

TeeMs: Projektinhallintasovellus

Janne Kuukkanen

Opinnäytetyö

Toukokuu 2017

Tekniikan ja liikenteen ala

Insinööri (AMK), Ohjelmistotekniikan tutkinto-ohjelma

| | | |
|---|-------------------------------------|-----------------------------------|
| Tekijä(t) Kuukkanen, Janne | Julkaisun laji Opinnäytetyö, AMK | Päivämäärä 10.05.2017 |
| | Sivumäärä 66 | Julkaisun kieli Suomi |
| | | Verkojulkaisulupa myönnetty: x |
| Työn nimi TeeMs: Projektinhallintasovellus | | |
| Tutkinto-ohjelma Ohjelmistotekniikka | | |
| Työn ohjaaja(t) Ari Rantala | | |
| Toimeksiantaja(t) | | |
| <p>Tiivistelmä</p> <p>Opinnäytetyön tavoite oli luoda Microsoftin .NET-pohjainen verkkosovellus, jolla voitaisiin hallita käynnissä olevia projekteja ja niitä työstäviä ryhmiä, jotka olisivat myös sovelluksen pääasiallisia käyttäjiä. Sovelluksella voitaisiin pitää kirjaa projektien eräpäivistä, tavoitteiden kuvauksesta sekä projektin eri osa-alueiden edistymisestä projekteihin kiinnitettävien työtehtävien muodossa. Sen lisäksi sovelluksella voitaisiin hallita projektiryhmien jäsenien lisäystä ja poistoa. Ryhmien jäsenille sovelluksessa on myös Chat- eli keskusteluhuone-ominaisuus, jonka avulla ryhmän jäsenet voivat keskustella toistensa kanssa.</p> <p>Sovellus toteutettiin C#- ja JavaScript-ohjelmointikielillä käyttäen ASP.NET-ohjelmistokehystä palvelinpuolen toteutukseen, SignalR-kirjastoa mahdollistamaan palvelin- ja käyttäjäpuolen reaaliaikainen kanssakäynti ja Entity Framework olio-relaatiokuvaus eli ORM-järjestelmää hakemaan ja tallentamaan tietoa MySQL-tietokannasta.</p> <p>Opinnäytetyön toteutus aloitettiin sovelluksen suunnitteluvaiheella, jossa ideoitiin ja suunniteltiin sovelluksen ominaisuudet, ulkonäkö sekä minkälaisia teknologioita ja kirjastoja sovelluksen toteutus vaatisi ja mihin verkkosijaintiin sovellus voitaisiin julkaista. Tätä seurasi sovelluksen toteutusvaihe, jossa sovellus ja sen toimintaan vaadittavat teknologiat, kirjastot ja ohjelmistot asennettiin tai ohjelmoitiin suunnitelmien mukaisiksi.</p> <p>Opinnäytetyön tuloksena saatiin ohjelmisto, joka vastaa vaatimusmäärittelyn tavoitteisiin ja jota voidaan käyttää ryhmäpohjaiseen projektinhallintaan. Sovelluksen kehityksessä hyödynnettiin onnistuneesti avoimen lähdekoodin työkaluja ja palvelinpuolen ratkaisuja sovelluksen julkaisussa.</p> | | |
| Avainsanat (asiasanat) Verkkosovellus, Projektinhallinta, ASP.NET, C#, JavaScript, Entity Framework, SignalR, MySQL | | |
| Muut tiedot | | |

| | | |
|---|--|---|
| Author(s) Kuukkanen, Janne | Type of publication Bachelor's thesis | Date 10.05.2017 Language of publication: Finnish |
| | Number of pages 66 | Permission for web publication: x |
| Title of publication TeeMs: Project management application | | |
| Degree programme Software Engineering | | |
| Supervisor(s) Ari Rantala | | |
| Assigned by | | |
| Abstract <p>The goal of the thesis was to create a Microsoft .NET based web application that could be used to manage projects and the groups working on them, who would also act as the primary users of the application. The application could be used to keep a record of a project's due dates, its description and its progress of the different subsections through assignments that can be linked to a project. In addition, the application could be used to add or remove members from a project group. For the members of a group, the application offers a Chat-feature the members can use to communicate among themselves.</p> <p>The application was made using C# and JavaScript programming languages, along with ASP.NET framework for the server-side implementation, SignalR library for enabling real time web functionality between server and client and Entity Framework ORM system to fetch and save data to a MySQL database.</p> <p>The thesis was started with a planning phase, where the features, appearance and the technologies and libraries the application would use were planned. The planning, was followed by the implementation phase, where the application and its peripherals were installed or programmed in accordance with the plans.</p> <p>The result of the thesis was a software product that can be used for team-based project management.</p> | | |
| Keywords/tags (subjects) Web application, Project management, ASP.NET, C#, JavaScript, Entity Framework, SignalR, MySQL | | |
| Miscellaneous | | |

Sisältö

| | |
|---|-----------|
| Sanasto | 5 |
| 1 Työn lähtökohdat | 8 |
| 1.1 Opinnäytetyön tausta..... | 8 |
| 1.2 Opinnäytetyön tavoitteet..... | 8 |
| 1.2.1 Pää tavoite | 8 |
| 1.2.2 Muut tavoitteet | 9 |
| 2 Vaatimusmäärittely | 9 |
| 2.1 Verkkosivusto | 9 |
| 2.1.1 Toiminnalliset vaatimukset..... | 9 |
| 2.1.2 Ei-toiminnalliset vaatimukset | 10 |
| 2.2 Tietokanta..... | 10 |
| 2.3 Dokumentointi | 10 |
| 3 Olemassa olevia projektinhallintasovelluksia | 10 |
| 3.1 Trello..... | 10 |
| 3.2 JIRA | 11 |
| 3.3 CEITON..... | 12 |
| 4 Suunnittelu ja taustatutkimus | 12 |
| 4.1 Yleistä | 12 |
| 4.2 Verkkosivuston ominaisuudet..... | 13 |
| 4.3 Verkkosivuston arkkitehtuuri | 13 |
| 4.4 Tietokanta..... | 15 |
| 4.5 Palvelin | 17 |
| 5 Sovelluksen toteutuksessa käytetyt työkalut ja teknologiat | 18 |
| 5.1 Entity Framework | 18 |

| | |
|---|-----------|
| | 2 |
| 5.2 SignalR | 20 |
| 5.3 Microsoft Visual Studio | 24 |
| 5.4 MySQL Workbench..... | 26 |
| 5.5 Github..... | 27 |
| 5.6 Microsoft Azure | 27 |
| 6 Opinnäytetyön toteutus | 29 |
| 6.1 Yleistä | 29 |
| 6.2 Toteutuksen suunnittelu | 29 |
| 6.3 Rekisteröinti ja sisäänkirjautuminen | 31 |
| 6.3.1 Käyttäjän rekisteröinti verkkosivulla | 31 |
| 6.3.2 Käyttäjän sisäänkirjautuminen | 31 |
| 6.4 Käyttäjän pääsivu | 32 |
| 6.5 Ryhmät verkkosivulla | 33 |
| 6.6 Ryhmien Chat-ominaisuus..... | 35 |
| 6.7 Projektit ja projektien työtehtävät verkkosivulla..... | 35 |
| 6.7.1 Projektit | 35 |
| 6.7.2 Projektien työtehtävät..... | 37 |
| 6.8 Käyttäjätietojen käsittely verkkosivulla | 38 |
| 6.9 Tietokanta..... | 39 |
| 6.9.1 Tietokannan asennus ja käyttöönotto sovelluksessa | 39 |
| 6.9.2 Tietokannan arkkitehtuuri | 41 |
| 6.9.3 Tietokannan relaatiot | 43 |
| 6.9.4 SignalR:n toiminta tietokannassa | 45 |
| 6.10 Sovelluksen julkaiseminen verkossa | 46 |
| 6.11 Testaus..... | 48 |
| 6.11.1 Yleistä..... | 48 |

| | |
|---|-----------|
| | 3 |
| 6.11.2 Testauksen toteutus | 48 |
| 6.12 Dokumentointi | 49 |
| 6.12.1 Yleistä..... | 49 |
| 6.12.2 Dokumentoinnin toteutus | 49 |
| 7 Opinnäytetyön tulokset..... | 50 |
| 7.1 Opinnäytetyön lopullinen tila..... | 50 |
| 7.2 Havaitut ongelmat..... | 51 |
| 8 Johtopäätökset ja jatkokehitys | 51 |
| Lähteet | 53 |
| Liitteet..... | 55 |
| Liite 1. Käyttäjän pääsivu sovelluksessa, Home.aspx..... | 55 |
| Liite 2. Käyttäjän pääsivulla sekä projektit ja ryhmät listaavilla sivuilla käytetty ohjelmistokoodi, projektien ja ryhmien näyttämiseksi..... | 56 |
| Liite 3. Ryhmäsivu sovelluksessa, Group.aspx | 59 |
| Liite 4. Ryhmien listaukseen käytetty sivu sovelluksessa. Tämä listaus poikkeaa käyttäjän pääsivun listauksesta näyttämällä ryhmien jäsenet, Groups.aspx..... | 60 |
| Liite 5. Käyttäjän Chat-sivu sovelluksessa, Chat.aspx | 61 |
| Liite 6. Projektisivu sovelluksessa, Project.aspx..... | 62 |
| Liite 7. Projektien listaukseen käytetty sivu sovelluksessa. Tämä listaus poikkeaa käyttäjän pääsivun listauksesta näyttämällä projektien työtehtävät, Projects.aspx..... | 63 |
| Liite 8. Projektin työtehtäväsivu sovelluksessa, Assignment.aspx..... | 64 |
| Liite 9. Tietokannan kokonaisdiagrammi | 65 |
| Liite 10. Microsoft Azure-pilvipalvelussa ajettujen rasiustestien tulokset | 66 |

Kuviot

| | |
|--|----|
| Kuvio 1. Sisäänkirjautumista seuraavan pääsivun malli..... | 15 |
| Kuvio 2. Opinnäytetyön palvelinratkaisun rautalankamalli..... | 18 |
| Kuvio 3. Entity Frameworkin luoma luokka tietokannan person-taulusta. | 19 |
| Kuvio 4. Entity Frameworkin käyttö. Kuvassa luodaan olio, joka sisältää tietokannan taulut ja sitten haetaan LINQ-kyselyllä person-taulusta henkilön käyttäjänimen perusteella oikea henkilö. | 20 |
| Kuvio 5. SignalR:n aloittamiseen tarvittava OWIN Startup-luokka..... | 21 |
| Kuvio 6. Projektin kuvan vaihtamiseen tarkoitettu funktio, joka seuraa, painetaanko btnChangeImage Id:llä yksilöityä nappia. | 22 |
| Kuvio 7. SignalR:n Hub-luokassa sijaitseva funktio, joka tallentaa projektien uudet kuvat tietokantaan. | 22 |
| Kuvio 8. Project-nimisen URL-parametrin asettaminen Hub-luokkaan, lopussa myös Hub-luokan yhteyden aloittavan funktion alku. | 23 |
| Kuvio 9. SignalR:n Hub-luokasta kutsuttu JavaScript-funktio, joka päivittää projektisivun kuvan ryhmän muille käyttäjille. | 24 |
| Kuvio 10. Intellisense täydentää edellisellä rivillä alustetun muuttujan nimen nykyisellä rivillä. | 24 |
| Kuvio 11. Visual Studion sisällä avattu Nuget-paketinhallintaominaisuus. | 25 |
| Kuvio 12. Microsoft Azuren kojelauta, jossa näkyy opinnäytetyön käyttämä MySQL-tietokanta kiinnitettynä etusivulle. | 28 |
| Kuvio 13. Ryhmän sivulla esitetty projektin info ja edistyminen..... | 34 |
| Kuvio 14. Kommentointi-toiminto Project.aspx-sivulla. | 37 |
| Kuvio 15. Työtehtävän tietojen muuttamiseen käytettävä ikkuna..... | 38 |
| Kuvio 16. Käyttäjän ja ryhmän jäsenten roolit tietokannassa. | 40 |
| Kuvio 17. EDM-tiedoston luontiin käytetty asennuskäyttöliittymä, jossa on nähtävissä paikalliseen tietokantayhteyteen käytetty "Connection String"-merkkijono..... | 41 |
| Kuvio 18. Project_Group- ja Project_Person-tauluissa käytössä oleva supporting-määrite. | 44 |
| Kuvio 19. Kuva Azuren kojelaudasta MySQL-tietokannan luontihetkellä. | 46 |
| Kuvio 20. Visual Studion verkkosovelluksen julkaisemiseen käytetty ikkuna. | 47 |

Sanasto

ADO.NET

ADO on .NET ohjelmistokehykselle rakennettu joukko luokkia, jotka altistavat tiedon siirtopalvelut .NET ohjelmistokehyksen kehittäjille, ADO.NET tarjoaa komponentteja hajautettujen tiedonjakamispalveluiden luomiseen sekä keinon relationaalisen, XML-pohjaisen ja sovellusdatan käsittelyyn .NET ohjelmistokehyksessä. (ADO.NET Overview. 2017.)

AJAX

Asynchronous JavaScript And XML. Ajax on tekniikka, jolla JavaScript voi lähettää palvelimelle pyynnön, vastaanottaa tuloksen ja päivittää ruutua sen mukaisesti. (Microsoft Ajax overview. 2017.)

ASP.NET

ASP on .NET ohjelmistokehyksen ohella julkaistu ohjelmistokehys, joka tarjoaa palveluita palvelin pohjaisten verkkosovellusten tekemiseen. (Anderson, R. Patel, S. Pasic, A. Dykstra, T. Learn ASP.NET? 2010.)

C#

C# on ohjelmointikieli, joka on kehitetty rakentamaan sovelluksia, jotka toimivat .NET ohjelmistokehyksellä. (Wagner, B. Wenzel, M. What is C#? 2016.)

CSS

Cascading Style Sheets. WWW-dokumenttien, kuten HTML-sivujen, sisällönhallintaan kehitetty tyyliohjeiden laji, jonka avulla dokumenteissa esiintyville elementeille voidaan säätää tyyliohjeita, joilla voidaan vaikuttaa elementin ulkonäköön verkkosivulla. (CSS Introduction. N.d)

Git

Git on avoimen lähdekoodin hajautettu versionhallintajärjestelmä ohjelmistoprojekteilte. (What is Git? N.d.)

Github

Versionhallintajärjestelmä ohjelmistojen lähdekoodeille, johon voidaan *Git*-ohjelmaa käyttämällä tallentaa versioita ohjelmistoista. Se tarjoaa *Git*-ohjelmalle verkkopohjaisen käyttöliittymän sekä useita ryhmätyöskentelyä helpottavaa ominaisuutta, kuten *Wiki*-sivun ja tehtävienhallintaan työkalut jokaiselle projektille. (Finley, K. What exactly is Github anyway? 2012.)

HTML

Hypertext Markup Language. Verkkosivujen merkintäkieli, jolla kerrotaan verkkoselaimelle, mitä elementtejä sivulla tulee näyttää. (What is HTML? N.d.)

HTTP

Hypertext Transfer Protocol. http on verkkosivujen ja palvelimien käyttämä, tilaton sovelluskerroksen tiedonsiirtoprotokolla. (Fielding, R. & Reschke, J. Hypertext Transfer Protocol (HTTP/1.1). 2014)

IDE

Integrated Development Environment. IDE on kokoelma erilaisia kehitystyökaluja yhteisen käyttöliittymän takana, mikä helpottaa niiden käyttöä ohjelmistotuotannossa. (Visual Studio IDE. 2015.)

LINQ

Language Integrated Query. LINQ on monen erilaisen datan tiedusteluun kehitetty kieli. Sillä voidaan hakea mm. relationaalista dataa tietokannoista, XML-dataa tai jopa tietorakenteita, kuten oliot. Kielen nimessä oleva "integrated" eli integroitu juontuu siitä, että kieli on osa ohjelmointikielen syntaksia. Esimerkiksi C# ja Visual Basic ovat ohjelmointikieliä, jotka toimitetaan LINQ-ominaisuuksilla. (Mayo, J. Introduction to LINQ. 2016.)

MySQL

MySQL on avoimen lähdekoodin tietokantaohjelmisto, joka käyttää relaatioita taulujen väliseen tiedon tallentamiseen, MySQL:ää ylläpitää yhdysvaltalainen yritys nimeltä Oracle Corporation. (What is MySQL? 2017.)

Pilvipalvelu

Pilvipalvelu on verkossa pyörivä ohjelmisto, joka tarjoaa käyttäjille erilaisia tietojen ja tiedostojen käsittelyyn tarkoitettuja palveluja, kuten palvelimia, tietokantoja, ohjelmia ja tilastoja internetin kautta ("pilvessä"). (What is cloud computing? 2017.)

SQL

Structured Query Language. Relaatiotietokantojen käyttämä kyselykieli tietokantakyselyjen toteuttamiseen. (What is SQL? N.d.)

URL

URL on toinen nimi verkko-osoitteelle. Se voi koostua joko sanoista (www.esimerkki.com) tai internetprotokolla (IP) osoitteesta (192.68.20.50). (HTML URL Encode. N.d.)

Verkkoselain

Ohjelma verkkosivun sisällön lataamiseen ja esittämiseen käyttäjän toimesta. (Garsiel, T. & Irish, P. How Browsers Work. 2011.)

VPN

Virtual Private Network. Verkko, joka mahdollistaa käyttäjän liittyä toiseen verkkoon tunnelointiprotokollan avulla. Tätä käytetään esimerkiksi, jotta yrityksen työntekijät voivat liittyä työpaikan verkkoon ja käyttää työpaikan sähköpostia kotikoneeltaan. (VPN. 2017.)

1 Työn lähtökohdat

1.1 Opinnäytetyön tausta

Suuri osa yritysten työskentelystä koostuu projektityöskentelystä. Sen vuoksi projektinhallinta on aina ajankohtainen aihe monella työpaikalla. Yhä useampi yritys ja organisaatio pyrkii siirtämään tai on jo siirtänyt oman projektinhallintansa digitaaliseen ympäristöön. Tämän vuoksi ne etsivät jatkuvasti uusia virtuaalisia työkaluja toimintansa helpottamiseksi.

Maailmassa on monenlaisia yrityksiä, joten projektinhallintatyökalujen tulee joko vastata monenlaisten yritysten tarpeisiin tai erikoistua jonkin tietyn alan kattavaan palveluun. Tämän vuoksi eri sovellukset vaihtelevat sekä ominaisuuksiltaan että tarjonnaltaan.

Mitä tällaisen projektinhallintaohjelmiston tekeminen vaatii? Millaisia ominaisuuksia siinä pitäisi olla, jotta sen avulla voidaan parantaa projektityöskentelyä? Tässä opinnäytetyössä toteutettiin tällainen projektinhallintasovellus.

Opinnäytetyön tekijä on koulutuksen aikana tutustunut moniin projektinhallintasovelluksiin ja ollut mukana projekteissa, joissa näitä sovelluksia on käytetty. Tämän vuoksi projektinhallintasovellusten toiminta ja käyttö projekteissa koettiin mielenkiintoiseksi aiheeksi, mikä johti päätökseen ottaa oman projektinhallintasovelluksen tekeminen opinnäytetyön aiheeksi.

1.2 Opinnäytetyön tavoitteet

1.2.1 Pää tavoite

Opinnäytetyön pää tavoite oli suunnitella ja toteuttaa verkkopohjainen projektinhallintasovellus, johon käyttäjä voi lisätä ryhmiä ja projekteja. Käyttäjän tulee myös pystyä lisäämään muita käyttäjiä ryhmiin ja yhdistämään ryhmä työstämään jotakin projektia. Näihin projekteihin tulee voida lisätä useita työtehtäviä, joilla on useita komponentteja. Lisäksi sovellukseen lisätään keskusteluhuone eli Chat-ominaisuus, jossa ryhmän jäsenet voivat keskustella keskenään.

Itse sovelluksen lisäksi. Sitä tukemaan suunniteltiin ja toteutettiin MySQL-tietokanta, johon tallennetaan tietoja yrityksistä, käyttäjistä ja projekteista. Tietokantaa suunniteltaessa ja toteutettaessa otettiin huomioon hyvät käytännöt tietokantojen toteutuksessa.

1.2.2 Muut tavoitteet

Muita tavoitteita opinnäytetyössä oli toteuttaa reaaliaikainen verkkosovellus. Tämä reaaliaikaisuus oli tarkoitus ottaa käyttöön ainakin sovelluksen Chat-ominaisuudessa, mutta resurssien riittäessä se voitaisiin laajentaa käsittelemään erinäistä dataa myös sovelluksen projekti- ja ryhmäsivuilla.

Henkilökohtaisena tavoitteena opinnäytetyössä oli oppia verkkosovellusten tietoturvan ja käyttäjäturvallisuuden varmistamisesta mm. sisäänkirjautuessa ja käyttäjän tietoja muutettaessa.

Toinen henkilökohtainen tavoite oli sovelluksen käyttäjähierarkian toteutus, eli ryhmien jäsenille voidaan asettaa erilaisia rooleja, kuten ylläpitäjä (Administrator) ja valvoja (Moderator) ja näiden roolien tulee vaikuttaa käyttäjien tietojen näkyvyyteen.

2 Vaatimusmäärittely

2.1 Verkkosivusto

2.1.1 Toiminnalliset vaatimukset

Web-sivustolla käyttäjän tulee pystyä rekisteröimään käyttäjätunnus ja sen avulla kirjautumaan sisään sovellukseen. Sovelluksessa käyttäjä voi lisätä ja poistaa ryhmiä tai ryhmänjäseniä, projekteja, projektin sisäisiä työtehtäviä ja näiden työtehtävien komponentteja. Käyttäjä voi myös profiilinäkymässä muokata omia tietojaan sekä nähdä muiden käyttäjien tietoja. Mikäli nämä ovat sen omissa asetuksissaan sallineet.

Käyttäjän tulee pystyä keskustelemaan omien ryhmiensä jäsenten kanssa sovelluksen Chat-palvelussa, joka toimii ryhmäpohjaisesti, eli jokaisella ryhmällä on erillinen keskustelu, johon käyttäjä voi liittyä vain, jos kuuluu tähän ryhmään.

2.1.2 Ei-toiminnalliset vaatimukset

Web-sivuston käyttäjien salasanoiden tulee olla suojattuna tietokannassa, eli käytännössä salasanoiden ei tallenneta käyttäjän syöttämässä muodossa, vaan ne muutetaan salausalgoritmin ja suolauksen kautta alkuperäisestä tunnistamattomaan muotoon, jollaisena ne tallennetaan tietokantaan.

Sovelluksen tulee toimia ainakin Windows-palvelimilla ja ASP.NETiä tukevassa pilvipalvelussa, kuten Microsoft Azure.

2.2 Tietokanta

Sovelluksen tietokannan tulee tallentaa tietoa ja relaatioita käyttäjistä, käyttäjien rooleista, ryhmistä ja projekteista. Käyttäjistä tallennetaan sisäänkirjautumistiedot ja vapaaehtoiset henkilötiedot ja sähköposti. Ryhmistä tulee tallentaa jäsenet ja näiden rooli ryhmässä. Projekteista tallennetaan projektia työstävät ryhmät ja projektin edistyminen sekä tiedot projektin työtehtävistä ja niiden komponenteista.

2.3 Dokumentointi

Opinnäytetyön dokumentointiin käytetään suunnitteluvaiheessa tehtäviä käyttötapauskaavioita ja asiankuuluvalla työkalulla tehtäviä verkkosovelluksen sivujen malleja. Opinnäytetyön loppudokumentointi ja toteutuksen kuvaus tehdään tässä raportissa.

3 Olemassa olevia projektinhallintasovelluksia

3.1 Trello

Trello on Fog Creek Software-yrityksen kehittämä projektinhallintasovellus, jonka ensimmäinen versio julkaistiin vuonna 2011. Trello toimii niin sanotulla ”Freemium”

kehitysstrategialla, jonka ajatus on antaa asiakkaille ilmaisia palveluita tuotteessa ja näiden avulla synnyttää mielenkiintoa tuotteen maksullisiin palveluihin. (Nations, D. What is freemium? 2017.)

Trello toimii käyttämällä Kanban-paradigmaa projektien hallintaan: Projektit esitetään tauluina, jotka sisältävät työtehtävälistoja. Nämä listat sisältävät lappuja, joita käyttäjä voi liikuttaa listojen välillä hiiren avulla vetämällä ja pudottamalla. Projektin jäseniä voidaan lisätä lappuihin näin merkiten heidät tekijöiksi eri työtehtäviin. (Getting started with Trello. 2017.)

Käyttäjän voivat lisätä Trellon lappuihin erilaisia työtehtäviin liittyviä tietoja, kuten valintaruutu tai eräpäivä. Käyttäjät voivat myös lisätä lappuihin otsikoita, joiden värikoodaaminen on mahdollista paremman järjestyksen aikaansaamiseksi projekteissa. Kaikki nämä tiedot näkyvät myös lapun pienemmässä muodossa työtehtävälistoissa, joiden välillä käyttäjä voi lappua siirtää.

Kaikki työtehtävälistat sijaitsevat tauluilla, joita käyttäjä voi itse tehdä tai toiset käyttäjät voivat kutsua muille omille laudoille nimeä tai sähköpostia käyttämällä. Laudoilla olevat käyttäjät ovat yleensä normaaleja käyttäjiä, mutta heille voidaan asettaa myös pääkäyttäjän oikeudet, jolloin hekin voivat kutsua käyttäjiä laudalle.

3.2 JIRA

JIRA on Atlassian yrityksen kehittämä bugien seurantaan ja projektinhallintaan keskittyvä sovellus, jonka ensimmäinen versio julkaistiin vuonna 2002. JIRA tarjotaan kolmessa eri paketissa: JIRA Core-paketti sisältää perusohjelman, JIRA Software sisältää JIRA Coren lisäksi JIRA Agile-paketin. Tämä paketti on tarkoitettu ohjelmistojen kehittäjille helpottamaan ketterän kehittämisen periaatteella toimivia ryhmiä, kolmas paketti on nimeltään JIRA Service Desk, joka on tarkoitettu IT-yrityksille ja palvelupisteille. JIRA on kehitetty Java-ohjelmointikielellä. (What are the differences between JIRA Software, JIRA Service Desk and JIRA Core. 2016.)

JIRAn projektit ovat kokoelma Aiheita (engl. Issues), joita hallitaan:

- Aihetyypeillä (engl. Issue type, kuten bugi, parannus johonkin toimintoon tai vaatimus)
- Näytöillä (engl. Screens. Mikä osa on milloinkin näkyvissä ja mikä ei ole)

- Työnkuluilla (engl. Workflows. Sarja tiloja ja siirtymiä joiden läpi aihe kulkee elinkaarensa aikana. Nämä tyypillisesti edustavat prosesseja)
- Käyttöoikeuksilla (engl. Permissions. Kenen sallitaan tekemään kukin prosessi aiheelle)
- Ilmoituksilla (engl. Notifications. Mitkä viestit välitetään kellekin ja missä vaiheessa työnkulkua). (Getting started with JIRA Software. 2017.)

3.3 CEITON

CEITON on verkkopohjainen omisteinen ohjelmisto, eli ohjelman käyttöoikeuslisenssi sallii ohjelmiston käyttämisen tietyin ehdoin, mutta sitä ei ilman erillistä lupaa saa levittää tai jakaa uudelleen. CEITON on tarkoitettu suunnittelemaan, hallinnoimaan, aikatauluttamaan ja laskuttamaan yritysten liiketoimintaprosesseja. (What is CEITON? 2017.)

CEITON sovelluksen aikataulusjärjestelmä käsittää projektien ja niiden tehtävien suunnittelun ja kirjanpidon, kun taas työnkulun hallinnointijärjestelmä sisältää informaation prosessoinnin ja yritysprosessien sujuvuuden koordinoinnin. Molemmat järjestelmät on integroitu saumattomasti, mutta niitä voidaan käyttää myös erillään.

4 Suunnittelu ja taustatutkimus

4.1 Yleistä

Suunnittelu on yksi ohjelmistojen valmistuksen tärkeimmistä vaiheista. Ilman suunnittelua ohjelmiston toteutus etenee täysin ”yritys ja erehdys”-metodilla, mikä voi johtaa vakaviin virheisiin, jotka huomataan vasta testauksessa. Nämä virheet voivat pysäyttää toteutuksen etenemisen pitkäksi aikaa, mikä syö paljon aikaa ja resursseja ja johtaa heikompaan lopputulokseen.

Ohjelmistojen valmistus aloitetaan yleensä suunnitteluvaiheella, jossa selvitetään tuotantoon käytettävissä olevat materiaaliset resurssit, työryhmän käytettävissä olevan ajan jakaminen suunnittelu-, toteutus- ja testausvaiheisiin, työtehtävien jakaminen työryhmän jäsenille näiden taitojen ja käytettävän ajan mukaan sekä riskiarviointi.

Resurssien selvityksessä tutkitaan ja valitaan vaatimusmäärittelyn pohjalta kaikki ohjelmiston toteutuksessa käytettävät ohjelmointiympäristöt, ohjelmointikieliset, tie-

tokannat, testausmenetelmät sekä muut työssä käytetyt työkalut ja tekniikat, kuten versionhallintaan käytettävät ohjelmat.

Riskiarvioinnissa keksitään riskejä, joita työryhmä saattaa kohdata tuotannon aikana. Samalla myös arvioidaan näiden riskien todennäköisyys sekä niiden haitallisuus ja neuvotellaan toimenpiteistä, joita käytetään riskien toteutuessa.

On kuitenkin lähes mahdotonta ottaa huomioon kaikki mahdolliset sovelluksen tekemiseen liittyvät seikat suunnitteluvaiheessa, joten ei ole suositeltavaa yrittää saada sovellusta lopulliseen muotoonsa jo suunnitteluvaiheessa. Tämä on totta varsinkin, jos kehittäjällä ei ole aikaisempaa kokemusta vastaavan sovelluksen toteuttamisesta.

4.2 Verkkosivuston ominaisuudet

Verkkosivuston ominaisuuksien suunnitteleminen aloitettiin tutustumalla projektityöskentelyyn ja olemassa oleviin projektityöskentelyyn liittyviin työkaluihin, kuten verkkopohjainen projektinhallintasovellus Trello sekä omiin kokemuksiin projektityöskentelystä. Näistä arvioitiin, mitkä ominaisuudet tarvitaan, jotta sovellus helpotaisi projektityöskentelyä ja mitkä ominaisuudet on mahdollista toteuttaa ajan ja kehittäjän taitojen puitteissa.

Lopuksi ominaisuudet, jotka valittiin osaksi sovellusta, listattiin tekstitiedostoon, jota myöhemmin käytettäisiin käyttötapausten ja käyttäjäpolkujen tekemiseen, jotta ominaisuudet saataisiin tarkennettua sellaiseen muotoon, että ne olisi helppo lisätä sovellukseen.

4.3 Verkkosivuston arkkitehtuuri

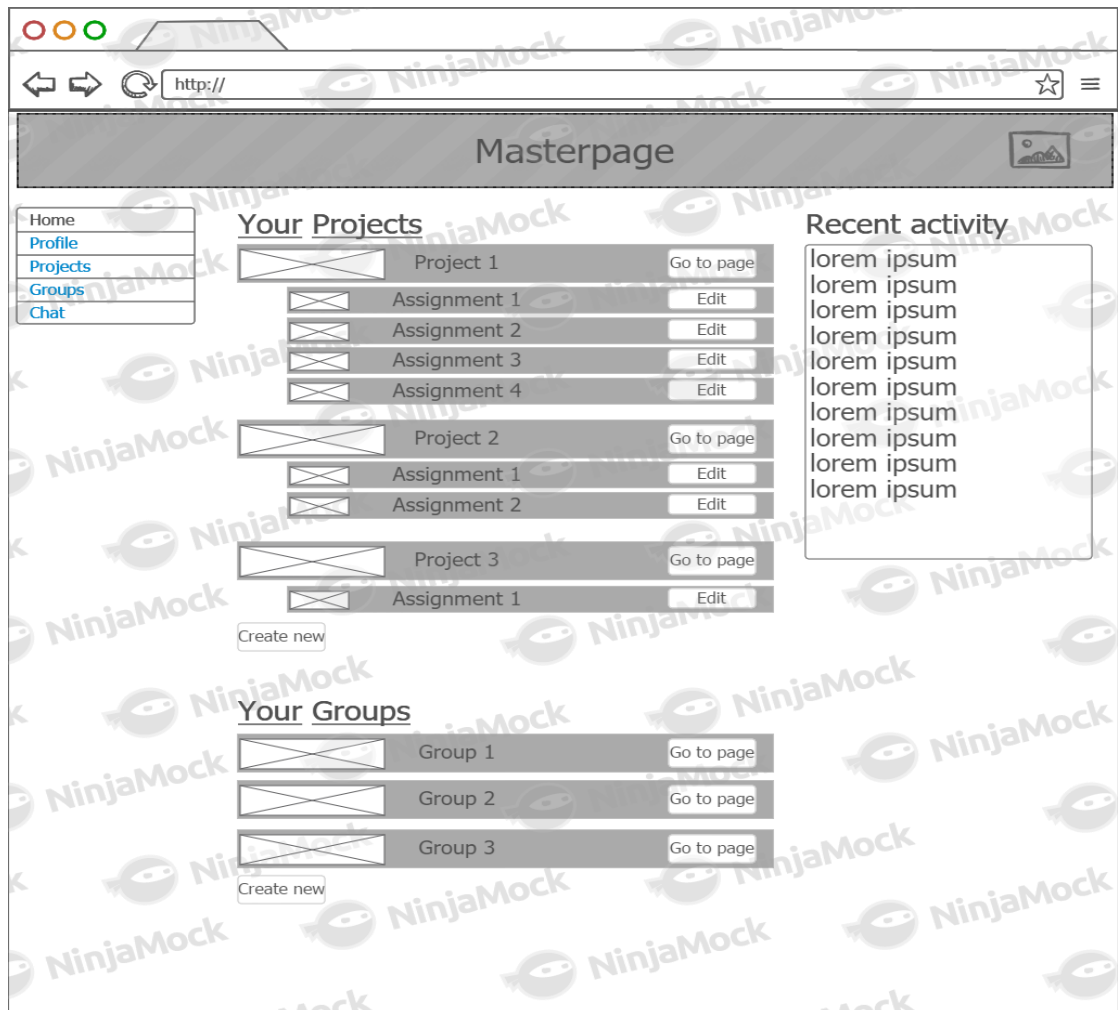
Verkkosivuston arkkitehtuurin suunnittelu aloitettiin tutkimalla, millaisia ominaisuuksia projektinhallintaan keskittyvä verkkosivusto tarvitsee ja pitämällä aivoriihi, jossa kirjattiin ylös mahdollisia ominaisuuksia ja ideoita, joita verkkosivulla voitaisiin toteuttaa. Näitä ideoita verrattiin ja arvioitiin tutkimuksen tuloksiin ja vaatimusmäärittelyyn. Arvioinnin tuloksena valittiin olennaisimmat ominaisuudet, joita lähdettiin toteuttamaan.

Kaikki halutut ominaisuudet merkittiin eriävästi niiden tarpeiden mukaan. Ominaisuudet, jotka todettiin olennaiseksi sovelluksen toiminnan kannalta, merkittiin ensimmäisen luokan prioriteetiksi, jotka on saatava valmiiksi tuotteen kehitysaikassa. Toisen luokan prioriteetit ovat ominaisuuksia, jotka todennäköisesti toteutetaan, mutta ne voidaan jättää pois, jos aika ei riitä ensimmäisen luokan ominaisuuksien tekemiseen. Kolmannen luokan prioriteetit ovat ominaisuuksia, jotka tehdään vain, jos aikataulusta ollaan selvästi edellä.

Verkkosivuston arkkitehtuurin suunnittelua jatkettiin valitsemalla mallinnusohjelma, jolla saatiin tehtyä rautalankamalleja verkkosivulle tulevista sivuista ja niiden sisällöstä sekä toiminnoista.

Mallinnusohjelmaksi valittiin NinjaMock sen helpon käytettävyyden ja yksinkertaisen käyttöliittymän takia. NinjaMock tarjoaa myös ilmaisohjelmaksi paljon tehokkaita ominaisuuksia, kuten linkityksen eri mallien välille käyttäen mallien HTML-elementtejä. Tämä mahdollistaa selvän rakenteen suunnittelun jo mallitasolla ilman minkäänlaisen ohjelmakoodin kirjoittamista.

Mallinnusohjelmalla suunniteltiin kaikki verkkosivustoon tulevat pääsivut ja niistä tehtiin rautalankamallit, joihin sisällytettiin sivujen ulkomuoto, html-elementit sekä linkit muiden sivujen malleihin. Myös sivujen käyttäjystävällisyyttä pyrittiin ottamaan huomioon, kuten esimerkiksi Home.aspx-sivun projektiluettelon työtehtävien (assignment) listauksen näyttämisen suunnittelussa kuvion 1 mukaisesti. Tätä listaus-ta päädyttiin myöhemmin käyttämään ainoastaan erillisessä, kaikki käyttäjän projektit listaavassa näkymässä, jonne käyttäjä voi navigoida verkkosivulla olevasta navigointipalkista.



Kuvio 1. Sisäänkirjautumista seuraavan pääsivun malli.

4.4 Tietokanta

Tietokannan suunnittelu aloitettiin tutkimalla, minkälaista tietokantaa sovelluksen tietojen tallentamiseen kannattaa käyttää. Ensimmäinen kysymys kannan selvittämiseksi oli SQL- tai NoSQL-kannan käyttäminen. SQL-kannan etuja ovat relaatiot eri taulujen välillä, mikä tekee tiedon käsittelystä helpompaa, mutta NoSQL-kannan avulla voitaisiin tallentaa ja käsitellä huomattavasti suurempia määriä tietoja ilman merkittävää hidastumista tietokannan toiminnoissa, mikä on toivottavaa sovelluksen Kommentti- ja Chat-ominaisuuksien kanssa.

SQL-kantojen relaatiot koettiin tärkeämmäksi ominaisuudeksi kuin NoSQL kantojen tarjoama nopeus. Sillä NoSQL kannat toimivat parhaiten, kun ne keskittyvät vain tietynlaisen datan käsittelyyn. Esimerkiksi, jos opinnäytetyö olisi keskittynyt ainoastaan

Chat-palvelun tekemiseen, olisi NoSQL-kanta ollut hyvä valinta palvelun viestien käsittelyyn, mutta koska opinnäytetyöhön kuuluu Chat-palvelun lisäksi monia muita ominaisuuksia, on tietokannan datan jäsenitys paljon tärkeämpää. Näistä syistä joutuin opinnäytetyössä päädyttiin käyttämään SQL-tietokantaa.

Seuraavaksi täytyi päättää, mikä tyyppinen SQL-kanta tietokannan tulisi olla. Tämän tyyppisiä kantoja on useita, kuten MySQL ja PostgreSQL. Tässä opinnäytetyössä päädyttiin käyttämään MySQL-kantaa. MySQL-tietokanta valittiin sen helpon käytön ja asennuksen vuoksi, joten se ei vaadi suurta määrää käytön opettelua ennen käyttöönottoa. MySQL on myös ilmainen ja sillä on valmiiksi laaja käyttäjäkunta, mikä helpottaa asennusta ja vianetsintää. Lisäksi MySQL on avoimen lähdekoodin ohjelmisto ja opinnäytetyön tekijällä oli myös MySQL-kannoista eniten kokemusta.

MySQL:n heikkouksia ovat sen skaalautuvuus, joustamaton skeema ja sen huono soveltuvuus tietokantaklusterointiin, mutta koska opinnäytetyön käyttäjäkunta ei lyhyellä tähtämellä tule olemaan järin suuri eivätkä opinnäytetyön resurssit mahdollista todella skaalautuvan sovelluksen julkaisemista, vaan sen toiminnan pullonkaulana olisi alusta lähtien tietokannan verkkosijainnin tarjoama tiedonsiirtonopeus. Eivät MySQL:n heikkoudet ole ongelmia. Lisäksi Tietokantaklusterointia ei ole suunniteltu suoritettavaksi opinnäytetyön aikana.

Seuraavaksi päätettiin tietokannan sijainti. Vaihtoehtoina oli rekisteröidä valmiiksi jokin Microsoftin palvelin, johon tietokanta ja myöhemmin valmis sovellus voitaisiin sijoittaa, käyttää valmiita JAMK:n tarjoamia tietokantoja VPN-yhteyden kautta tai tehdä tietokanta paikalliseksi omalle tietokoneelle.

Tietokannan sijainti suunniteltiin niin, että sovelluksen perusrakenteen kehityksessä tietokanta asennetaan paikallisesti omalle koneelle, mutta tietokantayhteyden testaus toteutetaan JAMK:n tarjoamilla tietokannoilla ja lopulta kehityksen loppuvaiheessa tietokanta siirretään yhdessä sovelluksen kanssa Microsoftin tarjoamaan palveluun, kuten Microsoft Azure. Tällä toteutuksella saadaan parhaiten testattua tietokantayhteys verkon läpi, mutta kaikki lokitiedostot ovat saatavilla kehityskoneella, jolloin mahdolliset ongelmat tietokannan puolella saadaan tutkittua tarkemmin. Tällä järjestelyllä sovelluksen lopullista sijaintia ei vielä tarvinnut lyödä lukkoon.

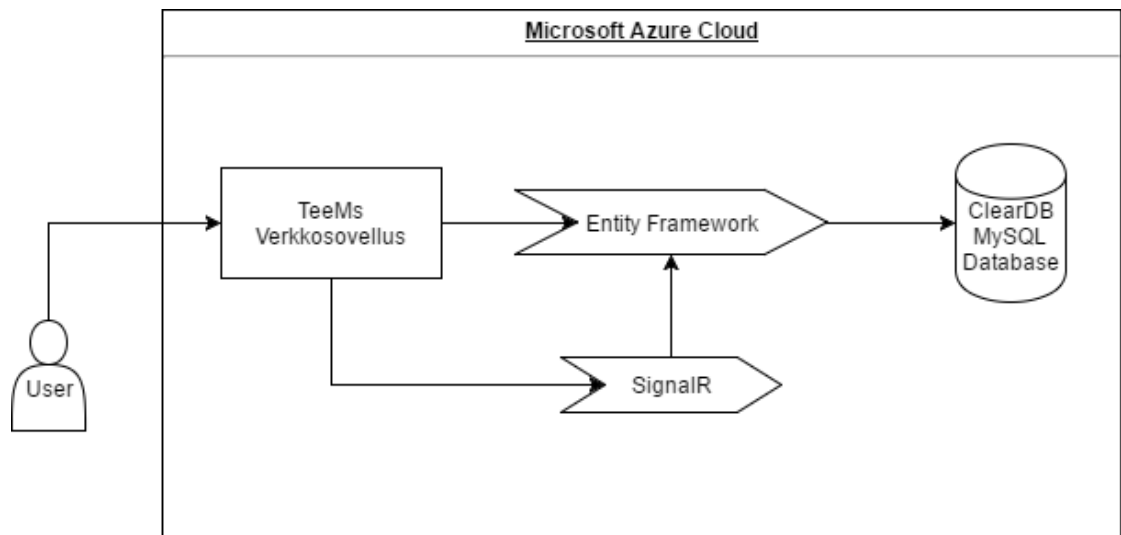
Koska tietokantana päätettiin käyttää MySQL-kantaa, oli tietokannan toteutuksessa käytettävien työkalujen päättäminen yksinkertaista. Sillä MySQL:llä on olemassa hyvät tarvikkeet tietokantojen luomiseen MySQL Installer-nimisen ohjelman muodossa, joka mahdollistaa sekä paikallisen tietokannan asennuksen, että itse tietokantojen ja tietokantojen mallin, eli skeeman luonnin asentamalla MySQL Workbench ohjelman. (MySQL Installer. 2017.)

Tietokannan taulujen ja relaatioiden suunnittelussa tarkasteltiin vaatimusmäärittelyn pohjalta, mitkä asiat olisivat sovelluksen kannalta tärkeitä, mitä tietoja niistä pitäisi tallentaa tietokantaan ja millaisia relaatioita sovellus tulisi tarvitsemaan haluttujen ominaisuuksien toteuttamiseksi.

4.5 Palvelin

Opinnäytetyön verkkosijainniksi valittiin Microsoft Azure-pilvipalvelu. Azure on luonnollinen valinta ASP.NETillä kehitetyille projekteille. Sillä se on Microsoftin itsensä ylläpitämä palvelu, mikä tekee sovelluksen verkkojulkaisusta helpon ja vaivattoman. Azure tarjoaa myös useita tietokantoja käyttäjilleen, mukaan lukien opinnäytetyön käyttämän MySQL-kannan, mikä osaltaan vaikutti päätökseen valita Azure.

Koska Azure on pilvipalvelu. Se tarjoaa joustavuutta uusien palvelujen luomisessa. Tämä on hyödyllistä, kun opinnäytetyölle luodaan verkkosijaintia. Kaiken lisäksi Azure tarjoaa myös monia palveluita ilmaiseksi, mikä puolestaan on toivottavaa, kun otetaan huomioon opinnäytetyön tekemisen rajalliset resurssit. Sovelluksen palvelinarkkitehtuurin rautalankamalli on esitetty kuviossa 2



Kuvio 2. Opinnäytetyön palvelinratkaisun rautalankamalli.

5 Sovelluksen toteutuksessa käytetyt työkalut ja teknologiat

5.1 Entity Framework

Microsoft ADO.NET Entity Framework on C#-ohjelmointikielelle toteutettu olio-relaatiokuvaus, ORM-järjestelmä (Object-Relational Mapping). Entity Frameworkin avulla voidaan halutun tietokannan tauluista tehdä luokkarakenne sovellukseen. Nämä rakenteet ovat kuvion 3 mukaisia luokkia. (Entity Framework. 2016.)

```

223 public partial class person
224 {
225     public person()
226     {
227         this.assignment_person = new HashSet<assignment_person>();
228         this.group_member = new HashSet<group_member>();
229         this.assignment_component_person = new HashSet<assignment_component_person>();
230         this.project_person = new HashSet<project_person>();
231         this.comment = new HashSet<comment>();
232         this.connection = new HashSet<connection>();
233         this.message = new HashSet<message>();
234         this.invite = new HashSet<invite>();
235     }
236
237     public int person_id { get; set; }
238     public string first_name { get; set; }
239     public string last_name { get; set; }
240     public string username { get; set; }
241     public string email { get; set; }
242     public int privacy { get; set; }
243     public System.DateTime creation_date { get; set; }
244     public Nullable<System.DateTime> edited { get; set; }
245     public int role_id { get; set; }
246
247     public virtual ICollection<assignment_person> assignment_person { get; set; }
248     public virtual ICollection<group_member> group_member { get; set; }
249     public virtual role role { get; set; }
250     public virtual ICollection<assignment_component_person> assignment_component_person { get; set; }
251     public virtual ICollection<project_person> project_person { get; set; }
252     public virtual ICollection<comment> comment { get; set; }
253     public virtual ICollection<connection> connection { get; set; }
254     public virtual ICollection<message> message { get; set; }
255     public virtual ICollection<invite> invite { get; set; }
256 }

```

Kuvio 3. Entity Frameworkin luoma luokka tietokannan person-taulusta.

Kuviossa 3 on kuvattu Entity Frameworkin luoma luokka, joka on luotu sovelluksen käyttämän tietokannan person-taulun pohjalta. Tätä luokkaa voidaan käyttää sovelluksen ohjelmoinnissa kutsumalla person-oliota. Tämä mahdollistaa tietokantadatan muokkaamisen ohjelmistokoodissa.

Näiden tietokannan taulujen luokkarakenteiden avulla saadaan huomattavasti helpotettua tiedonkulkua sovelluksen ja tietokannan välillä. Sillä se ei tarvitse monimutkaisia datarakenteita sovelluksen ohjelmakoodissa SQL-kyselyjen mahdollistamiseksi, vaan käyttäjä voi suoraan olla vuorovaikutuksessa luokkien kanssa ja kun haluttu data on ensin manipuloitu sovelluksen ja luokkien välillä se voidaan helpolla komennolla tallentaa tietokantaan.

Entity Framework toimii myös yhdessä LINQ-kyselyiden kanssa. LINQ-kyselyitä käytämällä yhdessä Entity Frameworkin kanssa, kuten kuviossa 4 on esitetty, sovellus voidaan myös turvata ns. SQL-injection hyökkäyksiltä. Sillä LINQ-kyselyt parametrisoivat tietokantaan lähetetyn datan. Mikäli Entity Frameworkin kanssa käytetään

normaaleja SQL-kyselyitä, käyttäjän tulee muistaa parametrisoida lähetettävä data itse.

```
885
886         TeeMsEntities context = new TeeMsEntities();
887
888         string uname = User.Identity.Name;
889         var dbperson = context.person.Where(person => person.username == uname).SingleOrDefault();
890
```

Kuvio 4. Entity Frameworkin käyttö. Kuvassa luodaan olio, joka sisältää tietokannan taulut ja sitten haetaan LINQ-kyselyllä person-taulusta henkilön käyttäjänimen perusteella oikea henkilö.

5.2 SignalR

ASP.NET SignalR on reaaliaikaisten verkkosovellusten kehittämiseen luotu kirjasto, joka mahdollistaa kaksisuuntaisen (engl. full duplex, FD) liikenteen palvelimen ja käyttäjän päätteen välillä. Palvelin voi SignalR:n avulla lähettää tietoa heti, kun se tulee saataville. SignalR tukee WebSocket-protokollaa ja käyttää muita yhteensopivia teknologioita. Mikäli WebSocket ei ole käytettävissä, johtuen esimerkiksi vanhasta verkkoselaimesta, tukeutuu SignalR vanhempiin tiedonsiirron menetelmiin. (Fletcher, P. Patel, S. Pasic, A. Dykstra, T. Introduction to SignalR. 2014.)

SignalR tarjoaa yksinkertaisen ohjelmointirajapinnan, jonka avulla voidaan luoda proseduurien etäkutsuja (RPC) palvelimelta päätteelle, jotka kutsuvat JavaScript-funktioita käyttäjän selaimessa, palvelinpuolen C#-koodista. SignalR sisältää myös ohjelmointirajapinnan yhteyksien hallinnointiin (kuten yhteyden luontiin ja katkaisemiseen) ja yhteyksien ryhmittelyyn.

SignalR-teknologiaa käytettiin muodostamaan reaaliaikaista toiminnallisuutta sovelluksen projekti-, ryhmä- ja Chat-sivuille. Projekti- ja ryhmä-sivuilla kaikki käyttäjien tekemät muutokset tallennetaan ensin tietokantaan ja sitten tieto muutoksesta lähetetään SignalR:n avulla muille projektia työstäville jäsenille.

SignalR toimii sovelluksessa ns. Hub-luokkien avulla. Käyttäjän navigoidessa sivulle, jolle on asetettu SignalR yhteyden avaava JavaScript-funktio. Ottaa JavaScript-funktio yhteyden palvelimella sijaitsevan Hub-luokan OnConnected-metodiin SignalR:n API:n kautta. Tässä metodissa käyttäjän senhetkinen verkkosijainti kirjataan tietokan-

taan ja hänet yhdistetään johonkin SignalR:n ryhmään riippuen siitä, mille ryhmä- tai projektisivulle hän saapui.

SignalR vaatii toimiakseen vielä ns. "OWIN Startup"-luokan. Tämän yksinkertaisen luokan tarkoitus on erottaa palvelin sovelluksesta, eli luokka luo yhteyden "/signalr" URL-osoitteeseen, jota kaikki SignalR:n toiminnallisuus sovelluksessa tulee käyttämään API:n kautta. OWIN-luokka on esitetty kuviossa 5.

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Web;
5  using Microsoft.Owin;
6  using Owin;
7  [assembly: OwinStartup(typeof(SignalRTeeMs.Startup))]
8
9  namespace SignalRTeeMs
10 {
11     public class Startup
12     {
13         public Startup()
14         {
15             //
16             // TODO: Add constructor logic here
17             //
18         }
19
20         public void Configuration(IAppBuilder app)
21         {
22             // Any connection or hub wire up and configuration should go here
23             app.MapSignalR();
24         }
25     }
26 }
```

Kuvio 5. SignalR:n aloittamiseen tarvittava *OWIN Startup*-luokka.

Kun käyttäjä on yhdistetty SignalR:n Hub-luokkaan ja hänen verkkosijaintinsa on kirjattu tietokantaan, seuraava toiminnallisuus tapahtuu, kun käyttäjä käynnistää verkkosivulla toiminnon, johon on sivulla kiinnitetty JavaScript tapahtumankäsittelijä. Esimerkiksi, jos käyttäjä on vaihtamassa projektin profiilikuvaa, painaessaan kuvan tallennukseen käytettävää "Change Image"-nappia, JavaScript ajaa kuviossa 6 esitetyn, JQuery-kirjastoa käyttävän funktion.


```

318     $('#btnChangeImage').click(function () {
319
320         var project_imageurl = $('#txtChangeImage').val();
321         $('#txtChangeImage').val('');
322
323         $('#imgProjectImage').attr("src", project_imageurl);
324
325         $('#divChangeTitleAndImage').css({ "display": 'none' });
326         $('#divTitleAndImageDefault').css({ "display": 'inline-block' });
327         $('#divChangeImage').css({ "display": 'none' });
328
329         project.server.saveProjectImage(project_imageurl);
330
331     });

```

Kuvio 6. Projektin kuvan vaihtamiseen tarkoitettu funktio, joka seuraa, painetaanko *btnChangeImage* Id:llä yksilöityä nappia.

Kuviossa 6 kuvatussa funktiossa projektikuvan näyttävä elementti projektisivulla tuodaan esiin ja vaihdettavan kuvan URL-osoitteen syöttämiseen käytetty tekstilaatikko piilotetaan CSS-tyyleillä. Olennaisin osa SignalR:n osalta on kuitenkin seuraava rivi:

```
project.server.saveProjectImage(project_imageurl);
```

Tällä rivillä JavaScriptissä kutsutaan SignalR:n *Hub*-luokassa sijaitsevaa "SaveProjectImage"-metodia, joka on esitetty kuviossa 7, jossa uuden kuvan URL-osoite tallennetaan tietokantaan ja kuvan vaihtamisesta ilmoitetaan muille vaihtamishetkellä samassa SignalR ryhmässä oleville jäsenille. Kutsuttavan metodin nimi täytyy kirjoittaa ns. "Lower Camel Case"-nimeämiskäytännöllä, eli funktion ensimmäisen kirjaimen täytyy olla pieni kirjain, koska SignalR:n syntaksi vaatii sitä.

```

405     public void SaveProjectImage(string project_imageurl)
406     {
407         try
408         {
409             ctx = new TeeMsEntities();
410
411             int project_id = int.Parse(this.Context.QueryString["Project"]);
412
413             var rightperson = ctx.person.Where(p => p.username == Context.User.Identity.Name).SingleOrDefault();
414             var rightconnection = ctx.connection.Where(con => con.person_id == rightperson.person_id).SingleOrDefault();
415             var rightproject = ctx.project.Where(pr => pr.project_id == project_id).SingleOrDefault();
416
417             rightproject.picture_url = project_imageurl;
418             ctx.SaveChanges();
419
420             // Call updateProjectTitle method on the client side
421             Clients.OthersInGroup(rightproject.name + rightproject.project_id.ToString()).updateProjectImage(project_imageurl);
422         }
423         catch (Exception ex)
424         {
425             throw ex;
426         }
427     }
428 }

```

Kuvio 7. SignalR:n Hub-luokassa sijaitseva funktio, joka tallentaa projektien uudet kuvat tietokantaan.

Kuviossa 7 kuvatussa funktiossa vastaanotetaan käyttäjäpuolen JavaScript-funktiosta lähetetty `project_imageurl`-muuttuja, haetaan Entity Frameworkin ja LINQ-kyselyjen avulla oikea projekti, jossa kuva muutetaan ja tallennetaan uuden kuvan URL-osoite tietokantaan. Lopulta kuvan muutoksesta ilmoitetaan muille SignalR-ryhmän jäsenille.

Kuviossa 7 kuvatussa funktiossa on pari SignalR:n toimintaan liittyvää kohtaa, jotka on syytä käydä läpi. Ensimmäinen niistä on seuraava rivi:

```
int project_id = int.Parse(this.Context.QueryString["Project"]);
```

Koska SignalR tekee jokaiselle käyttäjälle tilapäisen ilmentymän Hub-luokasta, ei SignalR:n palvelinpuolella voida käsitellä normaalisti saatavilla olevia resursseja, kuten URL-parametreja. Tämän vuoksi vastaavat resurssit tulee syöttää SignalR:n ilmentymään samalla, kun se luodaan. Tämä tehdään käyttäjäpuolen JavaScriptissa kuviossa 8 esitetyllä tavalla.

```
289 // Set connection parameters
290 $.connection.hub.qs = { 'Project': urlParams.get('Project') };
291
292 // Start the connection.
293 $.connection.hub.start().done(function () {
```

Kuvio 8. Project-nimisen URL-parametrin asettaminen Hub-luokkaan, lopussa myös Hub-luokan yhteyden aloittavan funktion alku.

Toinen kuviossa 7 esiintyvä mainitsemisen arvoinen asia on rivi, jossa muille SignalR-ryhmän jäsenille ilmoitetaan projektin kuvan vaihtuneen. Rivi on seuraavanlainen:

```
Clients.OthersInGroup(rightproject.name +
rightproject.project_id.ToString()).updateProjectImage(project_imageurl);
```

Rivin komento toimii seuraavasti. Ensimmäisessä kohdassa, eli `Clients`-komennolla haetaan kaikki kyseiseen Hub-luokkaan sillä hetkellä yhteydessä olevat käyttäjät. Seuraavaksi rajataan kaikista käyttäjistä ainoastaan yhteen ryhmään kuuluvat käyttäjät (paitsi kuvan muuttanut käyttäjä), haettavan ryhmän nimi on annettu sulkujen sisällä. Tässä tapauksessa siis projektin nimi ja Id yhdistettynä yhdeksi merkkijonoksi.

Komennon lopussa on annettu JavaScript-funktio, joka kutsutaan jokaisen rajatun käyttäjän kohdalla. Tämä funktio on esitetty kuviossa 9.

```

220     project.client.updateProjectImage = function (project_imageurl) {
221         // Change the project image.
222         $('#imgProjectImage').attr("src", project_imageurl);
223     };

```

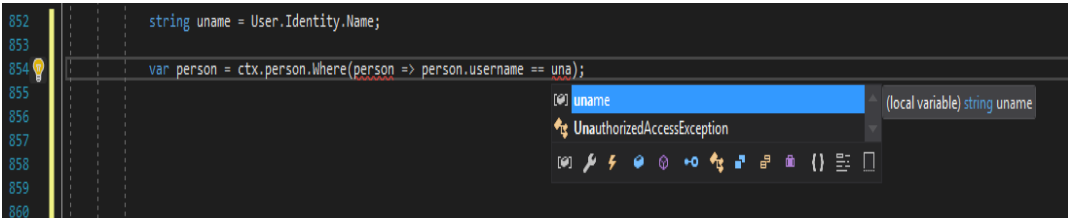
Kuvio 9. SignalR:n Hub-luokasta kutsuttu JavaScript-funktio, joka päivittää projektisivun SignalR-ryhmän muille käyttäjille.

5.3 Microsoft Visual Studio

Microsoftin Visual Studio on ohjelmointiympäristö (IDE), jota käytetään ohjelmien kehittämiseen Microsoftin Windows-käyttöjärjestelmälle, mutta myös verkkosivujen, verkkosovellusten ja palvelujen sekä mobiilisovellusten tekemiseen. (Visual Studio IDE. 2015.)

Visual Studio sisältää tekstinmuokkausohjelman, jolla muokataan sovelluksen lähdekoodia. Tämä muokkausohjelma tukee sekä Intellisenseä, että koodin refaktorointia, joka on prosessi, jossa ohjelma muuttaa lähdekoodia siten, että sen toiminnallisuus säilyy, mutta sen sisäinen rakenne paranee

Intellisense on Visual Studion sisäinen ohjelma, joka pyrkii täydentämään lähdekoodia sitä mukaan, kun sitä kirjoitetaan. Intellisense tarjoaa käyttäjälle käytettävissä olevia muuttujia ja funktioita, kuten kuviossa 10 on esitetty. Ottaen huomioon ovatko ne näkyvissä nykyisessä kontekstissa. Näin helpottaen lähdekoodin kirjoitusta ja estäen yleisiä kirjoitusvirheitä, jotka voisivat haitata ohjelman toimintaa.



The screenshot shows a code editor with the following code:

```

852     string uname = User.Identity.Name;
853
854     var person = ctx.person.Where(person => person.username == una);
855
856
857
858
859
860

```

The variable `uname` is highlighted in blue. A dropdown menu is open, showing the following options:

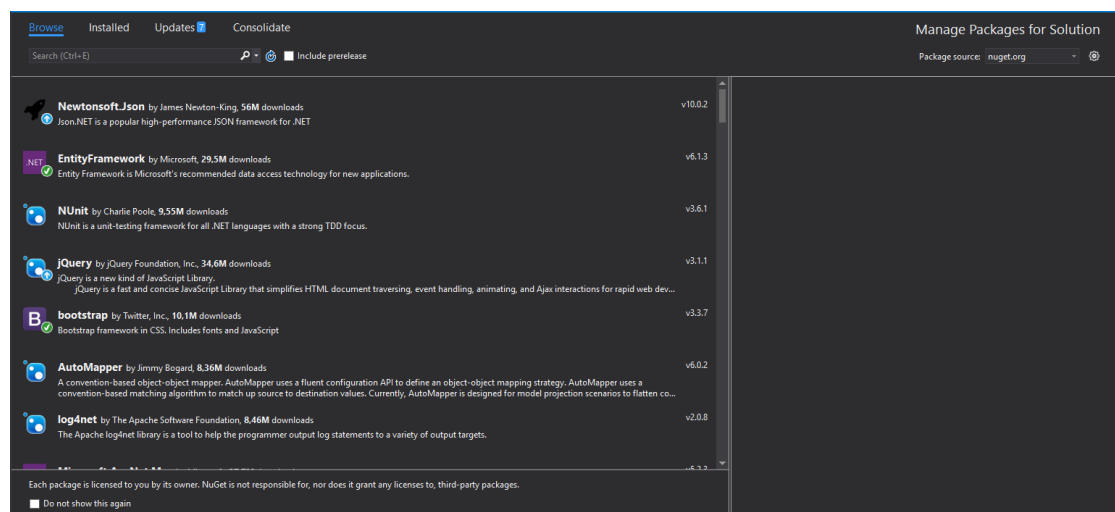
- `uname` (local variable) string uname
- `UnauthorizedAccessException`

The dropdown menu also contains several icons for navigation and editing.

Kuvio 10. Intellisense täydentää edellisellä rivillä alustetun muuttujan nimen nykyisellä rivillä.

Yksi tärkeimmistä ominaisuuksista Visual Studiassa opinnäytetyötä tehdessä on sen Nuget-paketinhallintaominaisuus, jonka avulla Visual Studiolla tehtyihin ohjelmistoprojekteihin voidaan ladata internetistä (www.nuget.org) kirjastoja ja ohjelmistokehyskirjastoja, joita voidaan suoraan lataamisen jälkeen alkaa käyttämään ohjelmistoprojektissa. Opinnäytetyön tekemisessä Nuget-paketinhallinnan avulla ladattiin Entity Frameworkin ja SignalR:n toimintaan tarvittavien tiedostojen lisäksi. Myös MySQL tietokantojen tukemiseen Visual Studiassa vaadittavat ajurit sekä JSON-datan jäsennointiin käytettävä Newtonsoft- ja muutamien sivuilla esiintyvien AJAX toimintojen mahdollistamiseen tarkoitettu AJAX Control Toolkit-kirjastot.

Nuget-paketinhallintaa voidaan käyttää kahdella tavalla, ensimmäinen on kuviossa 11 esitetty Visual Studion sisäinen Nuget-paketinhallinta, jolla voidaan hakea Nuget-paketteja Visual Studiassa sillä hetkellä avatulle projektille. Tällä tavalla Visual Studio avaa uuden välilehden, jossa käyttäjä voi valita esitetyistä Nuget-paketeista haluamansa kirjaston tai ohjelmistokehyskirjaston ja asentaa ne avattuun projektiin.



Kuvio 11. Visual Studion sisällä avattu Nuget-paketinhallintaominaisuus.

Toinen keino on Visual Studiassa avattava Nuget-konsoli. Tämän käyttämiseksi käyttäjä voi itse käydä verkkoselaimella Nugetin kotisivujen ”Galleria”-sivulla (<https://www.nuget.org/packages>) ja etsiä itse haluamaansa pakettia ja paketin versiota. Löydettyään haluamansa paketin käyttäjä voi kopioida paketin sivulta löytyvän komennon, joka syötetään Visual Studiassa avattuun konsoliin. Komento on yleensä seuraavanlainen:

Kyseisessä komennossa siis asennetaan MySQL-tietokantojen tukemiseen tarvittavat ajurit Visual Studiossa toteutettavaan projektiin, Ajurien haluttu versio ilmoitetaan parametrina "Version".

Visual Studion tarjoamaa Debug-ominaisuutta käytettiin myös opinnäytetyössä havaittujen bugien paikantamiseen ja korjaamiseen. Tämän toiminnon avulla kehittäjä voi pysäyttää ohjelman pyörimisen tietylle riville ja tarkastella esimerkiksi tässä ohjelman osassa olevien muuttujien arvoja. Näin tarkistaen ohjelmalogiikan toimivuuden.

5.4 MySQL Workbench

MySQL Workbench on työkalu tietokantojen suunnitteluun, joka integroi SQL kehityksen, hallinnoinnin, tietokantojen suunnittelun, luonnin ja ylläpidon yhdeksi ohjelmointiympäristöksi MySQL-tietokantajärjestelmälle. (MySQL Workbench. 2017.)

MySQL Workbench yksinkertaistaa tietokannan suunnittelua ja toteutusta tarjoamalla graafisen käyttöliittymän, jolla tietokannan kehittäjä voi mallintaa ja tallentaa tietokannan taulut, taulujen määritteet sekä kaikki relaatiot taulujen välillä. Tällaista mallia kutsutaan skeemaksi. Nämä tietokannan skeemat voidaan tallentaa ja niistä voidaan suoraan luoda taulut haluttuun tietokantaan.

MySQL Workbenchiä käytettiin opinnäytetyön tietokannan suunnitteluun, luontiin ja hallintointiin. MySQL Workbenchiä käytettiin myös tietokannan versionhallintaan, tallentamaan tietokannan uusista versioista varmuuskopiot ja tekemään tietokannan migraatiot.

MySQL Workbenchiä käytettiin myös testaukseen. Sillä tarkastettiin ovatko sovelluksessa tietokantaan lähetetyt syötteet saapuneet tietokantaan ja myös ovatko Entity Frameworkilla tehdyt muutokset tallentuneet tietokantaan.

5.5 Github

Github on verkkosivusto, joka toimii isännöintipalveluna Git-versionhallintaa käyttäville ohjelmistoprojekteille. Se tarjoaa saman hajautetun versionhallintajärjestelmän ja lähdekoodin hallinnoinnin kuin Git, mutta näiden lisäksi se sisältää myös omia ominaisuuksiaan, kuten ryhmätyöskentelyyn käytettävä bugien valvonta, toimeksiantojen hallinnointi ja Wiki-sivusto jokaiselle projektille.

Githubin käyttö on ilmaista julkisesti nähtävissä oleville ohjelmistoprojekteille, mutta se tarjoaa myös maksullisia palveluita, kuten yksityiset ohjelmavarastot ja verkkosivujen isännöintipalvelut. Opinnäytetyön kanssa on kuitenkin käytetty ainoastaan Githubin ilmaisia palveluita ohjelmistokoodin versionhallinnalle.

Opinnäytetyötä tehdessä käytettiin Microsoft Visual Studion ladattavissa olevaa Git-ominaisuutta, jolla ohjelmistoprojektiin tehdyn työn yhteensopivuus GitHubissa olevan koodin kanssa voidaan tarkistaa ja tehty työ voidaan ladata GitHub palveluun käyttämällä Visual Studion käyttöliittymää.

5.6 Microsoft Azure

Microsoft Azure on Microsoftin pilvipalvelu, joka on kehitetty verkkosovellusten rakentamiseen, käyttöönottoon ja hallinnointiin Microsoftin hallinnoimien datakeskusten kautta. Opinnäytetyössä Microsoft Azurea käytetään verkkosovelluksen ja tietokannan julkaisemiseen verkossa.

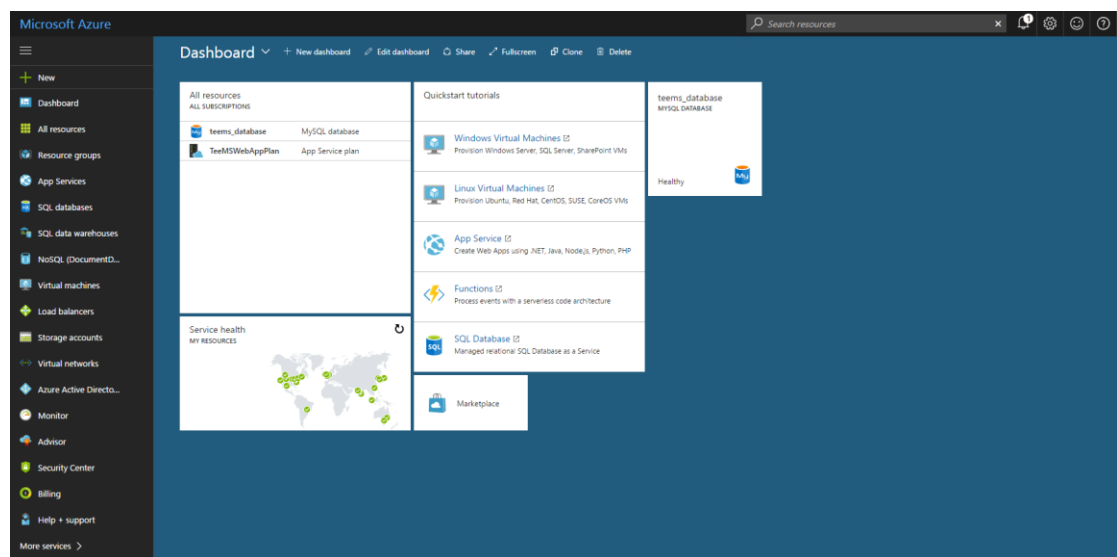
Pilvipalveluiden käyttö on yleistynyt ilmiö IT-maailmassa. Sillä niiden palvelut sopivat erityisesti yrityksille, joiden sovelluksilla on kasvavia tai hyvin vaihtelevia kaistanleveyden vaatimuksia, pilvipalveluissa kapasiteetin skaalaaminen tarpeiden mukaan on helppoa.

Pilvipalveluiden hyöty on myös niiden työskentelyä helpottavat ominaisuudet, niin kauan kuin kehittäjällä on yhteys internettiin, hän voi työstää sovellusta mistä tahansa.

Microsoft Azure-tallennustila (engl. Azure Storage) on saavutettavissa laajalla alueella ja kaikista sovelluksista, oli se sitten pilvessä, pöytäkoneessa, omalla palvelimella tai mobiililaitteella pyörivä sovellus. Azuren tallennustila tukee sekä Windows- että

Linux-käyttöjärjestelmiä ja useita ohjelmointikieliä, kuten .NET, Java, Node.js, Python, Ruby, PHP ja C++. (Macy, M. Gries, W. Myers, T. Archer, Tom. & Shahan, R. Introduction to Microsoft Azure Storage. 2017.)

Opinnäytetyön tekemisessä Azure-palveluun luotiin ensimmäiseksi käyttötili, jonka kautta kaikki mahdolliset asioinnit Azuren kanssa tehtiin, kun Azureen oli luotu käyttötili, päästiin Azuren kojelaudalle (engl. Dashboard), jossa opinnäytetyön sovelluksen vaatimat resurssit voitiin luoda, kojelauta on esitetty kuviossa 12.



Kuvio 12. Microsoft Azuren kojelauta, jossa näkyy opinnäytetyön käyttämä MySQL-tietokanta kiinnitettynä etusivulle.

Jotta Azuressa voidaan ryhmittää tietyt resurssit osaksi yhtä ohjelmistokokonaisuutta, täytyy niille luoda oma resurssiryhmä (engl. Resource Group), opinnäytetyön käyttämien resurssien kohdalla luotiin uusi resurssiryhmä, jonka nimeksi annettiin Data_storage.

Opinnäytetyön verkkosovelluksen julkaiseminen ja tietokannan asentaminen Azure-pilvipalveluun on kuvattu tarkemmin luvussa ”6.10 Sovelluksen julkaiseminen verkossa”.

6 Opinnäytetyön toteutus

6.1 Yleistä

Sovelluksen toteutukseen kuuluu varsinainen ohjelmointi-, kehitys- ja testaustyö, jolla sovellus saadaan kehittäjän haluamaan tilaan. Myös toteutus vaatii oman suunnittelunsa, varsinkin, jos kehittäjänä on ryhmä. Sillä ennen kuin toteutusta voidaan kunnolla alkaa tekemään, pitää kehittäjän tietää mitkä osat kokonaisuudesta toteutetaan ja missä järjestyksessä.

Sovelluksen toteutus käynnistetään yleensä aikataulutuksella, jossa jokaiselle kehittäjälle annetaan tehtävä ja aikaraja tehtävän tekemiselle, mutta koska opinnäytetyön sovelluksen tekemiseen osallistui vain yksi tekijä, oli olennaisempaa keskittyä jaksotamaan työn tekemiseen oleva aika niin, että mahdollisimman paljon moniosaisesta sovelluksesta saataisiin valmiiksi.

Vaikka opinnäytetyö koostuu monesta verkkosivusta, osa verkkosivujen toiminnasta on keskeisesti yhteydessä muiden sivujen sekä erinäisten luvussa 5 esiteltyjen teknologioiden toimintaan. Tämän vuoksi toteutusta ei esitellä sivu kerrallaan, vaan esittämällä toteutus sivujen toiminnan kautta.

6.2 Toteutuksen suunnittelu

Sovelluksen toteutuksen suunnittelu aloitettiin kirjaamalla sovelluksen päätehtävät Trello-nimiseen, verkkopohjaiseen projektinhallintasovellukseen ja asettamalla niille aika-arvio, eli kuinka kauan niiden valmistuminen kestää. Nämä tehtävät jaettiin myös pienempiin työtehtäviin tarkistuslistan muodossa. Tätä tarkistuslistaa päivitettiin aina, kun toteutus eteni.

Useasti toteutuksen aikana esiin nousee ongelmia tai esteitä, jotka voivat hidastaa tiettyjen osien valmistumista. On myös mahdollista, että toiset osat valmistuvat aikaisemmin kuin oli oletettu, mikä vapauttaa aikaa joko esiin nousseiden ongelmien tai muiden ominaisuuksien parantamiseen. Tämän vuoksi toteutuksen aikataulutusta ei voida täysin suunnitella kattamaan toteutuksen pienimpiäkin yksityiskohtia.

Sovelluksen toiminnan kannalta oli olennaista, että tietokantaan voitiin tallentaa käyttäjän tiedot. Sillä sovelluksen ei ole tarkoitus olla käytettävissä rekisteröimättömille käyttäjille. Tästä johtuen oli luontevaa aloittaa sovelluksen toteutus tietokannan tekemisellä.

Seuraavaksi toteutuksen suunnittelussa päätettiin, mitkä verkkosovelluksen osan ominaisuudet toteutetaan ensimmäisenä ja mitkä jätetään myöhemmäksi. Tämä jako toteutettiin versioimalla sovellus, joista ensimmäiseen versioon, eli versio 0.1, tulisi projektien, ryhmien sekä molempiin kuuluvien jäsenten lisäys ja poisto. Näiden lisäksi myös sisäänkirjautuminen ja uuden käyttäjäprofiilin rekisteröiminen tarkoitettiin lisättäväksi tässä versiossa.

seuraavaan versioon sisällöksi tulisi projektin sisäisten työtehtävien ja niiden sisältämien komponenttien lisääminen ja integroiminen ensimmäisen version ominaisuuksien kanssa, versio kahdessa lisättäisiin myös kommentointi-ominaisuus projekteihin ja työtehtäviin.

Verkkosovelluksen kolmas versio keskittyi lisäämään Chat-ominaisuuden, jota ryhmien jäsenet voivat käyttää ryhmien sisäiseen keskusteluun, Chat suunniteltiin käyttämään SignalR-tekniikkaa, jonka avulla mahdollistettiin viestien lähettäminen keskustelun muille jäsenille aina, kun ne tulivat saataville palvelimella. Chat-ominaisuus toimii niin, että käyttäjä keskustelee palvelimen kanssa käyttämällä SignalR:n Hub-luokkaa, nimeltään TeeMsHub, joka toimii ohjelmointirajapintana Client puolen JavaScriptin ja palvelimen C# ohjelmakoodin välillä.

Koska verkkosovelluksen kolmannessa versiossa toteutettu Chat-ominaisuus valmistui odotettua aikaisemmin, päätettiin SignalR:n avulla lisätä reaaliaikaista toiminnallisuutta myös projektien ja ryhmien sivuille. Tämä reaaliaikaisuus suunniteltiin koskemaan ainakin projektien kommenttien ja edistymisen päivittämistä sekä ryhmien ja projektien nimien ja kuvauksien muuttamista. Näiden ominaisuuksien lisääminen tulisi kattamaan sovelluksen neljännen version.

Lopullinen vaihe toteutuksessa suunniteltiin olevan opinnäytetyön verkkosovelluksen julkaiseminen Microsoft Azure-pilvipalveluun, joka vaatisi myös samassa sijainnissa olevan MySQL-tietokannan.

6.3 Rekisteröinti ja sisäänkirjautuminen

6.3.1 Käyttäjän rekisteröinti verkkosivulla

Käyttäjätunnuksen rekisteröinti tapahtuu Register.aspx-sivulla, jonne käyttäjä voi navigoida sisäänkirjautumissivulta. Käyttäjä täyttää rekisteröintisivulla tietonsa ja antaa itselleen käyttäjätunnuksen nimen ja salasanan. Tämän jälkeen sovellus tallentaa annetut tiedot tietokantaan, paitsi salasanan, joka ajetaan ensin salausalgoritmin lävitse ns. suolauksen kanssa.

Käyttäjältä kysytään käyttäjätunnuksen nimi, henkilön oma etu- ja sukunimi, sähköposti sekä salasana kahdesti. Salasana pyydetään käyttäjältä kahdesti kirjoitusvirheiden aiheuttamien ongelmien ehkäisemiseksi, kun käyttäjä on kirjoittanut salasanan kahdesti omiin tekstialueisiinsa, molempia tekstejä verrataan toisiinsa ennen tietojen tallennusta tietokantaan, jos tekstit eivät täsmää, tietoja ei tallenneta ja tekstien eroavaisuudesta ilmoitetaan käyttäjälle.

6.3.2 Käyttäjän sisäänkirjautuminen

Käyttäjän sisäänkirjautuminen tapahtuu sovelluksen Login.aspx-sivulla. Käyttäjä syöttää nimimerkkinsä ja salasansansa sovellukseen, jonka jälkeen tietokannasta haetaan nimimerkkiä vastaavat käyttäjätunnuksen tiedot ja salasanan kryptattu versio sekä sen suolaukseen käytetty merkkijono.

Käyttäjän syöttämä salasana ajetaan myös käyttäjän rekisteröinnissä käytettävän salausalgoritmin lävitse suolauksen kanssa ja siitä saatua tulosta verrataan tietokannassa olevaan kryptattuun salasanaan. Mikäli molemmat ovat yhteneviä keskenään, Käyttäjä ohjataan sovelluksen pääsivulle ja käyttäjälle luodaan autentikaatioväste, jonka avulla sovellus tunnistaa käyttäjän todennetuksi käyttäjäksi.

Käyttäjä ohjataan pääsivulle, jossa sisäänkirjautumisen yhteydessä luodun autentikaatiovästeen avulla heille noudetaan tietokannasta kaikki ryhmät, joissa käyttäjä on jäsenenä sekä projektit, joita käyttäjä on työstämässä. Pääsivulle saavuttuaan käyttäjälle avautuu myös mahdollisuus käyttää sivun vasemmassa laidassa olevaa navigointivalikkoa.

Salausalgoritmeilla salattuja salasanoja ei yleensä yritetä murtaa algoritmin matemaattisten ominaisuuksien kautta, joten parhaimman salauksen aikaansaaminen ei välttämättä ole niin yksiselitteistä, kuin kaikkein uusimman salausalgoritmin käyttäminen. Esimerkiksi SHA-256 salausalgoritmin muuttaminen SHA-512 algoritmiin ei tee salasanojen murtamisesta huomattavasti hankalampaa. Sillä salasanojen murtamiseen käytetään useasti ns. ”arvaa ja kokeile” tekniikkaa, jossa salasana arvataan, ajetaan salausalgoritmin lävitse ja katsotaan, oliko arvaus oikein. Tämän kaltaisen hyökkäyksen torjunnassa olennaista on salausprosessin hidastaminen, jos salaus kestää kauemmin, kestää jokaisen arvauksen tarkastaminen kauemmin ja hyökkääjää hidastamalla rajoitetaan, kuinka monta arvausta he voivat tehdä.

Opinnäytetyössä käyttäjän salasanan salaamiseen käytetään BCrypt-nimistä salausalgoritmia. Salasanojen salaamiseen käytettävien algoritmien, kuten BCryptin, PBKDF2:n ja SCryptin yhteinen ominaisuus on, että ne kaikki ajavat salausfunktioita monia tuhansia kertoja yhdellekin salasanalle. Tämä tarkoittaa, että mahdollinen hyökkääjä joutuu käyttämään yhtä salasanan arvausta kohti, huomattavasti enemmän aikaa ja resursseja.

6.4 Käyttäjän pääsivu

Pääsivulla, eli Home.aspx-sivulla, käyttäjä voi tarkastella kaikkia ryhmiä ja projekteja joihin he kuuluvat. Ryhmät esitetään listana, jossa jokaisen ryhmän kohdalla esitetään ryhmän nimi, ryhmän profiilikuva sekä linkki ryhmäsivulle. Projektit esitetään myös listana, nimen, projektikuvan ja projektisivun linkin kanssa. Lisäksi sivulla on napit uusien projektien ja uusien ryhmien luomista varten. (Ks. liite 1)

Pääsivun oikeassa laidassa näytetään käyttäjälle lähetetyt kutsut liittyä kutsussa esitettyyn ryhmään. Käyttäjä voi halutessaan joko hyväksyä tai kieltäytyä tällaisesta kutsusta, kieltäytyessään kutsusta, kutsu poistetaan käyttäjän pääsivulta, mutta hyväksyessään kutsun käyttäjä liitetään osaksi kutsussa esitettyyn ryhmään sekä kaikkiin ryhmän työstämiin projekteihin.

Kaikki käyttäjän pääsivun näyttämät tiedot yksilöidään käyttäjälle palvelimella käyttämällä sisäänkirjautumisen yhteydessä luotua autentikaatioevästettä, josta on mahdollista saada käyttäjän nimimerkki. Tämän nimimerkin avulla tietokannasta haetaan

Entity Frameworkin ja LINQ-kyselyiden avulla kaikki ryhmät ja projektit, joihin käyttäjä kuuluu sekä kaikki käyttäjälle osoitetut kutsut liittyä ryhmiin.

Käyttäjän ryhmät ja projektit esitetään sivulla luomalla ohjelmakoodissa tarvittavat HTML- ja ASP-elementit. Samalla täyttäen ne tietokannasta haetulla tiedolla. Koodissa elementeille annetaan myös CSS-tyylit, jotta ne näkyvät oikein käyttäjän pääsivulla. Samankaltaista tapaa käytetään myös projektit ja ryhmät listaavilla Projects.aspx- ja Groups.aspx-sivuilla, jotka ovat saavutettavissa verkkosivun vasemmassa laidassa olevasta navigaatiopalkista. Ryhmien ja projektien näyttämiseen käytetty koodi on nähtävissä liitteenä. (Ks. liitteet 2, 3 ja 4)

6.5 Ryhmät verkkosivulla

Ryhmien luonti tapahtuu Home.aspx-sivulla. Käyttäjä painaa 'Create Group'-nappia, jonka jälkeen sovellus luo uuden ryhmän sekä yhteyden ryhmän luoneen käyttäjän ja luodun ryhmän välillä tietokantaan. Uuden ryhmän nimeksi asetetaan "New Group X", jossa X on ryhmän Id tietokannassa.

Tämän jälkeen käyttäjä ohjataan suoraan uudelle ryhmäsivulle, jossa käyttäjä voi muuttaa ryhmän nimeä ja profiilikuvaa, lisätä ja poistaa jäseniä ryhmään ja siirtyä katsomaan ryhmän jäsenien tietoja. (Ks. liite 3)

Kun ryhmä on asetettu työstämään projektia, on ryhmäsivulla myös mahdollista katsoa projektin kuvaus ja projektin työtehtävien edistyminen, projektin kuvaus ja työtehtävien edistyminen näytetään kuvion 13 mukaisesti.

Group projects

My First Project ▾

Project info

| Project description | Project assignments | | | | | | | | | | | | | | | | | | |
|--|---|--|--------------|--|---|--|--|----------------|--------------|--|---|--|--|----------------|--------------|--|---|--|--|
| The goal of this project is to build a house | <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">Build the floor</td> <td style="padding: 2px;">50% Complete</td> <td style="padding: 2px;"><div style="width: 50%; height: 10px; background-color: #0070C0; border: 1px solid #000;"></div></td> </tr> <tr> <td colspan="3" style="padding: 2px;">Assignments: 4. Finished assignments: 2</td> </tr> <tr> <td style="padding: 2px;">Build the roof</td> <td style="padding: 2px;">80% Complete</td> <td style="padding: 2px;"><div style="width: 80%; height: 10px; background-color: #0070C0; border: 1px solid #000;"></div></td> </tr> <tr> <td colspan="3" style="padding: 2px;">Assignments: 5. Finished assignments: 4</td> </tr> <tr> <td style="padding: 2px;">Build a window</td> <td style="padding: 2px;">25% Complete</td> <td style="padding: 2px;"><div style="width: 25%; height: 10px; background-color: #0070C0; border: 1px solid #000;"></div></td> </tr> <tr> <td colspan="3" style="padding: 2px;">Assignments: 4. Finished assignments: 1</td> </tr> </table> | Build the floor | 50% Complete | <div style="width: 50%; height: 10px; background-color: #0070C0; border: 1px solid #000;"></div> | Assignments: 4. Finished assignments: 2 | | | Build the roof | 80% Complete | <div style="width: 80%; height: 10px; background-color: #0070C0; border: 1px solid #000;"></div> | Assignments: 5. Finished assignments: 4 | | | Build a window | 25% Complete | <div style="width: 25%; height: 10px; background-color: #0070C0; border: 1px solid #000;"></div> | Assignments: 4. Finished assignments: 1 | | |
| Build the floor | 50% Complete | <div style="width: 50%; height: 10px; background-color: #0070C0; border: 1px solid #000;"></div> | | | | | | | | | | | | | | | | | |
| Assignments: 4. Finished assignments: 2 | | | | | | | | | | | | | | | | | | | |
| Build the roof | 80% Complete | <div style="width: 80%; height: 10px; background-color: #0070C0; border: 1px solid #000;"></div> | | | | | | | | | | | | | | | | | |
| Assignments: 5. Finished assignments: 4 | | | | | | | | | | | | | | | | | | | |
| Build a window | 25% Complete | <div style="width: 25%; height: 10px; background-color: #0070C0; border: 1px solid #000;"></div> | | | | | | | | | | | | | | | | | |
| Assignments: 4. Finished assignments: 1 | | | | | | | | | | | | | | | | | | | |

Show project

Kuvio 13. Ryhmäsivulla esitetty projektin info ja edistyminen.

Ryhmiin jäsenten lisääminen tapahtuu sivun oikeassa laidassa olevasta "Add members"-painikkeesta. Tämän painikkeen painaminen avaa käyttäjien hakuun tarkoitetun tekstilaatikon, johon voidaan kirjoittaa haetun käyttäjän nimimerkki ja painaa "Search"-painiketta. Tämä tuo haetun käyttäjän sivulle taulukkomuodossa, josta nähdään myös haetun käyttäjän nimi ja sähköposti. Käyttäjän hakuun käytettävään tekstilaatikkoon voidaan syöttää myös vain osa käyttäjän nimestä. Tällä tavalla kaikki käyttäjät, joiden nimimerkissä esiintyy syötetty osa, näytetään taulukossa.

Kun käyttäjä valitsee taulukosta henkilön, jonka haluaa liittää ryhmään. Tälle henkilölle lähetetään kutsu ryhmään liittymisestä. Tämä kutsu näytetään kyseisen henkilön pääsivulla ja kun käyttäjä hyväksyy kutsun, hänet liitetään ryhmään.

Haku-ominaisuudessa käytetty taulukko on toteutettu käyttämällä ASP:n DataGrid-luokkaa. Käyttäjän syöttämä merkkijono käydään ohjelmakoodissa läpi kirjain kerrallaan ja näitä verrataan tietokannassa oleviin käyttäjiin, kaikki merkkijonoa vastaavat käyttäjät syötetään DataGrid-luokkaan datalähteeksi.

Ryhmän jäsenen poistaminen ryhmästä tapahtuu valitsemalla poistettava käyttäjä sivun oikeassa laidassa olevilla valintanapeilla ja sen jälkeen painamalla "Remove Member"-painiketta. Tämä kutsuu palvelinpuolen funktion, joka poistaa jäsenen tietokantayhteyden ryhmään ja ryhmän työstämiin projekteihin.

Ryhmän jäsenille voidaan myös asettaa uusi rooli ryhmän sisällä "Assign new role"-painikkeella. Tämä painike avaa AjaxControlToolkit-kirjaston avulla tehdyn ModalPopupExtender-elementin eli ikkunan, joka näytetään käyttäjälle muiden elementtien päällä. Tässä ikkunassa käyttäjä voi valita ryhmän jäsenelle uuden roolin ja tallentaa sen tietokantaan.

6.6 Ryhmien Chat-ominaisuus

Sovelluksen Chat-ominaisuus on saavutettavissa navigoimalla Chat.aspx-sivulle navigointipalkista. Tälle sivulla tultaessa käyttäjä liitetään osaksi sen ryhmän keskustelua, joka on lisätty tietokantaan ensimmäisenä ja jossa käyttäjä on jäsenenä. Lisäksi kaikki ryhmät joissa käyttäjä on jäsen, näytetään painikkeina Chat-ikkunan vasemmalla puolella. Näiden painikkeiden avulla käyttäjä voi vaihtaa nykyistä keskusteluryhmäänsä ja nykyisen keskusteluryhmän jäsenet näytetään Chat-ikkunan oikealla puolella. (Ks. liite 5)

Keskusteluun osallistuminen tapahtuu kirjoittamalla Chat-ikkunan alapuolella olevaan tekstilaatikkoon haluamansa viesti ja painamalla "Send"-painiketta. Tämä käynnistää JavaScript-funktion, joka SignalR:n avulla ottaa yhteyden palvelimeen, jossa se tallennetaan tietokantaan ja kuulutetaan keskustelun muille jäsenille.

Viestit tallennetaan tietokantaan, jotta aikaisemmin käyty keskustelu voidaan näyttää käyttäjille näiden liittyessä keskusteluryhmään, keskustelut tallennetaan siis yksittäisinä viesteinä yhdessä lähettäjän ja keskusteluryhmän Id:n kanssa ja nämä viestit näytetään käyttäjille, kun he navigoivat Chat-sivulle tai vaihtavat keskusteluryhmää.

6.7 Projektit ja projektien työtehtävät verkkosivulla

6.7.1 Projektit

Käyttäjä voi luoda projektin, kun hän on luonut vähintään yhden ryhmän tai liittynyt jo olemassa olevaan ryhmään. Projektin luominen tapahtuu pääsivulla olevasta "Create new project"-painikkeesta. Tämä painike vie käyttäjän uuden projektin luontisivulle, jossa käyttäjä syöttää uuden projektin tiedot, kuten projektin nimen, ku-

vauksen ja projektikuvan URL-osoitteen sekä projektin eräpäivän ja ryhmän, joka projektia alkaa työstää.

Kun projekti on luotu, käyttäjä siirretään uudelle projektisivulle, jossa käyttäjä voi muokata projektin nimeä ja vaihtaa projektin kuvaa sekä lisätä ja poistaa ryhmiä avustamaan projektin tekemisessä. Lisäksi projektin sivulta voidaan siirtyä luomaan uusia projektin työtehtäviä tai katsomaan olemassa olevia työtehtäviä.

Projektisivulla käyttäjä voi myös nähdä tietoa projektin etenemisestä. Sivulla esitetään projektin luonnin aikana asetettu eräpäivä, kaikkien projektin työtehtävien edistymisen prosentuaalisesti perustuen työtehtävän valmistuneiden komponenttien määrään. (Ks. liite 6)

Ryhmien lisääminen ja poistaminen toteutettiin samankaltaisesti ryhmän jäsenten lisäämisen ja poistamisen kanssa Group.aspx-sivulla. Käyttäjän painaessa ryhmän lisäämiseen käytettävää painiketta, avautuu sivulle haku-toiminto, jolla haetaan henkilöiden sijaan ryhmiä. Projekteihin haku-toiminnon kautta lisätyt ryhmät saavat tietokannassa "supporting"-ominaisuuden, joka tarkoittaa, että lisätty ryhmä on projektissa avustavassa roolissa, ainoastaan projektin luonnin yhteydessä lisätty ryhmä ymmärretään projektin pääryhmäksi.

Projektisivulla voidaan myös kommentoida projektia. Käyttäjä syöttää viestinsä sivun alalaitaan sijoitettuun tekstilaatikkoon ja painettuaan "Send"-painiketta viesti tallennetaan projektin kommenttina tietokantaan. Lisättyjä kommentteja voidaan myöhemmin muokata ja poistaa. Kommentointi-toiminto on esitetty kuviossa 14.

Comments

Posted by Admin on 2.5.2017 00.06
How is the project progressing?

Posted by CommonUser on 2.5.2017 00.07
We should have a new release version ready next week.

Edit Delete

We should have a new release version ready next week.

Send Comment

Kuvio 14. Kommentointi-toiminto Project.aspx-sivulla.

Projektin kommentteja voidaan tehdä myös Chat-ominaisuuden kautta, kirjoittamalla Chat-viestin eteen ”@PROX”, jossa X on projektin tunnisteen numero, joka on esitetty projektisivulla, projektin nimen alapuolella.

Valmistunut tai muuten keskeytetty projekti voidaan arkistoida, arkistoituihin projekteihin ei voida tehdä enempää muutoksia, joten projektin arkistointi toimii käytännössä projektin viimeistelynä, projekti voidaan poistaa arkistoidusta tilasta oikean roolin ryhmässä omaavan käyttäjän toimesta.

6.7.2 Projektien työtehtävät

Projektien työtehtävien luonti alkaa projektin sivulta. Käyttäjä luo uuden työtehtävän painamalla ”Create new assignment”-painiketta, jonka jälkeen hänet viedään työtehtävän luomiseen tarkoitettulle sivulle (CreateAssignment.aspx). Tällä sivulla käyttäjä syöttää työtehtävän tiedot, kuten nimen ja kuvauksen sekä eräpäivän, jonka täytyy olla aikaisempi kuin projektin eräpäivä. Lisäksi työtehtävän luonut käyttäjä lisää jäseneksi työtehtävään, joka valitaan kaikista projektia työstävistä jäsenistä.

Kun uusi työtehtävä on luotu. Käyttäjä viedään uudelle työtehtävisivulle, jossa käyttäjä voi tarkastella työtehtävän tietoja sekä lisätä ja poistaa työtehtävän komponent-

teja, joiden avulla työtehtävän edistyminen määritetään ja lasketaan työtehtävän projektin ja tätä projektia työstävien ryhmien sivuilla. (Ks. liite 8)

Työtehtävään luotujen komponenttien tietoja voidaan muuttaa painamalla työtehtäväsivulla näkyvän komponentin nimeä. Tämä avaa kuviossa 15 esitetyn ikkunan, jossa käyttäjä voi muuttaa komponentin nimen, komponenttia työstävät jäsenet ja komponentin valmiustason, kun käyttäjä on tehnyt halutut muutokset, painamalla ”Save changes”-painiketta, tehdyt muutokset tallennetaan tietokantaan.

Kuvio 15. Työtehtävän tietojen muuttamiseen käytettävä ikkuna.

Työtehtävän komponentin muuttaminen valmiiksi kuviossa 15 kuvatun ”Completed”-valintanapin avulla, päivittyy työtehtävän edistyminen myös työtehtävän projektin sivulla, SignalR:n kautta.

Työtehtäväsivulla on samankaltainen kommentointi-toiminnallisuus kuin projektin sivulla sekä samankaltainen jäsenten lisääminen kuin Group.aspx-sivulla. Työtehtävien poistaminen on tehtävä työtehtäväsivulla. Käyttäjä painaa sivulta löytyvää ”Delete Assignment”-painiketta, joka käynnistää palvelinpuolen funktion, joka poistaa työtehtävän tietokannasta.

6.8 Käyttäjätietojen käsittely verkkosivulla

Käyttäjät voivat muuttaa oman käyttäjätilinsä tietoja Profile.aspx-sivulla. Muuttaakseen omia käyttäjätietojaan profiilisivulla. Käyttäjä syöttää uuden tiedon sille tarkoitettuun tekstilaatikkoon. Esimerkiksi, jos käyttäjä haluaa muuttaa käyttäjätilinsä nimimerkkiä, hän syöttää uuden nimimerkin ”User name” otsikoituun tekstilaatikkoon ja painaa ”Edit Profile”-painiketta. Tämän jälkeen uusi tieto tallennetaan tietokantaan.

Salasanan muuttaminen toimii samalla tavalla, mutta salasanan vaihtaminen vaatii ylimääräisen tekstilaatikon täyttämistä. Tämä vaaditaan samasta syystä kuin käyttäjätilin rekisteröinnin aikana. Se estää käyttäjän tekemiä kirjoitus- tai muita virheitä salasanan syöttämisen aikana.

Sovelluksessa on myös mahdollista käydä katsomassa muiden käyttäjien profiilia. Tämä voidaan tehdä ryhmän sivulta, jossa käyttäjä jonka profiilia yritetään käydä katsomassa, on jäsenenä. Ryhmän sivun oikeassa laidassa, ryhmän jäsenen poistopainike oikealla puolella, on "Show User"-painike, jota painamalla käyttäjä viedään sillä hetkellä valittuna olevan käyttäjän infosivulle. Käyttäjän infosivua ei näytetä, jos valittu käyttäjä on asettanut sen yksityiseksi profiilisivullaan.

6.9 Tietokanta

6.9.1 Tietokannan asennus ja käyttöönotto sovelluksessa

Tietokannan toteutuksen aloittamiseksi täytyi ensimmäiseksi tehdä sille sopiva palvelinsijainti, kuten tietokantaa suunniteltaessa oli päätetty, asennettiin MySQL-palvelinohjelma paikallisesti omalle työkoneelle, jotta kaikki mahdolliset lokitiedostot ovat saatavilla mahdollisten virheiden tapahtuessa.

Tietokannan toteutus aloitettiin asentamalla MySQL Installer-ohjelmistopaketti. Tässä ohjelmistopakettissa on sisällytetty MySQL Workbench ohjelma, jolla voidaan toteuttaa tietokannan malli ja sen mallin voi ohjelman avulla kääntää suoraan tietokannan luonti skriptaksi, jonka avulla tietokantaan voidaan lisätä taulut ja relaatiot.

MySQL Workbench ohjelmalla luotiin skeema jonka käyttö pohjautuu kolmeen päätauluun, jotka nimettiin Person, eli henkilö, Group, eli ryhmä ja Project, eli projekti. Päätaulujen lisäksi tietokanta perustuu viiteentoista "sivutauluun", joiden toiminta varsinaisella verkkosivulla perustuu päätaulujen toimintaan, lukuun ottamatta organization-taulua, jonka toiminnallisuutta ei sovelluksessa ole implementoitu ajanpuutteen vuoksi. (Ks. liite 9)

Tietokannan skeeman käyttö edellyttää, että tietokanta luodaan sille määritettyyn paikkaan. Tämän jälkeen Workbench ohjelman avulla tuotiin suunnitellut taulut, taulujen attribuutit ja relaatiot tietokantaan. Samalla Workbench loi näistä tiedoista

luontiskriptan, joka kopioitiin tekstitiedostoon. Sen avulla tietokanta voidaan hetkes-
sä luoda uudestaan, jos jonkin ongelman ratkaiseminen sitä vaatii.

Kun tietokantaan on lisätty taulut ja relaatiot, siihen syötettiin kuvion 16 mukaiset
käyttäjän roolien tiedot ja ryhmän sisäisten jäsenten roolien tiedot

```

}/*
User roles:
-*/

INSERT INTO `H3387`.`role` (`name`, `description`, `class`) VALUES ('Administrator', 'Highest Possible rank, only one should exist at any given time', '1');
INSERT INTO `H3387`.`role` (`name`, `description`, `class`) VALUES ('Moderator', 'Second highest rank, can delete basic users', '2');
INSERT INTO `H3387`.`role` (`name`, `description`, `class`) VALUES ('User', 'Basic user rank, given to anyone upon completing registration', '3');

}/*
Group_Member roles:
-*/

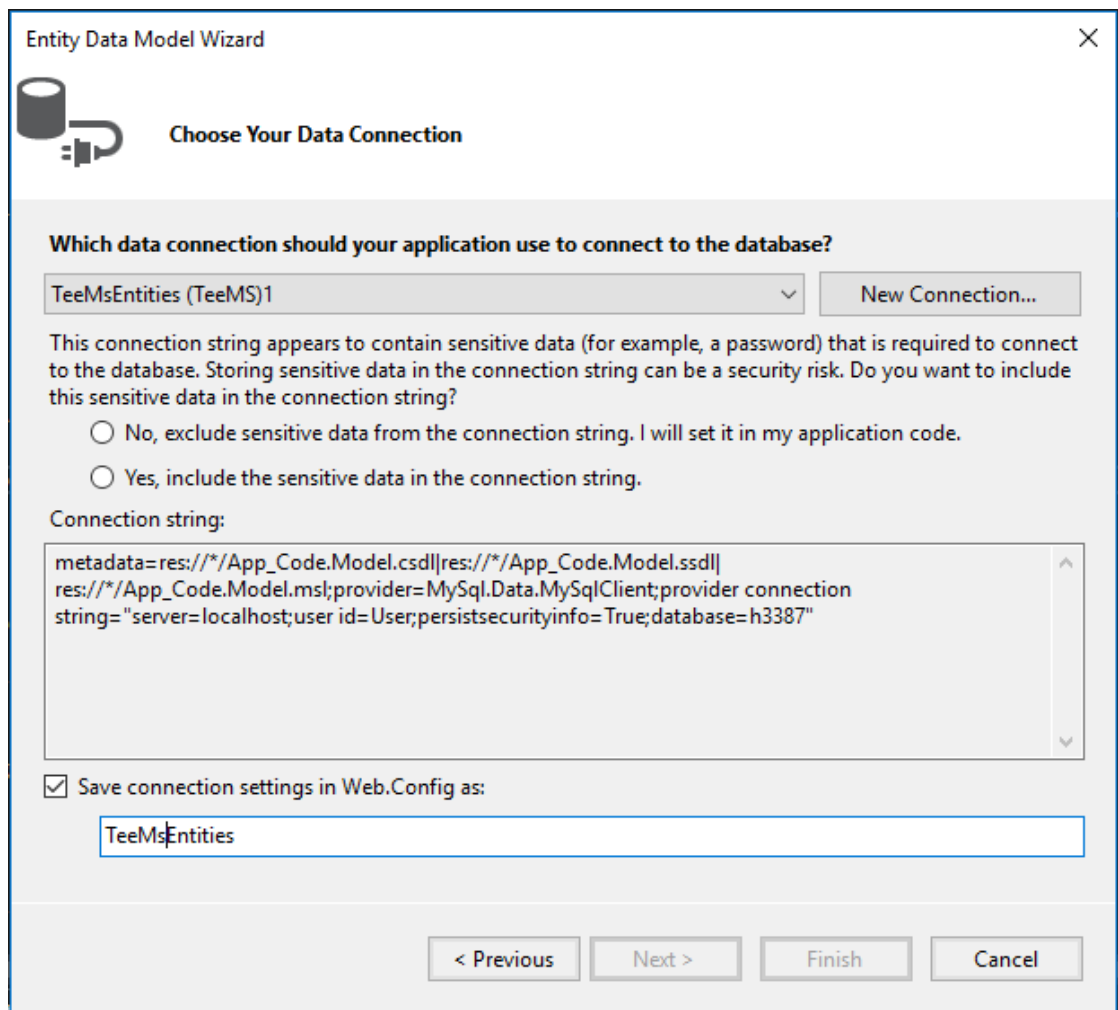
INSERT INTO `H3387`.`group_role` (`name`, `description`, `class`) VALUES ('Group Member', 'Basic member of the group', '4');
INSERT INTO `H3387`.`group_role` (`name`, `description`, `class`) VALUES ('Group Moderator', 'Promoted member of the group', '3');
INSERT INTO `H3387`.`group_role` (`name`, `description`, `class`) VALUES ('Group Administrator', 'Group leader', '2');
INSERT INTO `H3387`.`group_role` (`name`, `description`, `class`) VALUES ('Group Creator', 'Original creator of the group', '1');

```

Kuvio 16. Käyttäjän ja ryhmän jäsenten roolit tietokannassa.

Tietokannan käyttäminen sovelluksessa vaati myös sovelluspuolella tiedostojen asen-
tamista ja säätämistä. Ensimmäiseksi täytyi varmistaa, että sovelluksessa on käytössä
ajan tasalla olevat MySQL-ADO.NET ajurit. Tämä tehtiin Käyttämällä Visual Studio
IDE:n mukana tulleella Nuget-paketinhallinnalla, josta kerrottiin enemmän luvussa
”5.3 Microsoft Visual Studio”.

Seuraavaksi täytyi Microsoft Visual Studiossa lisätä opinnäytetyön projektiin
”ADO.NET Entity Data Model”-tiedosto (EDM). Tätä tiedostoa luodessa tiedostolle
annettiin myös sovelluksen käyttämän tietokannan yhteystiedot. Näistä tiedoista
tiedosto osaa automaattisesti luoda tietokantayhteyteen vaadittavan ”Connection
string”-nimisen merkkijonon, jonka Visual Studio osaa syöttää web.config-nimiseen
asetustiedostoon, kuten kuviossa 17 on esitetty. Samalla luotiin myös Entity Fra-
meworin käyttämät kontekstitiedostot.



Kuvio 17. EDM-tiedoston luontiin käytetty asennuskäyttöliittymä, jossa on nähtävissä paikalliseen tietokantayhteyteen käytetty "Connection String"-merkkijono.

edellä mainittu EDM-tiedoston luonnin jälkeen, oli sen sisällä nähtävissä diagrammi siihen yhdistetystä tietokannasta. Tätä diagrammia päivitettiin Visual Studion käyttöliittymän kautta, kun tietokantaan tehtiin muutoksia.

6.9.2 Tietokannan arkkitehtuuri

Person-taulu

Person tauluun tallennetaan käyttäjän perustiedot, kuten nimimerkki, käyttäjän oma nimi, sähköposti ja yksityisyysarvo. Näiden lisäksi tallennetaan käyttäjätunnuksen luontiaika ja käyttäjätietojen viimeisin muutosajankohta.

Person taulu sisältää myös viiteavaimen Role-nimiseen tauluun josta käyttäjälle haetaan rooli, kuten Moderaattori tai Administraattori, joilla hallitaan verkkosivun toi-

mintaa. Role-taulu sisältää roolin nimen, kuvauksen roolin tarkoituksesta ja luokan, joka määrittää, minkä tasoinen rooli on.

Login-taulu

Login-taulu on erillinen taulu, jolla ei ole mitään relaatioita muiden taulujen kanssa. Sitä käytetään ainoastaan säilyttämään käyttäjän suolattua ja salausalgoritmillä suojattua salasanaa sekä käyttäjän nimimerkkiä, joilla suoritetaan käyttäjän autentikointi sisäänkirjautumisen yhteydessä.

Group-taulu

Group-taulu sisältää tiedot sovelluksen ryhmistä, kuten ryhmän nimen, ryhmän alkuperäisen luojaan käyttäjätunnuksen, luontiajankohdan, viimeisen muokkauksen ajankohdan ja osoitteen ryhmän logokuvaan.

Group-taulun tarkoitus on tallentaa tiedot sovelluksessa luoduista ryhmistä. Nämä ryhmät sisältävät useita käyttäjiä ja voivat myös olla työstämässä montaa projektia.

Ryhmän alkuperäinen luoja on erikoisasema, joka annetaan käyttäjälle uuden ryhmän luonnin yhteydessä. Sen tarkoituksena on antaa ryhmälle eräänlainen pääkäyttäjä, joka hallinnoi ryhmää ja pystyy ainoana henkilönä ryhmässä poistamaan itsensä ryhmästä.

Project-taulu

Project-tauluun on tallennettu tiedot sovelluksessa luoduista projekteista, kuten projektin nimi, projektin kuvaus, alkuperäinen luoja, projektin eräpäivä, valmiustaso, viimeisin muokkausajankohta ja projektin logokuvan osoite.

Projekteista tallennetaan myös tunniste jonka avulla käyttäjä voi kommentoida projektia sovelluksen Chat-ominaisuuden kautta. Projektien valmiustaso on osoitettu Boolean arvolla, jos projekti on valmis, arvo on tosi ja sovellusta käytettäessä kyseinen projekti esitetään lukittuna, jolloin sen tietokantatietoja ei voi enää muuttaa.

Assignment-taulu

Assignment-tauluun tallennetaan tiedot projekteihin lisättävistä työtehtävistä, niiden nimet, työtehtävien kuvaukset, valmiustaso sekä Boolean muodossa, että prosentuu-

aalisesti, luontipäivä ja viimeisin muokkausajankohta. Tauluun tallennetaan tietysti myös viiteavain siihen projektiin, johon työtehtävä kuuluu.

Käyttäjä voi kirjoittaa kommentteja sekä projekteista, että työtehtävistä, joiden sisältö, lähettäjä ja luontipäivä tallennetaan tietokantaan. Kommentteja voidaan myös editoida myöhemmin, jolloin niistä tallennetaan myös muokkausajankohta.

Assignment_component-taulu

Assignment_component-taulu sisältää tallennetut tiedot projektien työtehtävien eri osista, joita käyttäjä voi luoda työtehtävään halutun määrän ja asettaa työtehtävää suorittavia henkilöitä toteuttamaan nämä osat. Komponenteista tallennetaan niiden nimet, valmiustaso Boolean tyyppisesti sekä viimeisin muokkausajankohta. Näiden lisäksi komponenteista on viiteavaimet sekä projektiin että työtehtävään.

Message-taulu

Message-, eli viesti-tauluun tallennetaan kaikki sovelluksen Chat-ominaisuuden kautta lähetetyt viestit, niiden nimet, luontiajankohta sekä viiteavaimet sekä viestin lähittäneeseen käyttäjään, että ryhmään jonka kanssa käyttäjä keskusteli lähetyshetkellä. Nämä viestit tuodaan takaisin Chattiin käyttäjän kirjautuessa uudelleen Chat-palveluun.

Comment-taulu

Comment-tauluun tallennetaan sekä projekti-, että projektin työtehtävät-sivuilla tehdyt kommentit, kommentteista tallennetaan tekstisisältö, lähittäjän nimimerkki sekä lähetyksen aika. kommentit erotellaan sovelluksessa projektin ja työtehtävien kommentteihin riippuen siitä onko kommenttiin tallennettu työtehtävän id, eli tunniste. Mikäli tunniste on tallennettuna tietokannassa, on kyseessä työtehtävän kommentti, jos ei ole, kommentti kuuluu projektisivulle.

6.9.3 Tietokannan relaatiot

Person- ja Group-taulujen välille muodostettiin 'monen suhde moneen'-relaatio perusteltuna sillä, että sovelluksen toiminnan mukaisesti käyttäjän tulee pystyä liittymään useaan ryhmään ja ryhmä ei olisi ryhmä ilman useita jäseniä, Person- ja Group

Assignment-työtehtävällä on 'monen suhde yhteen'-relaatio Project-työtehtävien kanssa. Sillä samaa työtehtävää ei ole tarkoitus jakaa eri projektien välillä, koska niitä käytetään hahmottamaan projektin valmistumista, Assignment-työtehtävällä on samankaltainen relaatio Assignment_component-työtehtävien kanssa samasta syystä ja työtehtävän komponentteja käytetäänkin sovelluksessa laskemaan työtehtävän valmiustaso prosentuaalisesti.

Comment-työtehtävällä on 'monen suhde yhteen'-relaatio sekä Project-, että Assignment-työtehtävien kanssa. Sillä molemmat käyttävät kommentointi-ominaisuutta ja vaikka kommentteja voi olla paljon kummallakin sivulla, on jokainen yksittäinen kommentti sidottu tiettyyn käyttäjään, aikaan ja paikkaan.

Invite-työtehtävää käytetään tallentamaan ryhmän moderaattorin tai valvojan lähettämä kysely, jossa haluttua käyttäjää pyydetään liittymään lähettäjän ryhmään, kutsu ryhmään esitetään vastaanottajan home.aspx-sivun viestit osiossa, jossa käyttäjä voi joko hyväksyä tai hylätä kutsun.

6.9.4 SignalR:n toiminta tietokannassa

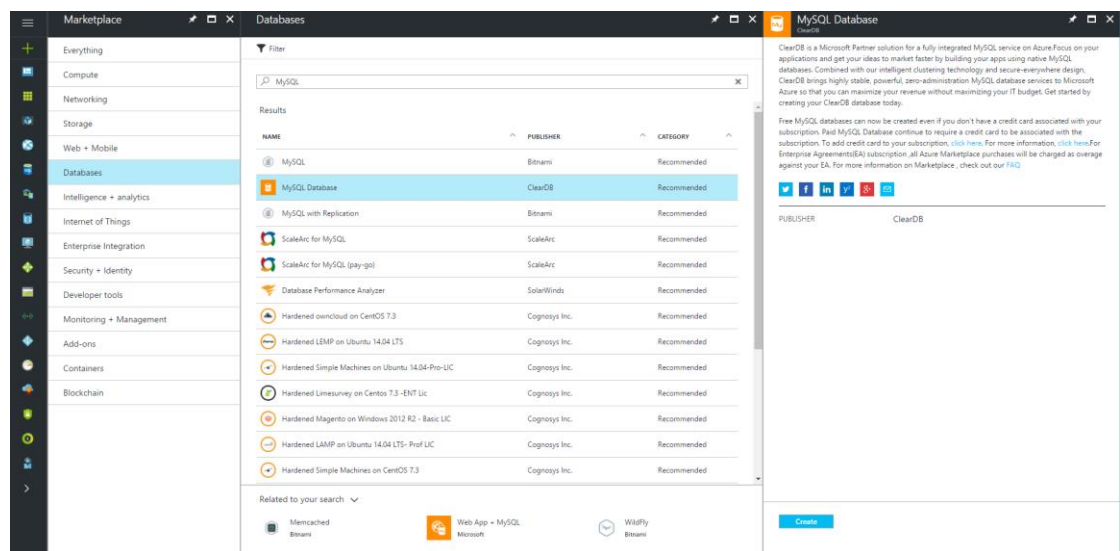
SignalR:n yhteyttä valvotaan tietokannassa Connection-työtehtävien avulla. Tähän työtehtävään tallennetaan tieto siitä, onko käyttäjä sillä hetkellä kirjautuneena sellaiselle verkkosivustolle, jossa SignalR on käytössä, kun käyttäjä rekisteröi käyttäjätunnuksensa, tallennetaan tietokantaan Connection-työtehtävään tieto uudesta käyttäjästä.

Kun käyttäjä avaa sivun, jolla SignalR on käytössä (Projekti-, projektin työtehtävä- tai ryhmä-sivut), tietokannassa muutetaan Connection-työtehtävien tietoja niin, että käyttäjän tila asetetaan yhdistyneeksi, yhteyden muodostamisajankohta otetaan talteen ja käyttäjän avaaman projektin, työtehtävän tai ryhmän tunnus tallennetaan viiteavaimeksi. Näiden tietojen avulla SignalR tietää millä sivulla käyttäjä kulkee ja mitä käyttäjä on sivulla työsteämässä.

Verkkosivun Chat-palvelua käytettäessä, kirjoittamalla projektin tunnusteen, joka on muodossa #PROXXX (Missä X:t ovat numeroita), ennen kommenttiviestin sisältöä viesti tallennetaan kommenttina tietokantaan. Tämän lisäksi SignalR lähettää käskyn päivittää projektin kommentit kaikille projektia sillä hetkellä työsteävälle.

6.10 Sovelluksen julkaiseminen verkossa

Tietokannan luonti Azure-palvelussa tapahtui hakemalla ensin oikeanlainen tietokantatyyppi. Tämä tehtiin valitsemalla Azuren kojelaudan pääsivun navigointipalkista vihreä plusmerkki ja sitten valitsemalla avautuvasta valikosta 'Databases'-kohdan. Tällä avautuvasta lisävalikosta valittiin vielä 'See all'-vaihtoehto sillä MySQL tietokanta ei ole valittavissa oletuksena. Seuraavaksi avautuneeseen hakuikkunaan kirjoitettiin etsityn tietokantatyyppin nimi, kuten kuviossa 19 on esitetty ja valittiin ClearDB-niminen MySQL Database. Lopuksi kyseisen tietokannan valinta varmistettiin painamalla 'Create'-painiketta.

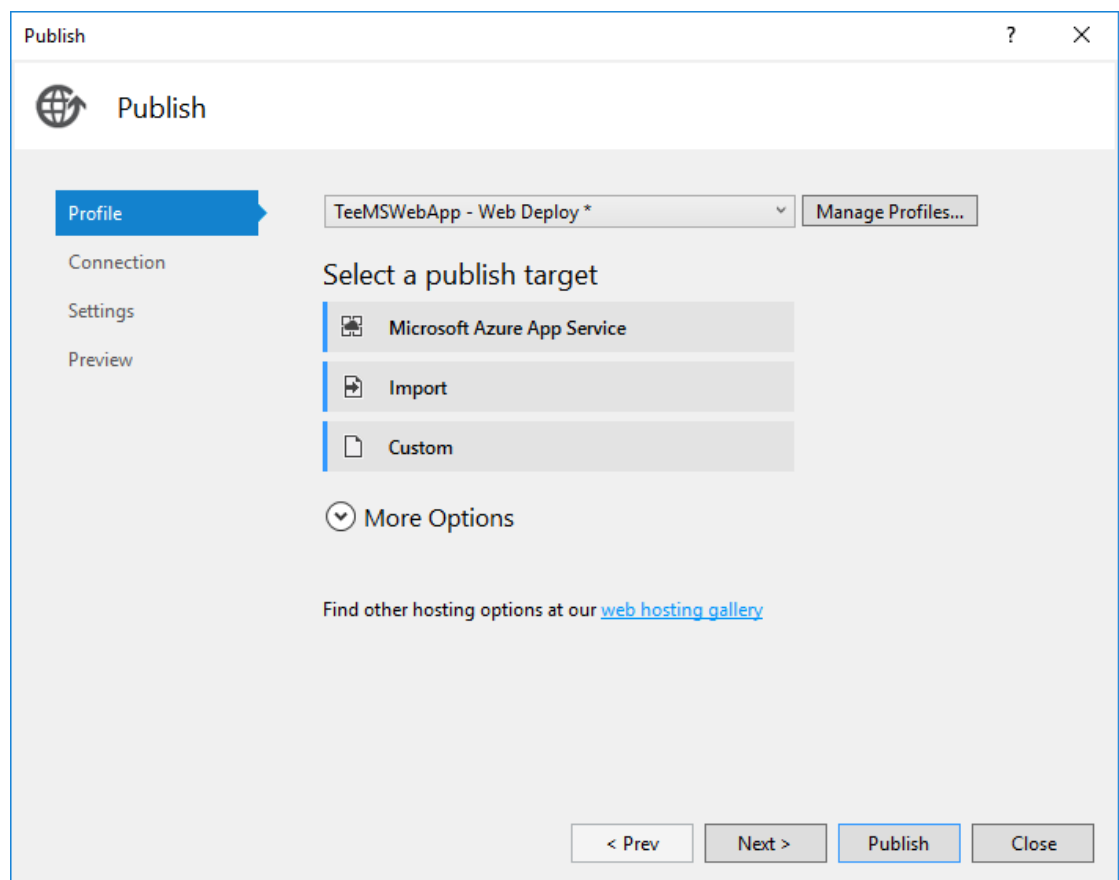


Kuvio 19. Kuva Azuren kojelaudasta MySQL-tietokannan luontihetkellä.

Seuraava askel tietokannan luonnissa oli tietokannan tietojen syöttäminen, Azure aukaisi ikkunan, jossa tietokannalle syötetään nimi, Azuren käyttäjätilin nykyinen tilaustyyppi, joka opinnäytetyön tapauksessa on ilmainen kokeiluversio, käyttääkö tietokanta jaettuja resursseja muiden käyttäjien kanssa Microsoftin palvelimilla, vai annetaanko sille yksilölliset, omistettut resurssit palvelimella, opinnäytetyön kohdalla valittiin jaetut resurssit. Lopuksi päätettiin vielä resurssiryhmä, palvelimen sijainti ja hinnoittelu, opinnäytetyön kohdalla palvelimen sijainniksi valittiin Pohjois-Eurooppa ja hinnoitteluksi Mercury-sopimus.

Verkkosovelluksen julkaiseminen Visual Studioon tapahtui Visual Studio Build-valikosta, jossa alavetovalikosta valittiin vaihtoehto "Publish Web App". Tämä avaa

kuvion 20 mukaisen ikkunan, jossa valittiin haluttu julkaisusijainti, opinnäytetyön tapauksessa valittiin ”Microsoft Azure App Service” ja syötettiin Azureen tehdyn käyttäjätilin tiedot, kun niitä valinnan jälkeen avautuneessa ikkunassa kysyttiin. Tämän jälkeen samassa ikkunassa luotiin uusi sovelluspalvelu, joka on kokoelma fyysisiä resursseja, joita käytetään sovelluksen isännöintiin. Koska opinnäytetyön toteutuksessa oltiin jo aikaisemmin luotu Azure-pilvipalveluun resurssiryhmä tietokannan luonnin yhteydessä, ei uutta resurssiryhmää tarvinnut luoda sovelluksen julkaisun yhteydessä.



Kuvio 20. Visual Studion verkkosovelluksen julkaisemiseen käytetty ikkuna.

Viimeinen askel sovelluspalvelun tekemisessä oli sovelluksen nimeäminen. Tätä nimeä käytetään oletuksena DNS nimeksi sovellukselle Azure-pilvipalvelussa, joten sen täytyy olla uniikki kaikkien Azuressa sijaitsevien sovellusten kesken, opinnäytetyön sovelluksen nimeksi annettiin ”TeeMsWebApp”. Tämän jälkeen Visual Studio komennettiin aloittamaan sovelluksen julkaisu Azure-pilvipalveluun.

6.11 Testaus

6.11.1 Yleistä

Ohjelmistojen testaus on olennainen osa ohjelmistokehitystä, ilman testausta ei kehittäjä voi puhua minkäänlaisella varmuudella ohjelmistojen toimivuudesta tai luotettavuudesta, testaamaton ohjelma voi olla käyttäjälle valtava riski, siksi ohjelmiston kattava testaus on aina suotavaa.

Testaamisen tavoite ei kuitenkaan ole ohjelmiston toiminnan osoittaminen, vaan virheiden paikantaminen ja mahdollisten ohjelmistokokonaisuuden pullonkaulojen löytäminen. Sillä siis pyritään ohjeistamaan kehittäjiä tuotteen nykyisestä laadusta ja ohjaamaan huomiota parannusta vaativiin seikkoihin.

Testaukseen käytettyjä menetelmiä on vuosien varrella kehitetty tuhansia, menetelmiä kokonaisten projektien testauksesta aina yksittäisten funktioiden testaukseen on olemassa monia eri käytäntöjä ja työkaluja, mutta selvää on, että ohjelmistoprojektien tapauksessa, testausta tulisi suorittaa jatkuvasti projektin alusta lähtien. Tällä estetään virheiden lipsahtaminen projektin pääkomponentteihin aikaisessa vaiheessa, joka voi säästää kehittäjiltä paljon päänvaivaa virheellisten komponenttien korvaamisessa siinä vaiheessa, kun projektiin on jo lisätty monia uusia komponentteja, jotka ovat riippuvaisia virheellisistä komponenteista.

6.11.2 Testauksen toteutus

Opinnäytetyön osalta testausta on toteutettu pääsääntöisesti kehittämisen yhteydessä tapahtuvalla manuaalitestauksella, eli ns. White-box testauksella. Sanalla White-box viitataan läpinäkyvään laatikkoon, jonka kaikki sisäiset toiminnot ovat nähtävissä. Tämä on verrattavissa ohjelmiston kehittäjään, joka testatessaan ohjelmistokoodin toimivuutta tai logiikkaa, ikään kuin katselee läpinäkyvän laatikon sisäistä toimintaa.

Sovelluksen ominaisuuksia testattaessa löydetyt bugit ja virheet on löydön jälkeen kirjattu tekstitiedostoon, yhdessä virheen kuvauksen ja tapahtumapaikan kanssa, kun bugeja ja virheitä on myöhemmin lähdetty korjaamaan, on niiden ratkominen selke-

ämpää, kun kaikki virheestä tiedetty, on helposti saatavilla, eikä ainoastaan muistin varassa.

Manuaalisen testauksen lisäksi, sovelluksen toimintaa Microsoft Azuren pilvessä on testattu Azuren tarjoamilla rasiustesteillä, joissa on voitu asettaa tietty määrä käyttäjiä rasittamaan sovellusta tietyksi aikaa, yksi tällainen testi on esitetty liitteenä. (Ks. liite 10)

6.12 Dokumentointi

6.12.1 Yleistä

Projektien kehitystyön dokumentoinnin tavoitteena on pysyä perillä projektin tavoitteista ja näiden tavoitteiden saavuttamisesta, monimutkaisten ohjelmistojen kehittäminen ymmärtämättä niiden tavoitetta tai tavoitteen katoaminen kehitystyön aikana johtaa tuloksiin, jotka eivät millään tasolla vastaa vaatimusmäärittelyssä asetettuja tavoitteita.

Dokumentoinnin todellinen tärkeys on kuitenkin vaikeasti ymmärrettävissä yksin työstetyistä projekteista, jos teet kaiken itse, jonkinlaisen kokonaiskuvan säilyttäminen on lähes välttämätöntä, mutta jos projektia on tekemässä kokonainen ryhmä ihmisiä, niin dokumentoinnilla tehdystä tiedonvälityksestä tulee projektin kannalta elintärkeää. Sillä omiin toimiinsa erikoistuneet tekijät eivät voi olla jatkuvasti keskenään yhteydessä ja nämä säröt kommunikoinnissa voivat johtaa ryhmän tekemään kahta eri projektia samanaikaisesti.

Yksin työstetyn projektin dokumentoinnissa on silti oma arvonsa, pitkään työstetyssä projektissa voi olla vaikeaa muistaa milloin jokin asia on lisätty tai muokattu, silloin on hyvä olla olemassa dokumentti, johon jokainen lisäys on kirjattu. Tämä helpottaa virheiden ja bugien korjauksessa ja ehkäisemisessä.

6.12.2 Dokumentoinnin toteutus

Opinnäytetyön dokumentointi on ensisijaisesti hoidettu tässä raportissa, mutta monia muitakin dokumentteja on kehitystyön aikana luotu. Näistä päällimmäisenä on jokaisen työpäivän päätteeksi muokattu tekstitiedosto, johon on kirjoitettu kaikki

sinä päivänä tehdyt asiat ja työtunnit. Tämän dokumentin avulla on voitu tarkistaa, milloin jotain ominaisuutta on viimeksi muokattu ja mitä muuta on sillä välin muutettu.

Kaikkien opinnäytetyössä käytettyjen työkalujen asennuksen aikana on niiden asennuksesta ja siihen käytetyistä tiedoista tehty muistiinpanoja, jotta sama tilanne saadaan vikatilanteissa toistettua. Opinnäytetyössä käytetystä tietokannasta on myös tehty tekstipohjainen SQL-luontiskripta, jolla tietokanta voidaan luoda uudestaan tyhjänä mahdollisissa vikatilanteissa. Lisäksi tietokannan eri versioista on tallennettu omat skeemansa, jotta vanhempaan versioon on voitu palata, jos uudessa versiossa on havaittu ongelmia. Samalla GitHub palvelua on käytetty ohjelmakoodin versionhallintaan.

7 Opinnäytetyön tulokset

7.1 Opinnäytetyön lopullinen tila

Opinnäytetyön kaltaiset ohjelmistoprojektit eivät periaatteessa ole koskaan valmiita, vaan niiden toimintoja voidaan lähes loputtomiin lisätä tai muokata paremmiksi. Tästä huolimatta opinnäytetyön vaatimusmäärittelyssä annetut tavoitteet on nykyisessä versiossa saavutettu suhteellisen hyvin, kaikki pääominaisuudet on saatu implementoitua ja sovelluksen käyttämä tietokanta ja palvelinympäristö on saatu asennettua ja toimimaan yhdessä sovelluksen kanssa.

Tietokannan koko kasvoi huomattavasti suunnitellusta. Tämä johtui projektin kehityksen aikana kasvaneista tavoitteista ja sovellukseen lisätyistä ominaisuuksista, muutamia eroavaisuuksia ja kehityksen aikana muuttuneita ajatuksia lukuun ottamatta, opinnäytetyö vastaa laajalti alussa tehtyjä suunnitelmia ja tietyiltä osin, on jopa niitä parempi. Sovelluksen ulkonäköön ja esitystapaan olisi pitänyt käyttää enemmän aikaa.

Olisi varmasti ollut myös hyödyllistä käyttää enemmän aikaa eri ohjelmistokehyksiin tutustumiseen, osan sovelluksen toiminnasta olisi varmaan voinut korvata jo olemassa olevalla ratkaisulla, nyt valtava osa sovelluksen toiminnallisuutta on itse tehtyä ”laatua”.

Lopullisena tuloksena on kuitenkin toimiva ohjelmisto, jossa on mukavan laaja kirjo ominaisuuksia. Lisäksi sen tekoon on käytetty ohjelmistoja, kirjastoja ja tekniikoita, joita varmasti käytetään myös työelämässä.

7.2 Havaitut ongelmat

Itselleni huomattavin ongelma opinnäytetyötä tehdessä kumpuaa virheoletuksesta, että SignalR:n käyttäminen olisi todella aikaa vievää puuhaa. Minulla ei ollut minikäänlaista aikaisempaa kokemusta reaaliaikaisten verkkosovellusten tekemisestä ja aluksi ajattelin, että rajaan sen käytön ainoastaan sovelluksen Chat-ominaisuuteen, jotta se olisi eristetty muusta ohjelmistosta ja näin helpommin hallittavissa.

Ohjelmisto siis rakennettiin alun perin toimimaan pitkälti ilman SignalR:n osallisuutta projekteihin tai ryhmiin ja näillä sivuilla olevien ominaisuuksien päivitys tehtiin ASP:n UpdatePaneleiden kautta, kuitenkin, kun SignalR:n kanssa alkoi työskennellä. Se osoittautuikin varsin helppokäyttöiseksi ja päätin lisätä sen toiminnallisuutta myös projekteihin ja ryhmiin.

vaikka onnistuinkin saamaan reaaliaikaista toiminnallisuutta molempiin, ei aika enää riittänyt viemään ajatusta loppuun asti. Sen vuoksi osa reaaliaikaan perustuvasta toiminnallisuudesta tuottaa pieniä ongelmia käyttäjäystävällisyyteen. Nämä eivät kuitenkaan ole kriittisiä virheitä ja täysin korjattavissa investoimalla sovelluksen kehittämiseen lisää aikaa.

Lisäksi sovelluksen monimutkaisuus altistaa sen tietyiltä osin bugeille ja monet näistä on varmasti jäänyt huomaamatta testauksesta huolimatta, eli myös testaukseen olisin voinut käyttää enemmän aikaa.

Lopuksi vielä mainittakoon, että sovellukseen ei ole tehty käyttöohjeita, mikä olisi toivottavaa, mutta se ei ollut prioriteetti sovellusta tehdessä.

8 Johtopäätökset ja jatkokehitys

Opinnäytetyö oli itselleni valtavan opettavainen. Olen tyytyväinen päätökseeni ottaa riskin ja sisällyttää opinnäytetyöhön itselleni täysin tuntemattoman

reaaliaikasovellusten tekemiseen käytettävän teknologian. Sen kanssa työskentely oli kaikkein palkitsevinta vaikka ohessa pari ongelmaakin syntyi.

Muita vähän käsiteltyjä aiheita olivat Entity Framework ja Microsoft Azure. Entity Framework oli sen verran tuttu, että sen käyttö oli opetettu opinnäytetyön tekemistä edeltävänä syksynä, mutta siitä huolimatta opinnäytetyötä tehdessä. Sen kanssa työskentelystä oppi koko ajan jotain uutta. Azure oli taas toinen ohjelmisto jota en ollut käyttänyt aiemmin. Sen opettelu ja käyttö oli kuitenkin helppoa ja sujui nopeasti.

Olen omalta osaltani tyytyväinen opinnäytetyössä saavutettuihin tavoitteisiin ja osin jopa yllättynyt omiin saavutuksiini. Monessa asiassa on kuitenkin vielä parantamista ja varsinkin suunnittelu ja testaus tulee olemaan erityishuomiossani tulevaisuudessa.

Sovelluksen jatkokehityksessä aloittaisin kehittämällä sovellusta käyttäjäystävällisempään suuntaan ja paikkaamalla havaitut ongelmat. Siitä eteenpäin sovellukseen voidaan lisätä enemmän projekteista ja niiden työskentelystä kertovia taulukoita ja muita toimintoja, jotka antavat käyttäjille paremman kokonaiskuvan heidän projektiansa edistymisestä.

Lähteet

ADO.NET Overview. 2017. Microsoft Developer Networkin sivuilla oleva yleiskatsaus ADO.NET-teknologiaan. Viitattu 28.4.2017. [https://msdn.microsoft.com/en-us/library/h43ks021\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/h43ks021(v=vs.110).aspx)

Anderson, R. Patel, S. Pasic, A. Dykstra, T. Learn ASP.NET? 2010. Määritelmä ASP.NETin kotisivuilta. Viitattu 30.1.2017. <https://www.asp.net/aspnet>

CSS Introduction. N.d. W3Schoolsin CSS oppaan esittelysivu. Viitattu 28.4.2017. https://www.w3schools.com/css/css_intro.asp

Entity Framework. 2016. Microsoft Developer Networkin sivuilla oleva Entity Framework dokumentaatio. Viitattu 8.5.2017. [https://msdn.microsoft.com/en-us/library/ee712907\(v=vs.113\).aspx](https://msdn.microsoft.com/en-us/library/ee712907(v=vs.113).aspx)

Fielding, R. & Reschke, J. Hypertext Transfer Protocol (HTTP/1.1). 2014. IETF:n (Internet engineering Task Force) sivuilla oleva luonnos HTTP:n viestittämisestä ja reitittämisestä. Viitattu 5.2.2017. <https://trac.ietf.org/trac/httpbis/browser/draft-ietf-httpbis/26/draft-ietf-httpbis-p1-messaging-26.txt>

Finley, K. What exactly is Github anyway? 2012. Techcrunch sivulla oleva artikkeli Githubsta. Viitattu 10.2.2017. <https://techcrunch.com/2012/07/14/what-exactly-is-github-anyway/>

Fletcher, P. Patel, S. Pasic, A. Dykstra, T. Introduction to SignalR. 2014. Microsoftin sivuilla oleva esittely SignalR-teknologiasta. Viitattu 30.4.2017. <https://docs.microsoft.com/en-us/aspnet/signalr/overview/getting-started/introduction-to-signalr>

Garsiel, T. & Irish, P. How Browsers Work. 2011. Artikkelin verkkoselainten toiminnasta. Viitattu 30.1.2017. <https://www.html5rocks.com/en/tutorials/internals/howbrowserswork/>

Getting started with JIRA Software. 2017. JIRAn kehittäjän Atlassianin sivuilla oleva dokumentaatio JIRAn käytöstä. Viitattu 8.5.2017. <https://confluence.atlassian.com/jirasoftwareserver073/getting-started-with-jira-software-861254171.html>

Getting started with Trello. 2017. Trello sivuilla oleva aloittelijan opas Trello käyttöön. Viitattu 8.5.2017. <https://trello.com/guide>

HTML URL Encode. N.d. W3Schoolsin HTML-oppaan URL osuus. Viitattu 3.2.2017. http://www.w3schools.com/html/html_urlencode.asp

Visual Studio IDE. 2015. Microsoft Developer Networkin sivuilla oleva esittely Visual Studio IDEstä. Viitattu 8.5.2017. <https://msdn.microsoft.com/en-us/library/dn762121.aspx>

Macy, M. Gries, W. Myers, T. Archer, Tom. & Shahan, R. Introduction to Microsoft Azure Storage. 2017. Microsoft Azuren sivuilla oleva esittely Azuren tallennustilasta. Viitattu 30.4.2017. <https://docs.microsoft.com/en-us/azure/storage/storage-introduction>

Mayo, J. Introduction to LINQ. 2016. Csharp.com sivulla oleva esittely LINQ:sta osana oppisarjaa. Viitattu 17.3.2017. <http://csharp-station.com/Tutorial/Linq/Lesson01>

Microsoft Ajax overview. 2017. Microsoft Developer Networkin sivuilla oleva katsaus Ajaxiin. Viitattu 5.2.2017. <https://msdn.microsoft.com/en-us/library/bb398874.aspx>

MySQL Installer. 2017. MySQL tietokannan sivu, jossa kerrotaan, mikä MySQL Installer on ja tarjotaan latauslinkki. Viitattu 10.2.2017. <https://dev.mysql.com/downloads/installer/>

MySQL Workbench. 2017. MySQL.com sivuilta löytyvä selitys MySQL Workbench-ohjelmasta. Viitattu 9.5.2017. <https://www.mysql.com/products/workbench/>

Nations, D. What is freemium? 2017. lifewire.com sivulla oleva artikkeli, jossa kerrotaan mitä freemium tarkoittaa ja miten se vaikuttaa pelialalla. Viitattu 8.5.2017. <https://www.lifewire.com/what-is-freemium-1994347>

VPN. 2017. Computerhope sivulla oleva määritelmä VPN:stä. Viitattu 10.2.2017. <http://www.computerhope.com/jargon/v/vpn.htm>

Wagner, B. Wenzel, M. What is C#? 2016. Määritelmä Microsoft Developer Network verkkokirjastosta. Viitattu 30.1.2017. <https://msdn.microsoft.com/en-us/library/kx37x362.aspx>

What are the differences between JIRA Software, JIRA Service Desk and JIRA Core. 2016. Akeles.com sivulla oleva selitys JIRAn pakettien eroista. Viitattu 8.5.2017. <https://www.akeles.com/what-are-the-differences-between-jira-software-jira-service-desk-and-jira-core/>

What is CEITON? 2017. Ceiton.com sivulla oleva blogi, jossa selitetään, mikä CEITON on. Viitattu 8.5.2017. <http://ceiton.com/blog/en/what-is-ceiton/>

What is cloud computing? 2017. aloittelijan ohjeistus pilvipalveluista Microsoft Azuren sivulla. Viitattu 28.4.2017. <https://azure.microsoft.com/en-us/overview/what-is-cloud-computing/>

What is Git? N.d. Määritelmä Gitin korisivuilta. Viitattu 30.1.2017. <https://git-scm.com/>

What is HTML?. N.d. W3Schools.n HTML-oppaan HTML esittely. Viitattu 3.2.2017. http://www.w3schools.com/html/html_intro.asp

What is MySQL? 2017. Määritelmä MySQL:n internetsivujen manuaalista. Viitattu 30.1.2017. <https://dev.mysql.com/doc/refman/5.7/en/what-is-mysql.html>

What is SQL?. N.d. W3Schoolsin SQL-esittely. Viitattu 30.1.2017. http://www.w3schools.com/sql/sql_intro.asp

Liitteet


Liite 1. Käyttäjän pääsivu sovelluksessa, Home.aspx

The screenshot shows the user's home page in the TeeMs application. The page has a light blue header with the 'TeeMs' logo. On the left, there is a navigation menu with links for Home, Profile, Projects, Groups, Chat, and Sign out. The main content area is divided into two columns. The left column is titled 'Your Projects' and contains a project card for 'My First Project' with a blue gear icon. Below this is a 'No Image' placeholder with a 'Build a program' link and a 'Create new Project' button. The right column is titled 'Messages' and contains an empty message box. Below the 'Your Projects' section is a 'Your Groups' section with two group cards: 'TietstiMuokkaajat' with a group of people icon and 'Projektit' with a group of people icon. A 'Create new Group' button is located at the bottom of the 'Your Groups' section. At the very bottom of the page, it says 'Created By: Janne Kuukkanen.'

TeeMs

- Home
- Profile
- Projects
- Groups
- Chat
- Sign out


Your Projects


 My First Project

No Image [Build a program](#)

[Create new Project](#)

Your Groups

 TietstiMuokkaajat

 Projektit

[Create new Group](#)

Created By: Janne Kuukkanen.

Liite 2. Käyttäjän pääsivulla sekä projektit ja ryhmät listaavilla sivuilla käytetty ohjelmistokoodi, projektien ja ryhmien näyttämiseksi

```
protected void FillDivs()
{
    // Fill your projects section with the users projects and assignments
    // and fill your groups with the users groups

    UserContentManager contentmanager =
    new UserContentManager(User.Identity.Name);

    try
    {
        divYourGroups.Controls.Clear();
        divYourProjects.Controls.Clear();

        List<group> usergroups = contentmanager.GetUserGroups();
        List<project> userprojects = contentmanager.GetUserProjects();

        // First fill the projects section of the page
        foreach (var project in userprojects)
        {

            HtmlGenericControl projectdiv = new HtmlGenericControl("div");
            HtmlGenericControl projectinnerdiv = new HtmlGenericControl("div");
            HtmlGenericControl projectlink = new HtmlGenericControl("a");
            HtmlGenericControl projectimage = new HtmlGenericControl("img");
            HtmlGenericControl projectcut = new HtmlGenericControl("hr");

            projectdiv.Attributes.Add("class", "w3-container");
            projectdiv.ID = "divProject" + project.project_id;

            projectinnerdiv.Attributes.Add("class", "w3-container");

            projectimage.Attributes.Add("alt", "Project image");
            projectimage.Attributes.Add("height", "150px");
            projectimage.Attributes.Add("width", "150px");
            projectimage.Attributes.Add("style", "float:left;");

            // if the group has an existing picture in the database, we'll fetch it
            if (project.picture_url != null)
            {
                if (project.picture_url != String.Empty)
                {
                    projectimage.Attributes.Add("src", project.picture_url);
                }
            }
            else
            {
```

```

        projectimage.Attributes.Add("src", Request.ApplicationPath + "Images/no_image.png");
    }

    projectinnerdiv.Controls.Add(projectimage);

    projectlink.Attributes.Add("href", String.Format("Project.aspx?Project={0}",
    project.project_id));
    projectlink.Attributes.Add("style", "float:left; margin-top:5%; margin-left:5%;");
    projectlink.InnerText = project.name;

    projectinnerdiv.Controls.Add(projectlink);

    projectcut.Attributes.Add("style", "color:#000;background-color:#000; height:5px;");

    projectdiv.Controls.Add(projectinnerdiv);
    projectdiv.Controls.Add(projectcut);

    divYourProjects.Controls.Add(projectdiv);
}

// Next we'll fill the groups section of the page
foreach (var group in usergroups)
{
    HtmlGenericControl groupdiv = new HtmlGenericControl("div");
    HtmlGenericControl groupinnerdiv = new HtmlGenericControl("div");
    HtmlGenericControl grouplink = new HtmlGenericControl("a");
    HtmlGenericControl groupimage = new HtmlGenericControl("img");
    HtmlGenericControl groupcut = new HtmlGenericControl("hr");

    groupdiv.Attributes.Add("class", "w3-container");
    groupdiv.ID = "divGroup" + group.group_id;

    groupinnerdiv.Attributes.Add("class", "w3-container");

    groupimage.Attributes.Add("alt", "Group image");
    groupimage.Attributes.Add("height", "150px");
    groupimage.Attributes.Add("width", "150px");
    groupimage.Attributes.Add("style", "float:left;");

    // if the group has an existing picture in the database, we'll fetch it
    if (group.group_picture_url != null)
    {
        if (group.group_picture_url != String.Empty)
        {
            groupimage.Attributes.Add("src", group.group_picture_url);
        }
    }
}

```

```
    }
    else
    {
        groupimage.Attributes.Add("src", Request.ApplicationPath + "Images/no_image.png");
    }

    groupinnerdiv.Controls.Add(groupimage);

    grouplink.Attributes.Add("href", String.Format("Group.aspx?Group={0}",
    group.group_id));
    grouplink.Attributes.Add("style", "float:left; margin-top:5%; margin-left:5%;");
    grouplink.InnerText = group.name;

    groupinnerdiv.Controls.Add(grouplink);

    groupcut.Attributes.Add("style", "color:#000;background-color:#000; height:5px;");

    groupdiv.Controls.Add(groupinnerdiv);
    groupdiv.Controls.Add(groupcut);

    divYourGroups.Controls.Add(groupdiv);
}
}
catch (Exception ex)
{
    lbMessages.Text = ex.Message;
}
}
```


Liite 3. Ryhmäsivu sovelluksessa, Group.aspx

TeeMs

- [Home](#)
- [Profile](#)
- [Projects](#)
- [Groups](#)
- [Chat](#)
- [Sign out](#)

TekstiMuokkaajat

Group tag: #PRO105



Change picture

Group Members

⌘ Admin
⊙ CommonUser

Add members
Assign new role
Remove Member
Show info

Group projects

My First Project ▾

Project info

| Project description | Project assignments | | | | | | | | | | | | |
|--|---|-----------------|--------------|---|--|----------------|--------------|---|--|----------------|--------------|---|--|
| <p style="font-size: x-small; margin: 0;">The goal of this project is to build a house</p> | <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px 5px; font-size: x-small;">Build the floor</td> <td style="padding: 2px 5px; font-size: x-small;">50% Complete</td> </tr> <tr> <td style="padding: 2px 5px; font-size: x-small;">Assignments: 4, Finished assignments: 2</td> <td style="width: 100px;"></td> </tr> <tr> <td style="padding: 2px 5px; font-size: x-small;">Build the roof</td> <td style="padding: 2px 5px; font-size: x-small;">80% Complete</td> </tr> <tr> <td style="padding: 2px 5px; font-size: x-small;">Assignments: 5, Finished assignments: 4</td> <td style="width: 100px;"></td> </tr> <tr> <td style="padding: 2px 5px; font-size: x-small;">Build a window</td> <td style="padding: 2px 5px; font-size: x-small;">25% Complete</td> </tr> <tr> <td style="padding: 2px 5px; font-size: x-small;">Assignments: 4, Finished assignments: 1</td> <td style="width: 100px;"></td> </tr> </table> | Build the floor | 50% Complete | Assignments: 4, Finished assignments: 2 | | Build the roof | 80% Complete | Assignments: 5, Finished assignments: 4 | | Build a window | 25% Complete | Assignments: 4, Finished assignments: 1 | |
| Build the floor | 50% Complete | | | | | | | | | | | | |
| Assignments: 4, Finished assignments: 2 | | | | | | | | | | | | | |
| Build the roof | 80% Complete | | | | | | | | | | | | |
| Assignments: 5, Finished assignments: 4 | | | | | | | | | | | | | |
| Build a window | 25% Complete | | | | | | | | | | | | |
| Assignments: 4, Finished assignments: 1 | | | | | | | | | | | | | |

Show project

Created By: Janne Kuukkanen.

Liite 4. Ryhmien listaukseen käytetty sivu sovelluksessa. Tämä listaus poikkeaa käyttäjän pääsivun listauksesta näyttämällä ryhmien jäsenet, Groups.aspx

TeeMs

- Home
- Profile
- Projects
- Groups
- Chat
- Sign out

Your Groups



TekstiMuokkaajat

Group Members



Projektit

Group Members

Created By: Janne Kuukkanen.

Liite 5. Käyttäjän Chat-sivu sovelluksessa, Chat.aspx

The screenshot shows the chat interface of the TeeMs application. At the top, there is a light blue header with the 'TeeMs' logo in red and black. On the left side, there is a vertical navigation menu with the following items: Home, Profile, Projects, Groups, Chat, and Sign out. The main chat area is divided into three columns. The left column contains two buttons: 'TekstiMuokkaajat' (Text Editors) and 'Projektit' (Projects). The middle column displays a chat log with three messages: 'Admin: Hello Everyone! :)', 'CommonUser: Hello, how is the project coming along?', and 'Admin: It's progressing well, we should be ready for the testing phase in a couple of weeks.' The right column shows a list of participants: 'Admin' and 'CommonUser'. At the bottom of the chat area, there is a text input field and a 'Send' button.

Home
Profile
Projects
Groups
Chat
Sign out

TeeMs

TekstiMuokkaajat
Projektit

- Admin: Hello Everyone! :)
- CommonUser: Hello, how is the project coming along?
- Admin: It's progressing well, we should be ready for the testing phase in a couple of weeks.

- Admin
- CommonUser

Send

Created By: Janne Kuukkanen.


Liite 6. Projektisivu sovelluksessa, Project.aspx

TeeMs

- Home
- Profile
- Projects
- Groups
- Chat
- Sign out

My First Project

Project tag: #PRO100



Change picture

Groups working on the project

Choose Group ▾

Add a Group
Remove Group
Show info

Assignments

Choose Assignment ▾

Create new
Show info

Archive Project

Project description

The goal of this project is to build a house

Edit description

Workflow

Due date

< |> |<< |>> |<<< |>>> |<<<< |>>>>

miittu ke to pe la su

27 28 29 30 31 1 2

3 4 5 6 7 8 9

10 11 12 13 14 15 16

17 18 19 20 21 22 23

24 25 26 27 28 29

1 2 3 4 5 6 7

Project progress

Build the floor 50% complete

Assignments: 4. Finished assignments: 2

Build the roof 80% Complete

Assignments: 5. Finished assignments: 4

Build a window 25%

Assignments: 4. Finished assignments: 1

Comments

Send Comment

Created By: Janne Kuukkanen.

Liite 7. Projektien listaukseen käytetty sivu sovelluksessa. Tämä listaus poikkeaa käyttäjän pääsivun listauksesta näyttämällä projektien työtehtävät, Projects.aspx

The screenshot displays the TeeMs application interface. At the top, there is a light blue header with the 'TeeMs' logo in red and black. Below the header is a navigation menu on the left with links for Home, Profile, Projects, Groups, Chat, and Sign out. The main content area is divided into sections. The first section is titled 'Your Projects' and contains a blue snowflake icon and a link to 'My First Project'. Below this is an 'Assignments' section with a table listing tasks: 'Build the floor', 'Build the roof', and 'Build a window'. A horizontal line separates this from the next section, which features a large 'No Image' placeholder and a link to 'Build a program'. Another 'Assignments' section is visible below. At the bottom, there is a 'Create new Project' button and a footer note: 'Created By: Janne Kuukkanen.'

TeeMs

Home
Profile
Projects
Groups
Chat
Sign out

Your Projects

 [My First Project](#)

Assignments

| |
|-----------------|
| Build the floor |
| Build the roof |
| Build a window |

No Image [Build a program](#)

Assignments

[Create new Project](#)

Created By: Janne Kuukkanen.

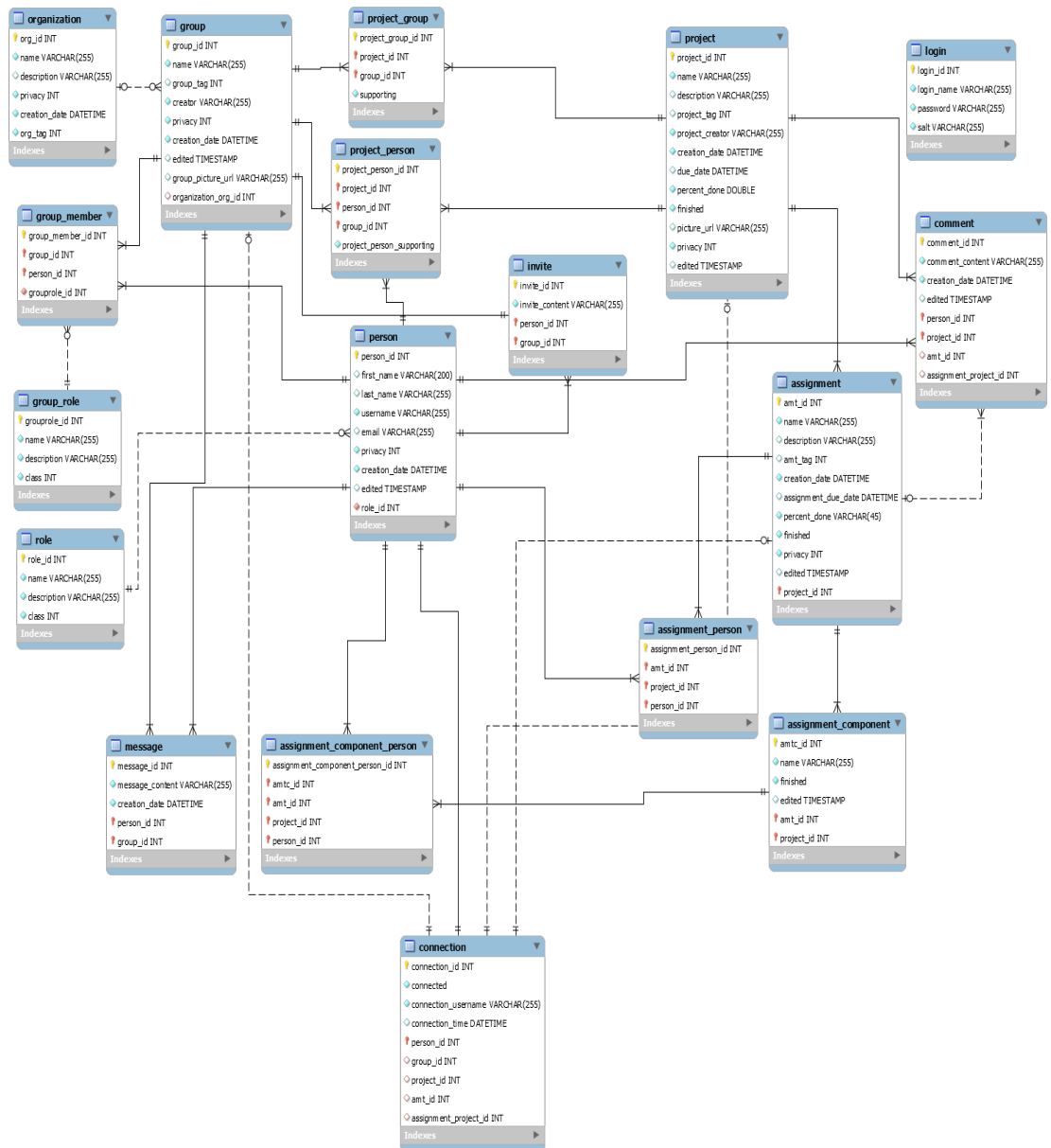
Liite 8. Projektin työtehtäväsivu sovelluksessa, Assignment.aspx

The screenshot displays the 'Teems' web application interface for an assignment page titled 'Build the floor'. The page is divided into several sections:

- Header:** The 'Teems' logo is prominently displayed in the top left corner.
- Navigation:** A vertical menu on the left side includes links for Home, Profile, Projects, Groups, Chat, and Sign out. A 'Back to project page' button is located at the top of the main content area.
- Assignment Details:**
 - Title:** 'Build the floor'
 - Description:** 'The goal of this assignment is to build a floor for our house.' This text is contained within a text area with an 'Edit description' button below it.
 - Members:** A section titled 'Assignment members' shows a list of roles: 'Admin' and 'CommonUser'. Below this list are buttons for 'Add Member' and 'Remove Member', and a red 'Delete Assignment' button.
- Workflow:**
 - Assignment Components:** A horizontal flow of four components is shown, each with a color-coded background and a list of roles:
 - First Component (Green):** Admin
 - Second Component (Green):** Admin, CommonUser
 - Third Component (Blue):** Admin, CommonUser
 - Fourth Component (Blue):** CommonUser
 - Action:** A 'Create new component' button is positioned below the workflow diagram.
- Comments:** A section for user feedback featuring a text input field and a 'Send Comment' button.

Created By: Janne Kuukkanen.

Liite 9. Tietokannan kokonaisdiagrammi



Liite 10. Microsoft Azure-pilvipalvelussa ajettun rasitustestin tulokset

