisi liian vaikeilta. Jokaisen tehtävän tulee suorittaa viikon kuluessa. Tehtävien jakaminen pieniin osiin vähentää työmäärän tuntua.

Riskianalyysi

- Tekijänoikeus ja lisensointi
    - Täytyy hankkia lisenssi virallisten tietojen, logojen ja datan käyttämistä varten. Lisenssi tulisi hankkia joltain tekijänoikeuksien haltijalta, kuten Formula One

Group (FOG). Ilman lisenssiä on lähes mahdotonta pitää yllä F1-tulospalvelua ilman, että siitä joutuu oikeustoimiin.

- Tietosuojalainsäädäntö
    - Sivustolla kerättävien käyttäjätietojen on noudatettava Euroopassa käytössä olevaa tietosuojalainsäädäntöä (GDPR). Käytännössä tarkoittaa sitä, että tietojen täytyy olla suojattuna ja käyttäjille täytyy ilmoittaa niiden asianmukaisesta käytöstä. Tämä on välttämätön vaihe tulospalvelun ylläpitämistä varten.
- Datan luotettavuus ja saatavuus
    - Tulospalvelun datan (esim. kierrosajat ja sijoitukset) täytyy olla luotettavaa, nopeaa ja reaaliaikaista. Tästä syystä täytyy käyttää vain luotettavia ja nopeita tietoverkkoja, joilla on alhainen latenssi. Oikeiden tiedonsiirtoprotokollien käyttäminen vaikuttaa merkittävästi datan siirtymisen nopeuteen ja eheyteen. Riskinä on käyttäjien menettäminen kilpailijoille.
- Palvelimen kapasiteetti ja skaalautuvuus
    - Täytyy olettaa, että ainakin kisojen aikana sivusto tulee olemaan ison kuorman alaisena. Tätä varten palvelinten kapasiteetti täytyy olla tarpeeksi iso, jotta sivusto ei hidastu tai kaadu kovassa käytössä. Pilvipalvelut olisivat paras vaihtoehto tässä vaiheessa, koska ne ovat tunnettuja niiden nopeasta skaalautuvuudesta ja kyvystä käsitellä suuria määriä dataa reaaliaikaisesti.
- Sivuston reaaliaikainen päivitys
    - Sivuston täytyy päivittää kierrosajat ja muut tulokset reaaliaikaisesti, jotta siitä olisi hyötyä käyttäjille. Ajax-menetelmä kuulostaisi aika hyvälle tässä tapauksessa, sillä suurin osa käyttäjistä ei vahdi sivustoa jatkuvasti, vaan käy vain vilkaisemassa tuloksia. Toisaalta Websocket saattaisi toimia paremmin muiden teknologioiden kanssa, sillä Ajax ei välttämättä ole enää kovin nykyaikainen vaihtoehto.
- Käyttäjäkokemus sivustolla

  - Sivuston täytyy olla helppokäyttöinen ja helposti navigoitavissa. Tärkeää olisi myös sivuston käytettävyys eri laitteilla. Eli sen tulisi toimia moitteetta tietokoneella, mobiililaitteilla ja tableteilla. Sivustolla tulee olemaan paljon kilpailua, joten sille olisi hyvä keksiä ominaisuuksia, joilla se erottuisi muista tulospalveluista.

- Liiketoiminta
  - Riippuen siitä, pidetäänkö sivustoa yllä mainostuloilla vai onko se maksullinen, saatetaan törmätä melko vaikeaan esteeseen. F1-kisojen seuraajat voivat olla jo sitoutuneita käyttämään muita tulospalveluja, joka vaikeuttaisi huomattavasti tulospalvelun ylläpitämistä.

- Tietoturva
  - Koska F1-kisat ovat erittäin suosittuja ja niitä katsoo todella moni, ovat niihin liittyvät sivustot kuten tulospalvelut varmasti jonkin sortin kyberhyökkäysten kohteita. Hakkerointiyritykset ja DDoS-hyökkäykset ovat erittäin todennäköisiä riskejä, ja niihin varautuminen on prioriteettina todella korkealla. Käyttäjien ja sivuston välinen viestintä tulee olla salattuna SSL-salauksella tai jollain muulla menetelmällä, jotta varmistetaan sivuston turvallinen käyttö.

- Rahoitus ja budjetti
  - Reaaliaikaisen tulospalvelun ylläpitäminen suuren käyttäjämäärän kanssa on kallista. Data, pilvipalveluiden palvelinkapasiteetti ja sivuston kehittäminen itsessään kustantavat paljon. Riippuen myös palkattavan henkilöstön kokemuksesta ja osaamisesta se saattaa koitua isoksi investoinniksi.

Toimintatavat esim. Yhteydenpito

Olemme luoneet Discord ryhmän, jossa kaikki ryhmän jäsenet ovat mukana. Jos on jotain tärkeää, ilmoitamme siitä Discordissa ja kerromme projektin etenemisestä. Jos joku on poissa, pidämme lyhyen kokouksen silloin, kun se sopii kaikille ja siellä keskustellaan myös jos on haasteita.

3 VAATIMUKSET

Toiminnalliset / ei-toiminnalliset vaatimukset jaoteltu

Sivustollamme on hakutoimisto, joka löytyy heti ylhäältä ja vasemmalla puolella sijaitsee navigointipalkki. Navigointipalkki helpottaa sivustolla liikkumista. Sivustolta löytyy myös live Leaderboard. Se näyttää kuljettajien sijoitukset reaaliajassa. Sector Times ja Mini Sectors näyttävät väreillä merkitetyt kohdat esimerkiksi liila, vihreä ja keltainen. Väreillä voit nähdä kuljettajan parantuneen kierrosajan. Sivustolta voit myös seurata renkaiden käyttöä ja pit stoppeja(box). Samalla näet mitkä renkaat heillä ovat käytössä esimerkiksi soft, medium tai hard.

Käyttäjät voivat seurata kilpailun tuloksia reaaliajassa, mutta ei ole suoraa videolähetystä. Reaaliaikainen data sisältää kuljettajien kierrosajat ja sijainnit. Käyttäjä näkee, kuka johtaa kilpailua ja missä kukin kuljettaja on. Kilpailun jälkeen päivitämme myös tiimien ja kuljettajien kokonaispistemäärät.

Kaikki datavirta ja lähteet saadaan API lähteistä sekä F1 autoissa olevista 150-300 sensorista, jotka seuraavat ajoneuvojen suorituskykyä reaaliajassa. Nämä tiedot tallennetaan AWS pilvipalveluun.

Koska F1 lisenssi on kallis, tarjoamme Premium jäsenyyden, joka maksaa noin 10 euroa kuukaudessa. Premium käyttäjät saavat laajemmat analytiikkatietoja ja lisäominaisuuksia, kuten renkaiden seurantatiedot, reaaliaikaiset varikko ja tallin tiedot.

Priorisointi

Tärkein tavoite tässä Formula 1 tilasto ohjelmistossa on, että kaikki tulokset, jotka pistämme ohjelmisto palveluun, pystyy katsoa ja että ohjelmiston käyttäjä saa mahdollisimman täyden kuvan kyseisestä Formula 1 kaudesta. Projektissa tärkeimpänä on tuloksien muistiin painaminen tai jonnekin niiden kirjoittaminen, että voimme saada ne ohjelmistoon nopeammin meidän muistiinpanoistamme. Meidän on oltava tarkkoja siinä, ettemme vahingossa kirjaa yhtä

tietoa väärin, ettei henkilö, joka käyttää ohjelmaamme saa vääriä tietoja tietystä kisasta. Tärkein osa projektissamme on kuitenkin tietojen esille asettaminen ohjelmassamme ja että ohjelma on mahdollisimman helppokäyttöinen. Meille on myös tärkeää ottaa selvää, että lähde, jota käytämme on luotettava kuten Formula 1 omat nettisivut on luotettavin ohjelmisto kaikkien tulosten selvitykseksi.

Vaatimusten testaaminen

Käytämme vaatimusten testauksessa useita erilaisia testaustyyppejä, kuten esimerkiksi yksikkötestauksia, integraatiotestauksia, käyttöliittymätestauksia sekä kuormitus- ja suorituskykytestauksia. Jokaista toiminnallista ja ei-toiminnallista vaatimusta varten luodaan testitapaukset ja määritetään hyväksymiskriteerit, joiden avulla voidaan todeta, onko vaatimus täytetty.

Kokonaisuutena vaatimusten testauksen tavoitteena on varmistaa, että sivusto toimii odotetulla tavalla kaikissa tilanteissa ja täyttää sekä tekniset että käyttäjien odotukset.

Profiilit/sidosryhmät

1. F1-fanit
    - Suuri käyttäjäryhmä, joka haluaa seurata kisoja reaaliajassa. Faneille on tärkeä nähdä ajantasaiset tulokset, kilpailjoiden välinen aikaero sekä kierrosaikojen vertailut. He etsivät tietoa suosikiikuljettajiensa sijoituksista ja haluavat myös tarkastella kauden edistymistä.
2. Kilpailijat ja tallit
    - Kuljettajat ja heidän tiiminsä seuraavat omia ja muiden suorituksia analysoidakseen tuloksia ja kehittääkseen strategioita seuraavia kisoja varten.
3. Media

- Urheilutoimittajat ja uutistoimistot, jotka tarvitsevat tarkkoja ja reaaliaikaisia tietoja kisatuloksista, voidakseen välittää tärkeimmät hetket ja tulokset yleisölle nopeasti ja luotettavasti.

4. Kisajärjestäjät ja valvontajärjestöt

    - Kisjohen järjestäjät ja Fia varmistavat, että kaikki tiedot ovat tarkkoja ja sääntöjen mukaisia ennen julkaisua. Tämä lisää tulospalvelun luotettavuutta ja varmistaa virallisetkisatulokset.

User story -esimerkit

1. F1-fanina

    - Haluan nähdä reaaliakaiset tulokset ja kilpailjoiden välisen aikaeron, jotta voin seuratasuosikkikuljettajani suoritustaa kisan aikana.’

2. Kilpailijana

    - Haluan saada tarkat tiedot kierrosajoistani ja sijoituksistani verattuna muihin, jotta voin analysoida ja kehittää oma suoritustani.

3. Median edustajana

    - Haluan saada ajantasaisia ja luotettavia tietoja kisojen etenemisestä, jotta voin välittää tärkeimmät hetket yleisölllle nopeasti ja tarkasti.

4. Kisajärjestäjänä

    - haluan, että kaikki tulokset ovat tarkistettuja ja sääntöjen mukaisia ennen julkaisua, jotta virheet voidaan minimoida ja kisatulokset pysyvät luotettavina.

4 ARKKITEHTUURI- JA MODUULISUUNNITTELU, SAAVUTETTAVUUS

Koko järjestelmän kuvaus

Sivustollamme on hakutoiminto, joka löytyy heti ylhäältä ja vasemmalla puolella sijaitsee navigointipalkki. Navigointipalkki helpottaa sivustolla liikkumista. Sivustolta löytyy myös live Leaderboard. Se näyttää kuljettajien sijoitukset reaaliajassa. Sector Times ja Mini Sectors näyttävät väreillä merkitetyt kohdat esimerkiksi liila, vihreä ja keltainen. Väreillä voit nähdä kuljettajan parantuneen kierrosajan. Sivustolta voit myös seurata renkaiden käyttöä ja pit stoppeja(box). Samalla näet mitkä renkaat heillä ovat käytössä esimerkiksi soft, medium tai hard.

Frontendin toteutuksessa valittiin HTML, CSS ja Reactia. Reaaliaikaiseen livetulosten päivittämiseen käytetään WebSocket.

Backendissä käytetään Node.js ja Express.js. Tietokannaksi valittiin PostgreSQL, koska sitä käytetään virallisesti F1-tietokannoissa.

Tietoturvassa käytetään HTTPS- protokollaa, mikä suojaa tiedonsiirron salauksia. Helmet.js suojaa yleisiltä verkkouhilta. Sivustolla käytämme myös OAuthia eli käyttäjä voi hallita mitä tietoja ja oikeuksia sovellukselle myönnetään.

Ositus

Suurin osa frontendistä tulee meiltä itseltämme, eli sivuston suunnittelu ja käyttöliittymät. Frontendin toteutukseen käytämme HTML, CSS ja Reactia. Backend ja tietokanta tulevat pitkälti Formula1 lisenssistä, joka mahdollistaa heidän tietojensa käytön järjestelmässä. Käytämme sivustolla HTTPS-protokollaa.

Sivuston ylläpito tapahtuu lyhyissä jaksoissa. Tämä varmistaa ylläpidon sujuvuuden ja vähentää bugiriskiä.

Tietokanta liittyy järjestelmäämme Formula1 lisenssin kautta. Se siis edellyttää, että saamme reaaliaikaista dataa, jonka avulla voimme päivittää kuljettajien kierrosaikoja ja sijoituksia reaaliajassa. Tietokannasta saamme raakadataa, jota muokkaamme niin, että se olisi selkeä käyttäjälle, esimerkiksi kierrosajoiksi. Premium-käyttäjät voivat lisäksi hallita enemmän sitä, mitä tietoja he haluavat nähdä, kuten renkaiden historia.

Kun kyseessä on reaaliaikainen data, käytämme WebSocketia, joka mahdollistaa reaaliaikaisen datan käsittelyn, ja pyrimme minimoimaan viivästyksiä.

Tietoturvasyistä käytämme HTTPS protokollaa, joka varmistaa, että käyttäjän data on suojattu sen liikkuessa palvelimen välillä. Käytämme myös OAuth-järjestelmää, joka mahdollistaa kirjautumisen esimerkiksi Google-tilillä. Helmet.js lisää turvallisuutta ja estää klikkauksen kaappauksia ja muita uhkia.

Moduulisuunnittelu

Johdanto

Moduulisuunnittelun tarkoitus on muuttaa arkkitehtuurisuunnittelussa määritellyt rakenteelliset osat toimiviksi komponenteiksi ohjelmistossa.

Tavoite

Sivuston moduulisuunnittelun päätavoite on parantaa sen ylläpidettävyyttä, skaalautuvuutta ja joustavuutta. Erilaiset moduulit auttavat kehittäjiä muokkaamaan ja laajentamaan sivuston eri osia ilman, että muutokset vaikuttavat sivuston muihin osiin.

Moduulijaottelu

Luetellaan seuraavaksi sivuston mahdollisia moduuleja:

- **Käyttäjähallintamoduuli**
  - Käyttäjätunnistus ja käyttöoikeuksien hallinta.

- **Tulosten hallintamoduuli**
  - Tietokantojen ja tulosten käsittely.
  - Vastaa tulosten tallentamisesta, muokkaamisesta ja hakemisesta.

- **Käyttöliittymämoduuli**
  - Vastaa sivuston visuaalisen ja käytettävyyden puolesta.
  - Mahdollistaa eri käyttöliittymäkomponenttien (taulukot, grafiikat) uudelleenkäytön.

- **Ilmoitusmoduuli**
  - Lähettää käyttäjille ilmoituksia (esim. sähköposti, push-ilmoitukset).
  - Mahdollistaa viestien suuntaamisen eri paikkoihin ilman vaikutuksia muihin osiin.

Huomioitavaa moduulien suunnittelussa

Koska moduulisuunnittelun tulisi helpottaa kehittäjien toimintaa, on hyvä pitää muistissa, että yhden moduulin tulisi aina hoitaa vain yhtä osaa, ja että eri moduulien välillä tulisi myös olla mahdollisimman vähän riippuvuuksia toistensa kanssa. Jos jotain korjauksia joudutaan tekemään, tehdään niitä aina yhteen moduuliin kerrallaan.

Saavutettavuus avattu, miten huomioidaan

Ohjelmistossa on ilmainen versio, jossa näkyvät perustiedot kisan tuloksista kuten Kuskien järjestys, kisa-aika ja nopeimman kierroksen ajat. Ilmaisessa versiossa on myös Kaikkien mestaruuspisteet ja tallien mestaruuspisteet. Ohjelmistossa on myös Premium versio, joka maksaa 10 euroa kuukaudessa, jolloin voit saada tarkempia tietoja ja LIVE-ratakartan kisojen aikana, joka näyttää tarkalleen missä kohdassa rataa kuskit reaaliajassa. Premium versio tukee meidän toimintaamme, jotta pystymme maksamamaan lisenssit.

Riippuvuudet

Yleiset riippuvuudet

Ohjelmiston toiminta riippuu täysin virallisista F1-tietokannoista ja API-rajapinnoista, jotka toimittavat reaaliaikaiset tulokset ja muut tärkeät tiedot. AWS-pilvipalvelut ovat kriittisiä datan tallennuksessa ja käsittelyssä sekä järjestelmän skaalautuvuuden takaamisessa. Docker mahdollistaa järjestelmän ajamisen eri ympäristöissä ja siirrettävyyden muihin pilvipalveluihin. Frontendissä käytetään Reactia, ja backend on rakennettu Node.js:n ja Express.js:n avulla. Tietokantana toimii PostgreSQL, ja tietoturvaa varmistetaan Helmet.js:llä ja OAuthilla. Näiden työkalujen valinta perustuu niiden laajaan ylläpitoon ja vakauteen.

Tekniset riippuvuudet

Järjestelmä on täysin riippuvainen tietokannasta, sillä kaikki data tallennetaan ja haetaan PostgreSQL:stä. Jos tietokanta ei ole käytettävissä, järjestelmä ei voi tarjota reaaliaikaisia tai historiallisia tietoja. Tätä varten järjestelmässä tulisi olla fallback-mekanismeja, kuten välimuistipalveluita, jotka voivat toimia vikatilanteissa.

Tietoturva varmistetaan HTTPS-protokollalla ja OAuth-autentikoinnilla, mikä suojaa tiedonsiirtoa ja käyttäjien yksityisyyttä. Reaaliaikaisuus toteutetaan WebSocket-teknologialla, joka on integroitu backendissä ja mahdollistaa livetulokset käyttäjille. WebSocket skaalautuu hyvin pilvipalveluiden avulla, mikä mahdollistaa suurten käyttäjämäärien tukemisen esimerkiksi kisaviikonloppuina.

Infrastruktuuririippuvuudet

AWS on keskeinen datan tallennuksessa, käsittelyssä ja järjestelmän skaalautuvuuden takaamisessa. Docker-ympäristö määritellään sisältämään kaikki järjestelmän toimintaan tarvittavat osat, mikä varmistaa järjestelmän helpon hallinnan ja siirrettävyyden muihin pilvipalveluihin. CI/CD-prosessi toteutetaan GitHub Actionsilla, joka automatisoi testauksen ja käyttöönoton. Jos CI/CD-prosessi epäonnistuu, päivityksiä voidaan tehdä myös manuaalisesti.

Riippuvuuksien hallinta

Riippuvuuksien hallinta tapahtuu package.json-tiedoston kautta, jossa määritellään versiorajoitukset. Riippuvuudet päivitetään säännöllisesti, ja niiden vaikutukset testataan kehitysympäristössä ennen käyttöönottoa. Järjestelmään on suunniteltu fallback-ratkaisuja, kuten välimuisti ja paikalliset datakopiot, ulkoisten palveluiden katkosten varalta. Näin voidaan varmistaa perustoimintojen jatkuvuus ongelmatilanteissa.

Käyttäjäriippuvuudet

Käyttäjät ovat vahvasti riippuvaisia järjestelmän reaaliaikaisesta datasta. Jos tiedot eivät päivity ajantasaisesti, käyttäjäkokemus heikkenee huomattavasti. Näissä tilanteissa käyttäjille tiedotetaan palvelukatkoksista sovelluksen sisäisten ilmoitusten avulla. Näin käyttäjäpalaute ja luottamus palveluun voidaan ylläpitää myös ongelmatilanteissa, mikä takaa sovelluksen luotettavuuden ja pitkäaikaisen käytön.

5 TUOTTEENHALLINTA

Tuotteenhallinnan avulla varmistamme, että tulokset palvelussa ovat oikeat vertaamalla niitä tuloksiin, jotka ovat muissa palveluissa ja virallisessa lisenssissä, josta saamme kaikki tarkat tiedot. Palvelussa päivitämme sisältöä, joka F1-Session jälkeen ja joka kisaviikon lopun jälkeen. Tuotteenhallinta voi hieman viivästää projektin aikataulua siten, että emme tiedä, että pitenevät tarkistukset, jotka tarvitsee tehdä ennen julkaistusta.

Ohjelmiston versionhallinta tehdään GitHubissa, josta nähdään milloin ohjelmistoa on päivitetty ja mitä päivityksiä ohjelmistoon on milloinkin tehty. Versionhallinnan historian voidaan myös tallentaa GitHubiin siten, että voimme katsoa sieltä miten ohjelmistoa on päivitetty. Versiot tarkistetaan, jonkun kehittäjien toimesta ennen päivitysten julkaisemista, että kyseiset päivitykset eivät vaikuta mitenkään mihinkään vanhaan tietoon tai tiedostoon, jotta ohjelmisto pysyisi kunnossa. Uudet ominaisuudet otetaan käyttöön yleensä keskellä viikkoa, koska oletuksena on, että ohjelmiston käyttäjät käyttävät tulospalveluamme Viikonloppuisin, jolloin keskellä viikkoa ei ole niin paljon kävijöitä sivulla.

Dokumentti pitää olla hyvin järjestetty, ja siinä pitää olla selkeät ohjeet tulospalvelun ylläpitämiseen ja käyttämiseen. Aina kun joku tekee muutoksia sen pitää dokumentoida. Kun uusi ominaisuus lisätään, kehitystiimi päivittää dokumentaation. Vastuuhenkilö varmistaa, että dokumentti pysyy jatkuvasti ajan tasalla eli, kuka, milloin ja mitä muutoksia on tehnyt.

Verkkosivullamme on palautuskenttä, jonka kautta käyttäjät voivat antaa palautetta. Palautteen perusteella korjaamme mahdollisesti viat nopeasti. Palautetta hyödynnetään myös tuotteen kehittämisessä. Jos asiakas ehdottaa, että kisojen aikana tulisi näkyä kuljettajien parhaat kierrokset, niin kehitämme sen. Palautteet luokitellaan tärkeysjärjestykseen. Palautetta voi laittaa sähköpostiin niin tiedetään mitä korjata. Ei ole jatkuvaa julkaisua, vaan päivitämme tarpeen mukaan. Joskus teemme visuaalisia muutoksia. Uutiskirjeitä käytetään palvelun mainostamiseen. Käytämme automaattisia testejä ja monitorointijärjestelmiä, jotka tunnistavat mahdolliset virheet ja ylläpitävät järjestelmän laatua. Tuotteen elinkaari on pitkä, se on niin kauan kuin ihmiset ovat kiinnostuneita Formula 1.

6 TESTAUS

Tavoitteet

Testauksen tavoitteena on varmistaa, että tulospalvelu toimii odotetulla tavalla ja vastaa sille asetetut vaatimukset. Näitä vaatimuksia ovat käyttäjäystävällisyys, sivuston luotettava käyttö sekä ongelmaton käyttökokemus.

Testauksen laajuus

Testausten tulisi kattaa verkkosivun toiminnallisuus, käytettävyys, suorituskyky, tietoturva ja yhteensopivuus. Myös regressiotestaus olisi hyvä tehdä, eli testataan toimivatko kaikki sivuston osat mahdollisten muutosten tai päivitysten jälkeen.

Testausmenetelmät

Tulospalvelun toiminnan varmistamiseksi käytetään monia erilaisia testausmenetelmiä, joilla tarkastellaan järjestelmän eri osia. Testauksen päätarkoitukset ovat käyttäjäystävällisyyden, luotettavuuden ja sulavan käyttökokemuksen takaamisessa. Testausmenetelmät sisältävät seuraavat vaiheet:

1. **Manuaalinen testaus**

   Aluksi suoritetaan manuaalista testausta, jossa tarkistetaan järjestelmän perustoiminnot, kuten tulosten päivitys ja sivuston käyttö. Tämän avulla voidaan nopeasti tunnistaa helpoimmat virheet ja varmistaa, että järjestelmä täyttää käyttäjäkokemuksen vaatimukset.

2. **Yksikkötestaus**

Yksikkötestauksen tarkoituksena on testata tulospalvelun pienimpiä toiminnallisia osia, kuten kierrosaikojen ja sijoitusten laskentaa. Tämä vaihe varmistaa, että jokainen ohjelmiston osa toimii yksinään oikein ja odotetusti.

3. **Integraatiotestaus**

Integraatiotestauksessa varmistetaan, että tulospalvelun eri osat toimivat yhdessä. Esimerkiksi on tärkeää varmistaa, että tiedot päivittyvät oikein tietokannasta ja näkyvät käyttäjille ilman viiveitä tai virheitä.

4. **Käytettävyystestaus**

Käytettävyystestauksessa arvioidaan, kuinka helppoa sivuston käyttö on käyttäjälle. Tämä sisältää navigoinnin loogisuuden, tietojen selkeyden ja järjestelmän toimivuuden eri laitteilla ja selaimilla.

5. **Suorituskykytestaus**

Suorituskykytestauksessa tarkastellaan, miten järjestelmä kestää kuormitusta ja kuinka nopeasti se reagoi käyttäjän toimiin. Tämä vaihe on erityisen tärkeä, koska sivustolla voi mahdollisesti olla kisojen aikana paljon yhtäaikaisia käyttäjiä.

6. **Tietoturvatestaus**

Tietoturvatestauksessa tarkoituksena on varmistaa, että sekä käyttäjien että itse tulospalvelun tiedot ovat turvassa. Varmistettavat asiat ovat:

- **Tietoliikenne**
  Käytetään HTTPS-protokollaa tietojen salaamiseen
- **Pääsynhallinta**
  Varmistetaan, että luvattomat ylläpitotoiminnot eivät ole mahdollisia
- **Syötteiden validointi**
  Päämääränä on estää SQL-injektioiden mahdollisuus syöttökenttiä hyödyntämällä.

7. **Laatu**

1. Laatu ja testaus

Tuotteen laatu varmistetaan säännöllisellä testauksella, tarkistuksilla ja käyttäjäpalautteen keräämisellä. Testaukseen kuuluu yksikkö-, integraatio-, suorituskyky-, tietoturvatestaukset. Testauksen tulokset vaikuttavat suoraan julkaisupäätöksiin, virheet korjataan ennen julkaisua. Käyttäjäpalautteen avulla tunnistetaan kehitystarpeita ja lisätään ominaisuuksia, jotka parantavat käyttökokemusta.

2. Tuotteen julkaisu ja jakelu

Tuote julkaistaan pilvipalvelun kautta, ja päivityksistä tiedotetaan sovelluksen sisäisillä ilmoituksilla ja sähköpostilla. Päivitykset noudattavat vaiheittaisssa julaisustrategiaa, jossa isot ominaisuusoäivitykset julkaistaan major-versioina ja pienet virhejorjaukset minor-versioina. Tuote on suunniteltu toimimaan saumattomasti eri käyttöympäristöissä, ja pilvipalvelut varmistavat skaalautuvuuden ja käytettävyyden. Dockerin avulla käyttöönotto on nopeaa ja vaivatonta.

8. Regressiontestaus

Jos järjestelmään tehdään päivityksiä tai muutoksia, regressiontestauksella varmistetaan, etteivät uudet muutokset riko aiemmin toimivia ominaisuuksia.

LÄHTEET

Tähän tulevat lähteet ja rivivälinä on 1. Lähteet merkitään nimi-vuosi-järjestelmän mukaisesti Harvard-tyylillä. Tarkemmin ohjeita ja esimerkkejä löydät Opinnäytetyöohjeesta https://libguides.centria.fi/c.php?g=691790&p=4956682 .

Lähteet laitetaan aakkosjärjestykseen. Kaikista lähteistä täytyy löytyä tekstiviite tekstistä.

Lähteiden väliin jätetään yksi tyhjä rivi.

https://www.formula1.com/en/information/guidelines.4EOKE9RRqevL4niTK9kWyt

https://www.quora.com/How-is-data-transmitted-from-F1-cars-to-the-support-team-

https://www.quora.com/Which-programming-language-is-mostly-used-for-creating-websites

https://www.reddit.com/r/formula1/comments/i46mlj/web_application_i_developed_f1latest/

https://towardsdatascience.com/formula-1-race-predictor-5d4bfae887da

https://developer.mozilla.org/en-US/docs/Web

https://oauth.net/2/

https://developer.mozilla.org/en-US/docs/Web/API/WebSockets_API

https://q-factory.fi/testausblogi/laadunvarmistus-ehkaisee-virheiden-syntymista-ja-leviamista/

https://fi.wikipedia.org/wiki/Ohjelmistotuotanto#Ohjelmistosuunnittelu
https://openf1.org/

https://ohjelmistotuotanto-hy.github.io/osa4/#olio--ja-komponenttisuunnittelu

Kurssin materiaali: 05: Ohjelmistosuunnittelu

Kysytty chatgpt aiheista kysymyksiä
Aiheet:

- Toiminnalliset / ei-toiminnalliset vaatimukset
- Moduulisuunnittelu
- Rajapinnat
- Vaatimusten testaaminen
- Tuotteenhallinta
- Testaus

DEMO

https://f1-dash.com/dashboard

https://pienyrittaja.fi/gdpr/