

EKSAMEN i IS-APP2000V
HØST/VÅR 2017-2018

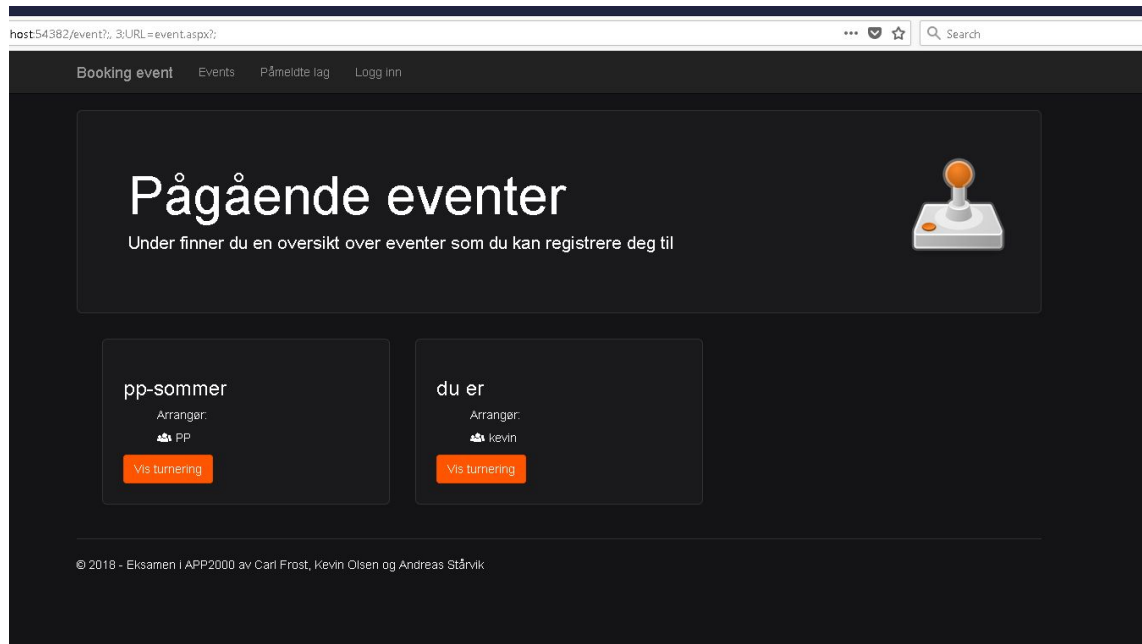


Av Kevin olsen, Carl Frost og Andreas Stårvik

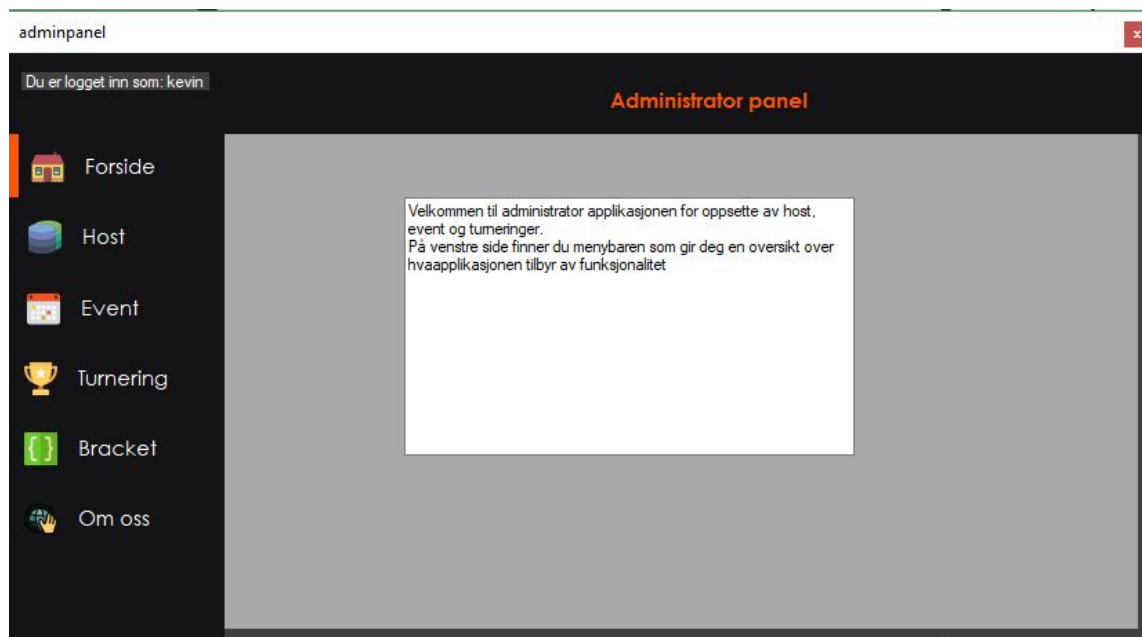
(Se brukermanual før bruk)

Introduksjon

Forside: asp.net



Forside: Windows Forms



C-Team er en gruppe bestående av Kevin Olsen, Andreas Stårvik, og Carl Christian Frost. Gruppen startet første semester med syv medlemmer etter en sammenslåing hvor to medlemmer falt fra tidlig. Etter første studieår var fullført skilte gruppen lag med enda to medlemmer og sto igjen med det som idag er C-Team.

Vi har som prosjekt valgt å utvikle en applikasjon som håndterer booking og administrering av e-sportsarrangementer. Vi har valgt dette prosjektet da gruppens medlemmer har god erfaring med e-sport og er interessert i emnet på fritiden.

I starten av prosjektet satt vi opp ønsker og kriterier for innhold og funksjonalitet som ville være den grunnleggende delen av applikasjonen. Ettersom vi har jobbet mer med prosjektet kom det opp en rekke ønsker og forslag som kunne være med på å utbedre applikasjonen for å etterhver gjøre den kommersielt konkurransedyktig.

Gjennom det året vi har arbeidet på prosjektet har vi gjort oss en rekke lærdommer. Vi har oppdaget nye måter å jobbe på med tanke på prioritering av tid. Vi har oppdaget den praktiske betydningen av å jobbe i scrum. Vi har sett hvordan et en visjon byr på utfordringer når den skal realiseres, selv etter god planlegging, vi har sett at små detaljer kan by på utfordringer som krever mer tid enn forutsatt, og vi har erfart forskjellen på realistiske og ønskelige mål.

Prosjektets målsetning

Målsettingen med dette prosjektet er å utvikle en applikasjon som behandler turneringsoppsett for e-sport arrangementer.

- Windows Forms
 - Det skal være forvalg for oppsett av de fleste spill som spilles i e-sport med tanke på lagkonstruksjon, poenggivning, rundeantall og andre relevante opsjoner for oppsett.
 - Det skal være mulig for spillere å melde seg på med sine respektive lag.
 - Administratoren skal kunne administrere vertskap, eventer, turneringer og fremgang.
- ASP.NET:
 - En portal for brukere/spillere og besøkende.
 - Informasjon om arrangementer kan leses.
 - Man skal kunne se poeng og fremdrift for de ulike lagene i turneringer.
 - Man skal kunne se detaljert informasjon om lag og spillere.
 - Det skal være mulighet for å kjøpe billetter til arrangementer.
 - Det skal være mulighet for oppsett av eksterne lenker til arrangementer og vertskap og sponsorer. (eventuelt nettbutikk osv.)

Ut ifra disse punktene har vi valgt å sette sammen et prosjekt som er mulig å bruke, men mangler store deler av målsettingen. Det vil si at vår målsetting er for et fullverdig produkt som kan brukes på det kommersielle markedet, mens det som blir levert inn som eksamensprosjekt er en grunnleggende løsning med stort utbedringspotensial som vi ønsker å jobbe videre med som bacheloroppgave. Så for å si det kort og godt, de punkter fra målsettingen som ikke er oppfylt til innlevering av eksamen er punkter for videreutvikling.

Kort om løsningen

Brukerne av systemet får tilgang gjennom en nettside utviklet med ASP.NET.

- Denne vil bestå av en forside som viser eventer, en oversikt over registrerte lag med spillere og selve oppsettet (bracketten) for turneringen.
- Mulighet for lagleder å registrere lag og spillere, og mulighet for lagleder å administrere laget og spillerne.
- Det skal brukes en innloggingsfunksjon for registrering og endring av lag.
- Resten av ASP.NET siden vil være åpen for allmenheten.

Administratoren skal gjennom windows forms:

- Opprette/endre/slette vertskap.
- Opprette/endre/slette event.
- Opprette/endre/slette turnering.
- Kunne manuelt dra lag videre i turneringsoppsettet (drag and drop).

Resultater – hva løsningen leverer

Løsningen leverer i dag:

- Et grunnleggende system for oppretting av vertskap, event, turneringer, lag og spillere.
- Det opprettes en bruker, og deretter kan man opprette et lag. Denne brukeren registrerer da spillere til dette laget.
- Det opprettes admin-brukere for å håndtere adminpanelet, hvor man setter opp og administrerer vert, events og turnering, og fordeler lag i bracketen.
- Dra vinnende lag videre i turneringer, i bracketen, ved hjelp av “drag & drop”

Brukermanual

ASP.NET:

- I navigasjonslinjen i toppen kan du gå til hovedsiden (event), eller se påmeldte lag som viser alle lag sortert etter turnering_id.
- Du starter på forsiden der brukeren ser arrangementer.
- Manøvrer så videre inn på valgt arrangement for å finne turneringer.

- Derfra kan brukeren velge å se bracketen, altså påmeldte lag og turneringsforløpet, eller å registrere lag til valgt turnering.
- Trykker du på registrer lag så vil du bli sendt til innlogging. Der kan du registrere bruker for så og logge inn.
- Etter du har logget inn kan du registrere lag, laget vil nå vises i bracketten til valgt turnering.
- Når laget er registrert tar appen deg til mine lag for å se lagene dine.
- Inne på oversikten over mine lag kan du enten administrere laget (hvor du også kan slette laget), eller se spillere registrert til laget.
- Appen sier ifra om det ikke er noen registrerte spillere. Du kan uansett registrere nye spillere når du har trykket inn på vis spillere.
- Etter registrering av spillere kan du trykke på en knapp som går ut til oversikten over spillerne. Her kan du nå administrere spillerne (hvor du kan slette spilleren).

Windows Forms:

Host siden(tredje side):

- Legg til ny host ved å skrive inn navn og trykke på knappen
- Endre host, velg host i listen, skriv inn nytt navn og trykk deretter på knappen for å oppdatere navnet.
- Sletter host ved å velge i listen som i punktet over, trykker så på slett knappen.
- For å velge bort den hosten du har valg, for f.eks. å velge en annen, trykk på fjern host knappen.

Event siden(andre side) :

- For å få opp hosten som brukeren laget, er man nødt til å avslutte og starte programmet på nytt for å se den nylig registrerte hosten.
- Legge til nytt event ved å først velge host og trykke hent ut. Skriv deretter inn nytt navn og trykk på legg til knappen.
- For å endre event, velg eventet i listen. Skriv inn nytt navn og trykker deretter på endre knappen.
- For å slette vent, velg eventet i listen og trykk på slette knappen
- For å velge bort det eventet brukeren har valgt, for f.eks. å velge en annen, trykk på clear knappen.

I det man gjør en en endring i datagridview forsvinner informasjonen og man er nødt til å velge host på nytt og trykker på hente ut knappen for å få frem den oppdaterte informasjonen i datagridview.

Turnering side(fjerde side):

- Først velge hvilken host og trykk deretter på velg host knappen. Du får da ut alle eventer knyttet til hosten i listen ved siden av.
- Så velger du hvilket event og trykk deretter på velg event knappen. Du får da ut registrerte turneringer i eventet.
- Legg til ny turnering ved å skrive inn navnet og trykk på legg til knappen.
- Endre turnering ved å velge turnering, skriv inn nytt navn og trykker deretter på endre knappen
- For å slette turneringer velger du turnering og trykker på slette knappen
- For å velge bort den turneringen du har valgt, for f.eks. å velge en annen, trykk på clear knappen.

bracket (femte side)

- Velg turnering fra listen og trykk hent ut lag for å hente ut alle lagene som er registrert til turneringen. De registrerte lagene vil dukke opp i de åtte boksene helt til venstre.
- Brukeren kan dra lagene som skal spille mot hverandre videre til de neste åtte boksene og dra vinnerlagene videre til de fire neste boksene osv.
- Informasjonen vil lagre seg i databasen og i det man lukker og starter applikasjonen på nytt vil programmet hente tilbake hvilke lag som har spilt mot hverandre

Beskrivelse av programkode

Ved starten av dette prosjektet var vi alle sammen lite erfarne rundt objektorientert programmering og prosjektering på nivåer som dette prosjektet. Kunnskapen og erfaringen vår lå sterkest ved ikke objektorienterte språk og ved mindre prosjekter. Gjennom koden kan en tydelig se at den ikke er løst på en god og objektorientert måte. Dette grunnet lite erfaring, og vi hadde stor fokus på å komme i gang og få funksjoner til å fungere, dermed så vi oss litt blinde rundt den objekt orienterte muligheten for å løse det på. Som f.eks., i stede for å lage slette-funksjoner i flere klasser, kunne vi laget en vi kaller på. Etterhvert som vi har jobbet videre med applikasjonen har vi lært mye vedrørende kode og løsningsmuligheter. Vi har sett tilbake på tidlig kode og ser nå at vi tenkte mer rett fram og statisk. Vi har gjort noen forsøk på objekt orientert kode utover i prosjektet, men har valgt å ikke bygge om all kode vi allerede har gjort pga. tidsrammene.

På bildene under har vi laget to metoder (add og retrieve) som ofte blir brukt i Windows Forms appen. Det er en veldig essensiell kode som brukes i store deler av applikasjonen og

flere ganger. Retrieve brukes til å hente ut fra databasen og legge inn i et gridview. Add brukes for å legge til i databasen fra gridview. Men det er et eksempel på ting vi kunne gjort annerledes og vil gjøre annerledes i videreutviklingen av applikasjonen. Disse metodene er kodet fra bunn hver gang de brukes i en klasse. Det er tungvint, unødvendig og dårlig practice.

```
98 private void retrieve()
99 {
100     dataGridView1.Rows.Clear();
101
102     // sql setning for å hente ut data
103     string sql = "SELECT * FROM event WHERE host_id = '" + textBox1.Text + "'";
104     //
105     cmd = new MySqlCommand(sql, con);
106
107     try
108     {
109         con.Open();
110         adapter = new MySqlDataAdapter(cmd);
111
112         adapter.Fill(dt);
113         // for hver rad programmet finner i databasen fyller den inn dette i tre rader
114         foreach (DataRow row in dt.Rows)
115         {
116             populate(row[0].ToString(), row[1].ToString(), row[2].ToString());
117         }
118
119         con.Close();
120
121         dt.Rows.Clear();
122     }
123     catch (Exception ex)
124     {
125         con.Close();
126     }
127 }
128
```

```
fjerdeside.cs forstreside.cs [Design] adminpanel.cs [Design] femteside.cs andreside.cs - X
eksamen2018wvf eksamen2018wvf.andreside conString
131 private void add()
132 {
133     // sql for å legge til i event tabellen i databasen
134
135     string conString = "INSERT INTO event (event_navn, host_id) VALUES('" + this.textBox2.Text + "', '" + this.textBox1.Text + "')";
136     cmd = new MySqlCommand(conString, con);
137
138     // resterende av kode er for å sjekke om event navnet finnes fra før av i eventene. Man får registrert om det ikke finnes fra før av, mens om det finnes fra før av vil man få opp en
139     // feilmelding som forklarer dette. Det ligger også en feilmelding hvis programmet ikke skulle klare å gjennomføre sql setningen. Dette er mest for får del for å feilsøke underveis.
140     if (db.antallRader("Select * from event where event_navn = '" + this.textBox2.Text + "' AND host_id = '" + this.textBox1.Text + "'") == 0)
141     {
142         try
143         {
144             con.Open();
145             if (cmd.ExecuteNonQuery() > 0)
146             {
147                 MessageBox.Show("Data er registrert til tabellen");
148                 clearTxts();
149             }
150
151             con.Close();
152             retrieve();
153         }
154         catch (Exception e)
155         {
156             MessageBox.Show("FEIL");
157             con.Close();
158         }
159     }
160     else
161     {
162         MessageBox.Show("Navnet finnes fra før av ");
163     }
164 }
165
166
167
168
169
170
```

På nettsiden, utviklet i asp.net, er det veldig relevant og ofte bruk for å vise informasjon på siden. Enten events, turneringer, lag, spillere osv. Dette har vi laget metoder for, i hver, klasse. Det samme designet for visning brukes gjennomgående på nettsiden, og ved bruk av relativt like metoder. Alle metodene mottar data fra databasen som innparameter, prosesserer dette og sender det videre som HTML kode. Altså er dette også noe som kunne, i stor grad, vært programmert mer objekt orientert. En av metodene er vist på bildet under:

```

49 public string vislag(DataSet ds)
50 {
51     int antallLag = 0;
52     String templagNavn = "";
53     String divcontent = "";
54
55     //skriver ut alle lagene med hjelp av foreach på siden hvis datasettet ikke er tomt
56     if (ds.Tables[0].Rows.Count != 0)
57     {
58         foreach (DataRow radevent in ds.Tables["lag"].Rows)
59         {
60             String t_navn = Convert.ToString(radevent["t_navn"]);
61             String lag_id = Convert.ToString(radevent["lag_id"]);
62
63             if (templagNavn == "" || templagNavn != Convert.ToString(radevent["l_navn"]))
64             {
65                 if (antallLag > 0)
66                 {
67                     divcontent += @"</ul></div></div>";
68                 }
69
70                 templagNavn = Convert.ToString(radevent["l_navn"]);
71                 divcontent += @"<div class='col-sm-4'>
72                     <div class='well well-lg'>
73                         <h3>" + templagNavn + "</h3>" +
74                         "<ul class='meta-search'>" +
75
76                         "<li><i class='fa fa-trophy'></i> <span>" + t_navn + "</span></li> <br/> </ul>" +
77                         "<a href='visdeltager.aspx?lag_id=" + lag_id + @"' class='btn btn-primary' title='visspillere'>Spillere</a> " +
78                         "<a href='administrerlag.aspx?lag_id=" + lag_id + @"' class='btn btn-primary' title='administrerlag'>Administrer lag</a> </div> </div>";
79
80             }
81         }
82     }
83     else
84     {
85         // Feilmelding hvis du ikke har noen registrerte lag
86         minelagtext.InnerHtml = "<div class='alert alert-danger'><strong> FEIL!!</strong> Du har ingen registrerte lag.</div>";
87     }
88     return divcontent;
89 }

```

Vi brukte lang tid på bracketen, den krevde mye grubling, prøving og feiling. Funksjonaliteten bak drag'n drop gikk til slutt greit. Med tanke på tiden det krevde og tiden vi hadde igjen, så ble det bestemt at per nå skal det kun være turneringer på 8 lag for å forenkle bracketen slik at den ble ferdig. De åtte boksene helt til venstre i bracketen (se bildet under) skal ikke ha noen videre funksjon enn drag'n drop, de skal bare fylles med påmeldte lag fra databasen. Derfor har de fått en enklere kode uten utvidede drag'n drop hendelser. I koden under ser vi på de seks første linjene at textbox 1 og 2 har drag'n drop funksjoner mot textbox 9. Dette er et eksempel på koden som brukes fra textbox 1 til 8. Den må kjøres slik mot, ikke bare textbox9, men fra 9 til og med 16.

Textbox 9 og resten av textboxene har fått sin funksjonalitet gjennom koden vi ser under de seks første linjene i bildet under. Disse må ha hendelser for bl.a. å kunne kjøre spørringer mot databasen ved drag'n drop hendelsene, slik som textbox9 kjører en update setning.


```

textBox1.DragEnter += new DragEventHandler(textBox9_DragEnter);
textBox1.MouseDown += new MouseEventHandler(textBox9_MouseDown);
textBox1.DragDrop += new DragEventHandler(textBox9_DragDrop);

textBox2.DragEnter += new DragEventHandler(textBox9_DragEnter);
textBox2.MouseDown += new MouseEventHandler(textBox9_MouseDown);
textBox2.DragDrop += new DragEventHandler(textBox9_DragDrop);

private void textBox9_DragDrop(object sender, DragEventArgs e)
{
    TextBox tb = (TextBox)sender;
    tb.Text = (string)e.Data.GetData(DataFormats.Text);

    if (tb.Text != null)
    {
        MySqlCommand cmd = con.CreateCommand();
        cmd.CommandType = CommandType.Text;
        cmd.CommandText = "UPDATE kamp SET lag1 = '"+tb.Text+" ' WHERE turnering_id = 26 AND bracketstep = 1";
        cmd.ExecuteNonQuery();
    }
    else
    {
    }
}

private void textBox9_DragEnter(object sender, DragEventArgs e)
{
    if (e.Data.GetDataPresent(DataFormats.Text))
    {
        e.Effect = DragDropEffects.All;
    }
    else
    {
        e.Effect = DragDropEffects.None;
    }
}

private void textBox9_MouseDown(object sender, MouseEventArgs e)
{
    TextBox tb = (TextBox)sender;
    tb.SelectAll();
    tb.DoDragDrop(tb.Text, DragDropEffects.Copy);
}

```

For å vise bracketen på nettsiden har vi fylt en datatable hvor radene i datatable fylles ut, en etter en, i labelene vi har brukt som bokser. Koden nedenfor viser bare koden bak de første boksene som kun skal vise påmeldte lag, ikke progresjonen i turnering. Kode for textboxer med utvidede hendelser for det er bygd likt som i bildet under, bortsett fra at der fylles to labels fra hver rad (lag1 og lag2). I koden sjekkes alle rader om de er tomme. Med dette oppnådde vi at systemet ville fylle bracketen med tomme labels, hvis datatable (dt eller dt1) var tom for rader, slik at programmet ikke krasjet pga manglende rader. Senere laget vi en trigger i databasen som fylte tomme kamper slik at bracketen bare oppdaterte radene i "kamp" tabellen. På denne måten ville vi også unngått krasj i systemet grunnet tomme rader. Hvordan bracketen fylles illustreres under:

```

48 //Før hver label fylles, vil gjøres en sjekk om det finnes noe videre rader i datatablen. Dette gjør også
49 if (dt.Rows.Count > 0)
50 {
51     Label11.Text = dt.Rows[0]["lag_navn"].ToString();
52
53     if (dt.Rows.Count > 1)
54     {
55         Label12.Text = dt.Rows[1]["lag_navn"].ToString();
56
57         if (dt.Rows.Count > 2)
58         {
59             Label13.Text = dt.Rows[2]["lag_navn"].ToString();
60
61             if (dt.Rows.Count > 3)
62             {
63                 Label14.Text = dt.Rows[3]["lag_navn"].ToString();
64
65                 if (dt.Rows.Count > 4)
66                 {
67                     Label15.Text = dt.Rows[4]["lag_navn"].ToString();
68
69                     if (dt.Rows.Count > 5)
70                     {
71                         Label16.Text = dt.Rows[5]["lag_navn"].ToString();
72
73                         if (dt.Rows.Count > 6)
74                         {
75                             Label17.Text = dt.Rows[6]["lag_navn"].ToString();
76
77                             if (dt.Rows.Count > 7)
78                             {
79                                 Label18.Text = dt.Rows[7]["lag_navn"].ToString();
80                             }
81                         }
82                     }
83                 }
84             }
85         }
86     }
87 }

```

Spesielle problemer og hvordan disse (eventuelt) ble løst:

- Ta med innlogget ID videre til andre sider/controllers i admin panelet. Dette ble løst ved at vi sendte brukernavnet som en string fra logginn til adminpanelet. Vi sender så brukernavnet videre fra adminpanelet til userpanelene. Dette gjøres slik:

- Sendes fra logginn.cs.

```
0 references
public static class LoginInfo
{
    public static string brukernavn;
}
```

- mottas i adminpanel.cs.

```
public adminpanel(string brukernavn)
{
}
```

- Sendes fra adminpanel.cs videre til userpaneler.

```
// vi har her med oss brukernavnet gjennom string brukernavn og sender brukernavnet videre til andre tiler på denne måten.
this.tredjeside1.ID = db.returnid("select bruker_id from admin WHERE brukernavn = '"+ brukernavn + "'");
this.andreside1.ID = db.returnid("select bruker_id from admin WHERE brukernavn = '" + brukernavn + "'");
this.fjerdeside1.ID = db.returnid("select bruker_id from admin WHERE brukernavn = '" + brukernavn + "'");
this.femteside1.ID = db.returnid("select bruker_id from admin WHERE brukernavn = '" + brukernavn + "'");
```

- Jobbet lenge og slet med CSS og bootstrap ettersom ingenting skjedde. Fant til slutt ut at nettleseren lagret cache som gjorde at designet ikke oppdaterte seg. Løst med å jevnlig slette caching.

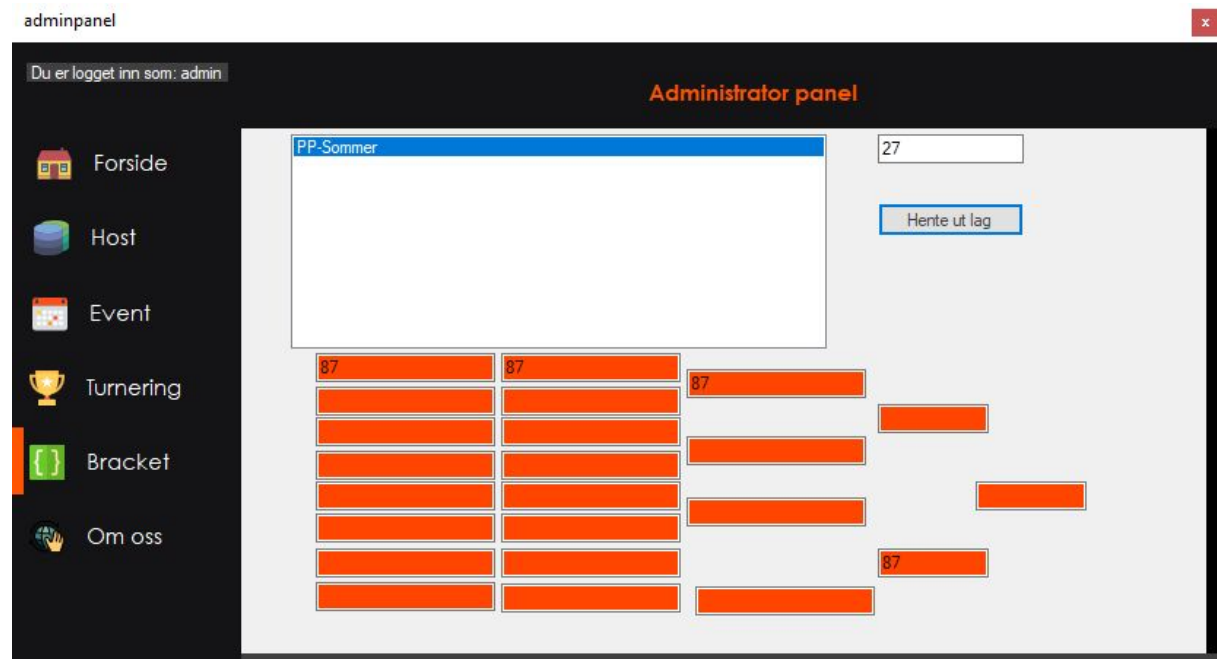
- Det var en utfordring å bruke et bilde som bakgrunn for turneringsbracket. Det ble løst ved å bruke labels som ble fylt fra databasen gjennom en datatable.

- Listeboksene i adminpanelet refresher seg ikke ved etter en operasjon. Skal man gjøre nye operasjoner, må man starte programmet på nytt slik at det refresher seg. Vi har prøvd flere muligheter:

- listBox.Refresh();
- this.listBox.Refresh();
- onClick - refresh muligheter
- og mye fler...

- Vi slet med at bracketen fylte databasen på flere steder enn der den skulle. Det ble til slutt løst med å effektivisere koden og gjøre den kortere. Dårlig og unødvendig kode skapte disse problemene.

- I bracketen har vi ikke fått den til å vise lagnavnet i boksen som ønskelig enda. Vi får bare vist lag_id per nå ettersom vi må bruke lag_id til å sende til databasen. Vi får vist lagnavn i de første boksene i .net, ettersom de ikke skal sende til database eller bruke verdien i boksen på samme måte som boksene videre. Å vise lagnavn i de første boksene fungerer ikke i Windows Forms enda, ettersom verdiene i de boksene skal brukes videre. Problemet er enda ikke løst. Under er problemet illustrert, første bilde er asp.net og andre er WF:



- Vi ønsket at listeboksen som inneholder "nasjon" i "administrespiller" skulle starte på den nasjonen som er valgt når spilleren ble registrert, slik at den holder seg på samme nasjon når du f.eks. bare skal endre navn. Dette ble løst med å hente ut nåværende nasjon i databasen (oldnasjon_id/oldnasjon_navn) og legge inn den som et nytt objekt i listeboksen (new ListItem) som legges på toppen. oldnasjon_id blir brukt som verdi når det brukes mot databasen, oldnasjon_navn er kun det som vises i listeboksen. Se bilde av kode:

```
// metode for å hente nasjon inn i liste boksen
private void hentenasjon()
{
    String koblingnasjon = "Host=cteam.mysql.domeneshop.no;Database=cteam;User=cteam;Password=AkVQirXSTZAg7b6";
    using (MySqlConnection cn = new MySqlConnection(koblingnasjon))
    {
        String oldnasjon_id = database.returnid("SELECT n.nasjon_id FROM deltager AS d INNER JOIN nasjon AS n ON n.nasjon_id = d.nasjon_id WHERE d.deltager_id = '" + d_id + "'");
        String oldnasjon_navn = database.returnid("SELECT n.land FROM deltager AS d INNER JOIN nasjon AS n ON n.nasjon_id = d.nasjon_id WHERE d.deltager_id = '" + d_id + "'");

        MySqlDataAdapter adp = new MySqlDataAdapter("SELECT land, nasjon_id from nasjon", cn);
        DataTable dt = new DataTable();
        adp.Fill(dt);
        if (dt.Rows.Count > 0)
        {
            nasjonlist.DataSource = dt;
            nasjonlist.DataTextField = "land";
            nasjonlist.DataValueField = "nasjon_id";
            nasjonlist.DataBind();
        }

        //Ønsket at listeboksen for nasjon startet på den allerede registrerte nasjonen ved endring av spiller. Har løst dette gjennom å legge den samme nasjonen inn som nytt item på toppen av listen.
        nasjonlist.Items.Insert(0, new ListItem(oldnasjon_navn, oldnasjon_id));
    }
}
```

Kjente problemer i windows forms

- Hvis en datagridview er tom og du forsøker å trykke i den, så vil programmet stoppe.
- Hvis man prøver å oppdatere/ slette noe i windows forms uten å ha valgt hva man skal oppdatere/ slette vil programmet stoppe.
- I det man gjør en endring i datagridview forsvinner informasjonen om man er nødt til å velge host på nytt og deretter event på nytt for å få ut den oppdaterte informasjonen i datagridview. Listen over eventer vil ikke refreshe seg og fylle seg på nytt med den samme informasjonen.

-

Dette er problemer som kan løses ved å bruke if eller else betingelser i koden. Dette ble ikke gjort noe med i denne omgang grunnet tidsbegrensninger.

Fremtidig arbeid / videre utvikling

Etter løsningen er ferdigstilt vil vi videreutvikle med mer avanserte funksjoner som:

Automatisering av poengfordeling.

- Vi ønsker å utvikle en løsning hvor man enten skriver inn matchpoeng for avansement, eller/og en løsning hvor man kobler applikasjonen til selve spillets database og oppdaterer automatisk gjennom spillets eget system.

Rangeringssystem. (Lagring av tidligere statistikk)

- Vi ser på mulighetene for å lagre spiller og lagstatistikk som skal være tilgjengelig gjennom den åpne ASP.net løsningen for publikum og andre interesserte.

Mer detaljerte profiler (Bilder og brukerinfo)

- Vi har og et mål om å legge til flere opplysninger om spillere og lag. Der man kan se bilder, muligens kontaktmulighet og andre praktiske og interessante punkter, bl.a. spiller statistikk, som kan sees av andre.

Mer profesjonelt og uniformt design.

- Designet vi har nå er ikke optimalt, og vi ønsker gjennom brukertesting å endre designet til noe som fremstår mer profesjonelt og konkurransedyktig på det kommersielle markedet.

Mulighet for å legge til knapper for sosiale medier gjennom adminpanelet som vises i asp.net

- I dag er sosiale medier en meget viktig kilde til informasjon og opplysninger om arrangementer og lignende. Vi ser på mulighetene for å lenke sosiale medieknapper til de respektive verter og eventer slik at man kan lett få tilgang til deres sider i sosiale medier, og potensielt til spiller og lag.

Legge til mulighet for fler turneringsoppsett med flere valg for størrelse på lag.

- Idag har vi 5 spillere, men det vil være snakk om f.eks 1-2 spiller for fifa, og eventuelt 8 spillere eller mer for andre typer spill. Dette vil være en del av adminpanelet hvor man velger ut ifra en liste hva slags turnering det er snakk om. Listen vil da tilpasse turneringsoppsettet til det spillet som er valgt. Vi ønsker også mulighet for flere størrelser på turneringene enn det er i dag. Per nå er det en fast mal på alle turneringer med maks 8 lag, vi ønsker å kunne lage mulighet for valg mellom oppsett på 8, 16, 32 og 64 lag.

Mulighet for salg av billetter til tilskuere og evt. Spillere.

- Vi har planer om å legge til mulighet for salg av billetter til events, eller lenker til der billetter selges.

Mer info om lag og spillere og mulighet for bedre posisjonering av denne infoen.

- Som med spillere ønsker vi at laginformasjon og statistikk gir best mulig innblikk, og vi ønsker at denne informasjonen skal være lett synlig for besøkende.

Statistikk for spillere og lag.

- Dette punktet går ut på mye av det samme som rankingsystemet, men det kan være ønske om å ta med statistikk som strekker seg utenom bare spillbasert informasjon. Eksempelvis tenker vi på hvor mange events personen og laget har vært med på, og andre type data som kan gi et innblikk i spillerens livsløp.

Mobilapplikasjon.

- Vi har et ønske om å utvikle løsningen som en mobilapplikasjon. Vi må se på mulighetene og hva som eventuelt skulle være applikasjonens funksjon.

Muligens legge hele løsningen inn i asp.net.

- Vi har og pratet om å lage hele løsningen fra windows form om til ASP.net for å gi folk muligheten til å gjøre alt applikasjonen tilbyr uten å laste ned noe.

Legge til kvalifiseringsdel. (Seeding)

- Dette er noe som de fleste turneringer har. En kvalifisering og et gruppespill. Vi har derfor planer om å lage mulighet for løsningen til å håndtere dette.

Spillerprofil/lagprofil (Lagleder lager lagprofil/Spillere lager selv spillerprofil.)

- Spillere kan koble forskjellige nickname til forskjellige spill.

Mulighet for å klikke på spillere for spillerinfo.

- Det som menes her er at man skal kunne klikke på en spiller på ASP.net siden og få ut informasjon om spilleren. (statistikk og annen relevant data)

Listeboksene i adminpanelet refresher seg ikke. Det må fikses.

Kilder

<https://stackoverflow.com/questions/17292366/hashing-with-sha1-algorithm-in-c-sharp>

https://stackoverflow.com/questions/1733912/how-do-i-handle-dragging-of-a-label-in-c?utm_medium=organic&utm_source=google_rich_qa&utm_campaign=google_rich_qa

<https://www.youtube.com/watch?v=d0J3VKBA4Xs>

https://stackoverflow.com/questions/6508736/response-redirect-passing-a-value-dynamically?utm_medium=organic&utm_source=google_rich_qa&utm_campaign=google_rich_qa

<https://forums.asp.net/t/446710.aspx?How+do+I+wait+for+10+seconds+before+doing+a+response+redirect+>

https://stackoverflow.com/questions/17468956/how-to-add-an-item-to-a-drop-down-list-in-asp-net?utm_medium=organic&utm_source=google_rich_qa&utm_campaign=google_rich_qa

https://stackoverflow.com/questions/22635042/how-to-get-column-values-from-datable-to-textbox?utm_medium=organic&utm_source=google_rich_qa&utm_campaign=google_rich_qa

<https://stackoverflow.com/questions/7765548/get-a-single-value-from-dataset-in-asp-net>

<https://stackoverflow.com/questions/5252934/getting-a-count-of-rows-in-a-datable-that-meet-certain-criteria>

<https://challonge.com/no>

<http://www.dreamincode.net/forums/topic/163976-passing-variables-from-a-form-to-a-user-control/>

<https://fontawesome.com/icons?d=gallery&c=business>

https://stackoverflow.com/questions/2618073/bring-winforms-control-to-front?utm_medium=organic&utm_source=google_rich_qa&utm_campaign=google_rich_qa

https://stackoverflow.com/questions/15003095/getting-value-of-selected-item-in-list-box-as-string?utm_medium=organic&utm_source=google_rich_qa&utm_campaign=google_rich_qa

Vi har generelt gjennom hele prosjektet hatt god bruk for Microsofts eget .net forum (<https://forums.asp.net/>) og stackoverflow (<https://stackoverflow.com/>) som ressurser for å finne informasjon og løsninger på diverse problemstillinger.

Ola Aulesjord Olsen som studieassistent