

Appendix

Table of contents

1	User story 2	2
2	User story 8	3
3	User story 12	5
4	User Story 14	6
5	Meeting Reports	7
5.1	Date: Tuesday, April 7th	7
5.2	Date: Monday, April 13th	8
5.3	Date: Tuesday, April 14th	9
5.4	Date: Friday, April 17th	11
5.5	Date: Monday, April 20th	12
5.6	Date: Wednesday, April 22nd	13
5.7	Date: Friday, April 24th	14
5.8	Date: Monday, April 27th	15
5.9	Date: Tuesday, April 28th	16
5.10	Date: Thursday, April 30th	17
5.11	Date: Monday, May 4th	17
5.12	Date: Tuesday, May 5	18
5.13	Date: Thursday, May 7th	18
5.14	Date: Monday, May 11th	19
5.15	Date: Wednesday, May 13th	20
5.16	Date: Monday, May 18th	20
5.17	Date: Wednesday, May 20th	20
5.18	Date: Monday, June 1st	21
6	First meeting with costumer - Requirements	22

1 User story 2

```

public int AddMember(Member member)
{
    try
    {
        String SQLQuery;
        if (member.Position == null)
        {
            SQLQuery = "INSERT INTO Member VALUES(@Name, @Email, @Birthday,
@Phone, @Address, @City, @ZipCode, @MemberType, @ApprovedBy, @Created, null);SELECT
CAST(scope_identity() AS int)";
        }
        else
        {
            SQLQuery = "INSERT INTO Member VALUES(@Name, @Email, @Birthday,
@Phone, @Address, @City, @ZipCode, @MemberType, @ApprovedBy, @Created, @Role);SELECT
CAST(scope_identity() AS int)";
        }
        SqlCommand command = new SqlCommand(SQLQuery, _sqlConnection);
        SqlParameter param = new SqlParameter("@Name", member.Name);
        command.Parameters.Add(param);
        param = new SqlParameter("@Email", member.Email);
        command.Parameters.Add(param);
        param = new SqlParameter("@Birthday", member.Birthday);
        command.Parameters.Add(param);
        param = new SqlParameter("@Phone", member.Phone);
        command.Parameters.Add(param);
        param = new SqlParameter("@Address", member.Address);
        command.Parameters.Add(param);
        param = new SqlParameter("@City", member.City);
        command.Parameters.Add(param);
        param = new SqlParameter("@ZipCode", member.ZipCode);
        command.Parameters.Add(param);
        param = new SqlParameter("@MemberType", member.MemberType);
        command.Parameters.Add(param);
        param = new SqlParameter("@ApprovedBy", member.ApprovedBy);
        command.Parameters.Add(param);
        param = new SqlParameter("@Created", DateTime.Now);
        command.Parameters.Add(param);
        if (member.Position != null)
        {
            param = new SqlParameter("@Role", member.Position);
            command.Parameters.Add(param);
        }
        _sqlConnection.Open();
        Int32 memberId = (Int32)(command.ExecuteScalar());
        return memberId;
    }
    catch (SqlException sqlEx)
    {
        ExceptionUtility.LogException(sqlEx, "Sql");

        throw new Exception(sqlEx.Message);
    }
    finally
    {
        _sqlConnection.Close();
    }
}

```

```
    }
}
```

2 User story 8

```
public void CreateInvoicePDF(string mail, string price, string invoiceId)
{
    Member member = _memberController.GetMemberObject(mail);
    string newPrice = price.Replace('.', ',');
    Dictionary<string, string> invoiceInfo = _memberController.GetInvoiceInfo();
    try
    {
        // Step 1: Creating System.IO.FileStream object
        string fileName = Server.MapPath("PDF/Faktura for " + member.Name +
        ".pdf");

        using (
            FileStream fs = new FileStream(fileName,
                FileMode.Create, FileAccess.Write, FileShare.None))
        // Step 2: Creating iTextSharp.text.Document object
        using (Document doc = new Document())
        // Step 3: Creating iTextSharp.text.pdf.PdfWriter object
        // It helps to write the Document to the Specified FileStream
        using (PdfWriter writer = PdfWriter.GetInstance(doc, fs))
        {
            // Step 4: Openning the Document
            var worker = new HTMLWorker(doc);
            doc.Open();
            worker.StartDocument();

            // Step 5: Adding a paragraph
            // NOTE: When we want to insert text, then we've to do it through
            creating paragraph

            Paragraph headlieParagraph = new Paragraph("K19s Venner");
            headlieParagraph.Font.Size = 25;

            Paragraph memberIdParagraph = new Paragraph("Medlems nr: " +
            member.Id);

            Paragraph memberNameParagraph = new Paragraph(member.Name);
            Paragraph memberAddressParagraph = new Paragraph(member.Address);
            Paragraph memberZipCodeParagraph = new Paragraph(member.ZipCode + " "
            + member.City);

            Paragraph memberEmailParagraph = new Paragraph(member.Email);

            iTextSharp.text.Image gif =
            iTextSharp.text.Image.GetInstance(Server.MapPath("../Images/logo.png"));
            gif.SetAbsolutePosition(450, 720);

            Paragraph infoParagraph1 = new Paragraph("K19s Venner");
            Paragraph infoParagraph2 = new Paragraph("Adresse: " +
            invoiceInfo["Address"].TrimEnd());
            Paragraph infoParagraph3 = new Paragraph("By: " +
            invoiceInfo["ZipCode"] + " " + invoiceInfo["City"].TrimEnd());
            Paragraph infoParagraph4 = new Paragraph("Telefon: +45 " +
            invoiceInfo["Phone"].TrimEnd());
            infoParagraph1.Alignment = Element.ALIGN_RIGHT;
            infoParagraph2.Alignment = Element.ALIGN_RIGHT;
            infoParagraph3.Alignment = Element.ALIGN_RIGHT;
```

```

infoParagraph4.Alignment = Element.ALIGN_RIGHT;

Paragraph p =
    new Paragraph(
        new Chunk(new iTextSharp.text.pdf.draw.LineSeparator(0.0F,
100.0F, BaseColor.WHITE,
            Element.ALIGN_LEFT, 1)));

Paragraph invoiceNumberParagraph = new Paragraph("Faktura nummer: " +
invoiceId);
Paragraph dateParagraph = new Paragraph("Dato: " +
DateTime.Today.ToShortDateString());
Paragraph bankParagraph = new Paragraph("Bank: " +
invoiceInfo["BankName"].TrimEnd());
Paragraph regParagraph = new Paragraph("Reg. nr: " +
invoiceInfo["RegNr"].TrimEnd());
Paragraph acNrParagraph = new Paragraph("Konto nr: " +
invoiceInfo["KontoNr"].TrimEnd());

dateParagraph.Alignment = Element.ALIGN_RIGHT;
invoiceNumberParagraph.Alignment = Element.ALIGN_RIGHT;
bankParagraph.Alignment = Element.ALIGN_RIGHT;
regParagraph.Alignment = Element.ALIGN_RIGHT;
acNrParagraph.Alignment = Element.ALIGN_RIGHT;

Paragraph invoiceParagraph = new Paragraph("Faktura");
invoiceParagraph.Font.Size = 25;
string html =
    "<br/><table border='1'><tr><td width='\"80%\"' style='\"text-
align:center\"><b>Betegnelse</b></td><td width='\"20%\"' style='\"text-
align:center\"><b>Beløb</b></td></tr><tr><td width='\"80%\"' style='\"padding-
left:10px;padding-bottom:5px\">Medlemskab</td><td width='\"20%\"' style='\"text-
align:right\">" + newPrice + "</td></tr></table><br/>";
    TextReader reader = new StringReader(html);

string total =
    "<br/><table border='0'><tr><td width='\"80%\"' style='\"text-
align:right\"><b>Total DKK</b></td><td width='\"20%\"' style='\"text-align:right\">" +
    newPrice + "</td></tr></table><br/>";
    TextReader reader2 = new StringReader(total);
Paragraph messageParagraph = new
Paragraph(invoiceInfo["Message"].TrimEnd());
doc.Add(headlieParagraph);
doc.Add(new Paragraph("\n"));
doc.Add(gif);
doc.Add(memberIdParagraph);
doc.Add(memberNameParagraph);
doc.Add(memberAddressParagraph);
doc.Add(memberZipCodeParagraph);
doc.Add(memberEmailParagraph);
doc.Add(infoParagraph1);
doc.Add(infoParagraph2);
doc.Add(infoParagraph3);
doc.Add(infoParagraph4);
doc.Add(p);
doc.Add(invoiceNumberParagraph);
doc.Add(dateParagraph);

```

```

        doc.Add(bankParagraph);
        doc.Add(regParagraph);
        doc.Add(acNrParagraph);

        doc.Add(invoiceParagraph);
        worker.Parse(reader);
        worker.Parse(reader2);
        doc.Add(messageParagraph);
        worker.Close();
        // Step 6: Closing the Document
        doc.Close();
    }
    Response.Write(SharePointHelper.UploadInvoice(fileName, member.Name + " "
+ member.Id, (HttpContext.Session["clientContext"]) + " - ");
    SendInvoice(mail, fileName);
}
// Catching iTextSharp.text.DocumentException if any
catch (DocumentException de)
{
    throw de;
}
// Catching System.IO.IOException if any
catch (IOException ioe)
{
    throw ioe;
}
}

```

3 User story 12

```

private void ShowNonActivatedMembers(List<Member> nonActivatedMembers)
{
    UserPanel.Controls.Clear();
    foreach (var member in nonActivatedMembers)
    {
        HtmlAnchor nameLabel = new HtmlAnchor { HRef = "#", InnerText =
member.Name };
        nameLabel.Attributes.Add("onclick", "showUser(" + member.Id + ")");
        UserPanel.Controls.Add(nameLabel);

        UserPanel.Controls.Add(new Label { Text = "<br/>" });

        Label emailLabel = new Label { Text = "E-mail: " + member.Email + "<br
/>" };
        Label birthdayLabel = new Label { Text = "Fødselsdag: " +
member.Birthday.Date.ToString("d") + "<br />" };
        Label phoneLabel = new Label { Text = "Telefon: " + member.Phone + "<br
/>" };
        Label addressLabel = new Label { Text = "Adresse: " + member.Address +
"<br />" };
        Label cityLabel = new Label { Text = "By: " + member.City + "<br />" };
        Label zipCodeLabel = new Label { Text = "Postnr: " + member.ZipCode +
"<br />" };

        Button acceptButton = new Button();
        acceptButton.Text = "Godkend";
        int memberId = member.Id;
        string memberEmail = member.Email;
    }
}

```

```

string memberName = member.Name;

acceptButton.Click += (sender, EventArgs) =>
{
    AcceptUser(memberId, memberEmail, memberName);

};

Button declineButton = new Button();
declineButton.Text = "Afvis";
declineButton.Click += (sender, EventArgs) =>
{
    DeclineUser(memberId);
};

HtmlGenericControl div = new HtmlGenericControl("div");
div.ID = member.Id.ToString();
div.Attributes.Add("class", "collapse");
div.Controls.Add(emailLabel);
div.Controls.Add(birthdayLabel);
div.Controls.Add(phoneLabel);
div.Controls.Add(addressLabel);
div.Controls.Add(cityLabel);
div.Controls.Add(zipCodeLabel);
div.Controls.Add(acceptButton);
div.Controls.Add(declineButton);
div.Controls.Add(new Label { Text = "<br/><br/>" });
UserPanel.Controls.Add(div);
}
}

```

4 User Story 14

GridView markup:

```

<asp:GridView ID="GridView1" runat="server" AllowSorting="True" AlternatingRowStyle-
CssClass="Alternate" AutoGenerateColumns="False" CssClass="GridView"
DataSourceID="SqlDataSource1" OnRowDataBound="GridView1_OnRowDataBound"
AllowPaging="True">
    <Columns>
        <asp:BoundField DataField="Name" HeaderText="Navn"
SortExpression="Name" />
        <asp:BoundField DataField="Email" HeaderText="Email"
SortExpression="Email" />
        <asp:BoundField DataField="MemberType" HeaderText="Status"
SortExpression="MemberType" />
        <asp:HyperLinkField DataNavigateUrlFields="Email"
DataNavigateUrlFormatString="EditProfile.aspx?edit=member&email={0}" Text="Redigér"
/>
        <asp:HyperLinkField />
    </Columns>
    <HeaderStyle BackColor="#0000FF" ForeColor="#000000" />
    <SortedAscendingHeaderStyle BackColor="#0000CC" />
</asp:GridView>
<asp:SqlDataSource ID="SqlDataSource1" runat="server"
ConnectionString="Data Source=qjaugbgpqz.database.windows.net;Initial
Catalog=membersystem;User ID=membersystem;Password=U'TP9UDkQ+eN(q$y"
ProviderName="System.Data.SqlClient" SelectCommand="SELECT [Name], [Email], [MemberType]
FROM [Member] WHERE ([Name] LIKE '%' + @Name + '%')">

```

```

        <SelectParameters>
            <asp:ControlParameter ControlID="MemberNameTextBox"
Name="Name" PropertyName="Text" Type="String" />
        </SelectParameters>
    </asp:SqlDataSource>

```

5 Meeting Reports

5.1 Date: Tuesday, April 7th

Time: 14:00

Duration: 2 hours

Attendees: Meik Børner(Customer), Salik and Nicolai(Team)

Location: Fjeldhammervej 17, Rødovre

Agenda: First meeting with the customer. Customer explains the project to the team. Team discusses with customer about procedure of the project

Items Discussed with the Customer:

- 1) Introduction
 - Meet and greet
 - Customer talks about the organization
 - Customer shows his office
 - Customer offers space for us to work if we want to work at his office (Available Monday-Thursday)
- 2) The project
 - Customer explains the problem
 - Customer writes down requirements for the product
 - Team asked customer to prioritize the stories. Customer scanned in all requirements together with priority of the stories and suggestions to solutions.
- 3) Discussion
 - Team discussed the requirements with the customer for better understanding of the project
 - Discussion of the technical requirements and different solutions to problems
 - Discussion of roles of individual members of the team
 - Team discussed future meetings with customer, and how the team wants to work with the project
 - All agreed to work with the SCRUM methodology meeting every end of a sprint with the customer (2 weeks/sprint)
 - The team wishes to start the first sprint on Monday, 13th April

5.2 Date: Monday, April 13th

Time: 10:00

Duration: 4 hours

Attendees: All team members

Reporter: Kasia

Agenda: General meeting, sharing concepts, installing and configuration of software.

Meeting progress:

1) Role distribution:

Nicolai – C# coding (Visual Studio)

Salik – database (SQL server, Azure)

Kasia – report and UI design (Axure)

○ Practices we are going to use:

- Pair programming
- Planning game
- Whole team
- Continuous integration
- Small releases
- Collective code ownership
- Test-driven development
- Backlogs
- Scrum events
- User story
- Retrospective
- Velocity tracking

2) Plan:

○ Sorting and naming User Stories.

Our User Stories (from 1 to 13):

- Login
- CreateMember
- EditData
- EditMember
- DeleteMember
- EditDocument
- EditMemberDocument
- StartSubscription
- SendInvoices
- SendReminders
- SignUp
- ApproveDeclineMember

- UpdateEmail
- Logging into the Sharepoint
- Setting up the Visual Studio Online and creating a new project.
We spent some time on this because we had a bit of a problem with sharing the project in Visual Studio Online. We had to figure out how to use it properly so that everyone can see the changes we made in it.
- Discussing the general concept about the project we're doing and diving the User Stories into Sprints. We had to decide which User Stories were the most difficult. Our scale was 1-10. We decided to do 3 sprints:
 - First Sprint (**April 14 – 29**)
User Stories: 1 – 5
 - Second Sprint (**April 29 - May 13**)
User Stories: 6, 7, 8, 11, 12 (13)
 - Third Sprint (**May 13 – 27**)
User Stories: 9, 10
- Creating a database design with primary and foreign keys
- Dividing tasks among members according to every User Story
- Creating a project in Microsoft Azure
Having not used this program before, we had a problem determining which option would be the best for us to use. We decided that Salik, who is responsible for databases, will read about Azure and make a decision.
We also had a problem with creating a free membership in Azure. It seems like that option is unavailable and you have to pay to use it. We decided to contact our Customer and ask them what we should do with it and whether they will pay for the subscription.

5.3 Date: Tuesday, April 14th

Time: 10:00

Duration: 4 hours

Attendees: All team members

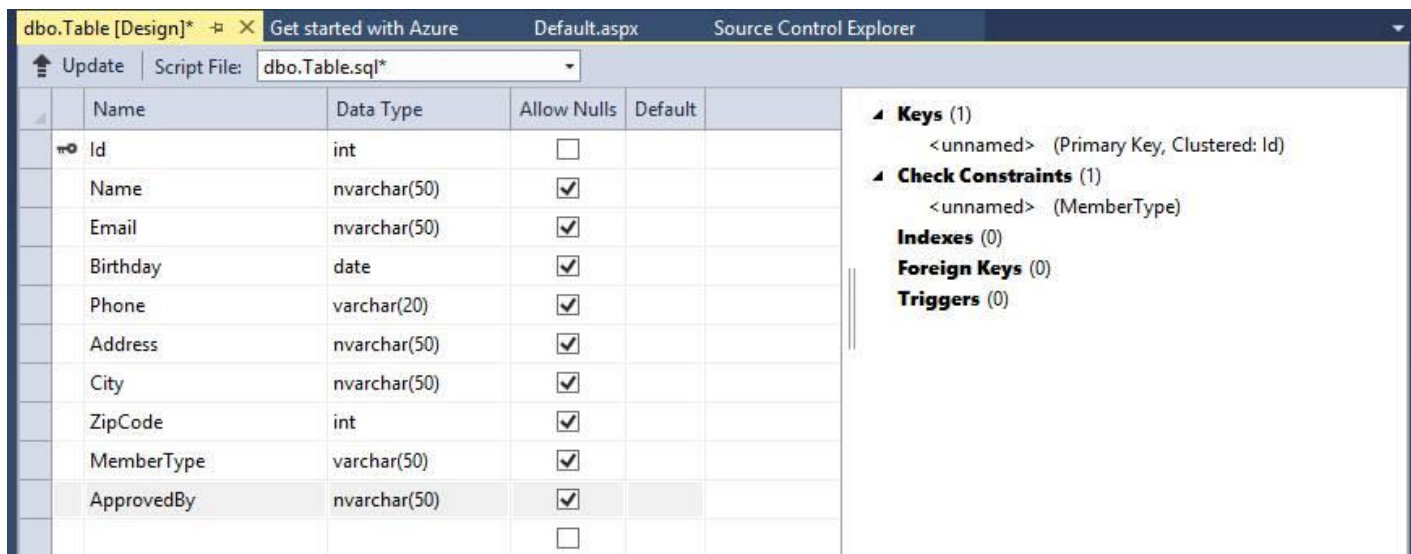
Reporter: Kasia

Agenda: Finishing the work we started the day before, start of the Sprint 1.

Meeting progress:

1. Finishing the configuration of software.

- Registration on Azure
We have decided on a paid subscription to Azure software in order to properly do our work. Also the customer required that we use Azure.
- Creating a database server.
Our attempt at creating a server was interrupted by a few problems. First of all, we couldn't connect one of our computers to the server. We thought it was a firewall problem and made a rule in our firewall but it didn't solve anything. It turned out we could connect directly to the Azure SQL database from Visual Studio, and that solved every problem we had so far with the SQL database. That was a solution to our problem, which allowed us to successfully connect Salik's laptop to the server.
 - We then proceeded to connect the rest of our computers to the database server. That went well and without any complications.
- Making a Member table in database.



	Name	Data Type	Allow Nulls	Default
PK	Id	int	<input type="checkbox"/>	
	Name	nvarchar(50)	<input checked="" type="checkbox"/>	
	Email	nvarchar(50)	<input checked="" type="checkbox"/>	
	Birthday	date	<input checked="" type="checkbox"/>	
	Phone	varchar(20)	<input checked="" type="checkbox"/>	
	Address	nvarchar(50)	<input checked="" type="checkbox"/>	
	City	nvarchar(50)	<input checked="" type="checkbox"/>	
	ZipCode	int	<input checked="" type="checkbox"/>	
	MemberType	varchar(50)	<input checked="" type="checkbox"/>	
	ApprovedBy	nvarchar(50)	<input checked="" type="checkbox"/>	
			<input type="checkbox"/>	

Keys (1)
<unnamed> (Primary Key, Clustered: Id)

Check Constraints (1)
<unnamed> (MemberType)

Indexes (0)

Foreign Keys (0)

Triggers (0)

- Connecting our database with Visual Studio.
In order to properly enter the data from a website into the database, we had to create a link between it and Visual Studio code.

2. SPRINT 1.

- Adding the second User Story (CreateMember).
We decided it was the best for us to start with this one instead of the first one, as by creating a database, we had already done part of the work. We just continued then, to finish the second User Story.
- Creating a simple prototype of UI for second User Story.
We used Visual Studio UI designer to create a few textboxes that will later be filled with data about new members.

- Designing the main page.
We discussed a bit about how our page will look in different resolutions and decided that we will come back to it later.
- Below the textboxes we put two buttons (Create and Cancel), depending on which one the Administrator is going to choose. Unfortunately, we realized that the buttons are not working. At first we thought it was a php problem, but after further investigation it turned out we just forgot to code the method for adding a Member.

5.4 Date: Friday, April 17th

Time: 10:00

Duration: 4 hours

Attendees: All team members

Reporter: Kasia

Agenda: SCRUM meeting, SPRINT 1.

Meeting progress:

1. SCRUM meeting
 - We asked each other what we had done last time and between the meetings. We also agreed to help each other if any of us had any problem with the task they were given.
 - Nicolai and Salik added the second User Story.
 - Salik then created some tests for the part of code we had made so far.
 - Kasia started working on UI in Axure, but had some problem with adding it into Visual Studio. The rest of the team then helped her with that.
 - We did not encounter any bigger problems, so far everything was going good.
2. Continuation of SPRINT 1.
 - We weren't sure how to code the part where you enter the birthday of a new member. We couldn't decide on a format. We wanted to use droplists or a calendar but Salik said that it would be easier for him to write a test for it if it was just 3 simple textboxes. We agreed that it would be simpler to implement.
 - A new, better UI was created in Axure. The HTML code generated by the program was included in our code. It turned out Axure solved our problem with the birthday textbox by allowing us to use a calendar. There were still some things that had to be regulated between the HTML file and the code in Visual Studio.
 - We also started implementing the first User Story (Login) into our code. It was easy to do because we only had to add SQL commands and code the part that checked whether the user is an Administrator or a Member.

5.5 Date: Monday, April 20th

Time: 10:00

Duration: 4 hours

Attendees: Salik and Nicolai

Reporter: Nicolai

Agenda: Work on finishing User Story 3 and 4 and implementing search in the product

Meeting progress:

1. SCRUM meeting
 - We talked about what we did last time. Salik and Nicolai almost finished User Case 3, and the team agrees that User Case 3 and 4 are much alike, so they will be done quickly after each other.
 - Salik talked about what he did last time, and that today he will work on finishing User Story 3 and together with Nicolai finish User Story 4.
 - Nicolai will be doing the search function while Salik is doing User Story 3, and together they will try to finish User Story 4.
 - Our only problem so far, is that our UI designer, Kasia, is leaving the country to finish her internship. The group has decided to continue the project and work remotely with Kasia over Skype/Facebook. Besides that, the group haven't encountered any major problems, and the minor problems and questions from last time, the group has solved right after the SCRUM meeting.
2. Planning
 - The group has looked at the burndown chart for the first week. If we finished what we wanted to do today, the group will be a bit ahead of the schedule. Since our first meeting with the customer, there has been added an extra User Story, which is the search function. The customer didn't think of that, and we will have to plan around it.
 - Nicolai has talked with the customer about setting up a meeting for the end of sprint 1. It is scheduled to Tuesday the 28th of April.

5.6 Date: Wednesday, April 22nd

Time: 10:00

Duration: 4 hours

Attendees: Salik and Nicolai

Reporter: Nicolai

Agenda: Work on finishing the rest of the User Stories, setting up meeting with supervisor and a meeting with customer

Meeting progress:

1. SCRUM meeting
 - We talked about what we did last time. Salik finished Use Case 3 and together with Nicolai, he finished Use Case 4. Nicolai finished the search function and worked together with Salik on finishing Use Case 4. The search function is a part of Use Case 4 and 5, so it was important to get it working with the product.
 - Today Salik will finish Use Case 5 and set up a meeting with our supervisor, Michael Claudius.
 - Nicolai will be integrating the shippable product into SharePoint and do a lot of testing. He will also set up the first sprint review meeting with the customer.
 - Last time, Salik encountered a problem with pulling user ID from the database. The problem was fixed when the team reviewed the code together.
2. Progress
 - The team is almost finished with Sprint 1, and Friday's SCRUM meeting will have information of what we did today and how we will be ready for the Sprint Review.
 - The team did a lot of pair programming today and helped/researched each others issues.
 - The team has set up meetings with customer and supervisor. The sprint review will be on April the 28th 10:00 am. The supervisor meeting will be on Monday at 9 am.
 - Prototypes of icons for the UI were created by Kasia, and uploaded for the rest of the team to view.

5.7 Date: Friday, April 24th

Time: 10:00

Duration: 4 hours

Attendees: Salik and Nicolai

Reporter: Nicolai

Agenda: Finish code and UI for the first sprint.

Meeting progress:

1. SCRUM meeting
 - Salik and Nicolai did a lot of code last time, and almost finished all the code tasks in sprint 1. It was tested with SharePoint, and it is 100% sure that we will have a shippable product for the sprint review.
 - Today the team will make the last code and try to do the User Interface. The User Interface needs to generally look a bit better and fit in with SharePoint's design. The team wants to finish code and UI today, so they can focus on testing the system on Monday and make it ready for review.
 - Last time the team had some small problems with the code. The problems consisted of pulling dates from the database into textboxes and the right format of those. They also had some JavaScript that needed to work with client side code. The problem was fixed with pair programming.
2. Getting ready for Sprint Review
 - The team has worked towards getting ready for the sprint review. Team has talked about working a bit in the weekend, so they can test all the functions together on SharePoint on Monday. Monday's work will consist of testing and talking with supervisor, if everything goes according to the plan.

5.8 Date: Monday, April 27th

Time: 09:00

Duration: 4 hours

Attendees: Salik and Nicolai

Reporter: Nicolai

Agenda: Supervisor(Michael) meeting, integrating code with product and testing

Meeting progress:

1. SCRUM meeting
 - Kasia worked on User Interface and its icons for the different functions.
 - Salik and Nicolai worked on finishing coding in the sprint. The coding was finished, and now there is only testing and integrating left.
 - Today the team will be integrating the code with SharePoint, and we will start getting ready for 2nd sprint, which starts on Wednesday 29th.
 - Kasia will return to Denmark soon, and will be joining the team as we start the 2nd sprint.
2. Supervisor meeting
 - The first hour of today was spent on talking with Michael about the project and the report.
 - Talked about how to start the report, and what the report should contain.
 - Michael gave the team a document with notes/guidelines.
 - Team made an agreement with Michael that they should have one meeting pr week.

5.9 Date: Tuesday, April 28th

Time: 10:00

Duration: 2 hours

Attendees: Meik Børner(Product Owner/Customer), Salik and Nicolai (The team)

Reporter: Nicolai

Agenda: Sprint Review Meeting

Location: Fjeldhammervej 17A, Rødovre

Meeting progress:

1. Showing demo
 - The team showed customer the demo of the product so far. It went well without problems.
 - The team asked question and had suggestion to some things, and there was a good dialogue of the future of the product.
2. Feedback
 - The customer wanted some changes for the product. He wanted a better look in the search function, and he gave a good example of how it should look.
 - He wanted the app to open in the browser and not through SharePoint, so only 'Bestyrelse'-members will be seeing the SharePoint site and regular members will be sent directly to the application.
 - The customer had some suggestions to making a database design and a more visual sprint backlog. This will be reviewed by team members in the Sprint Retrospective.
3. Success
 - Overall it was a success, and both parts decided to move on to next sprint(after the changes are made), and a date about two weeks from now is to be set for the second sprint review.

5.10 Date: Thursday, April 30th

Time: 10:00

Duration: 4 hours

Attendees: Salik and Nicolai

Reporter: Nicolai

Agenda: Sprint Retrospective, making changes

Meeting progress:

1. Sprint Retrospective
 - Team started the day by holding a Sprint Retrospective meeting about the course of the project. They discussed which practices they should start doing, stop doing and continue doing. There was a good discussion and an overall agreement to the practices, and it can be found in the Sprint Retrospective document.
2. Making changes
 - Before starting the sprint tomorrow, Friday 1st of May, the team decided to implement the changes the customer asked for. If it can't be ready today, they will be making it in small bits through the 2nd sprint.

5.11 Date: Monday, May 4th

Time: 10:00

Duration: 4 hours

Agenda: Sprint 2

Meeting progress: SignUp User Story

- After implementing the changes our customer asked for, we were ready to move into Sprint 2.
 - Salik has already done the ApproveDeclineMember User Story, so we can move into the next one - SignUp.
 - Adding e-mail signup and verification into the code.
 - Testing the method using our own e-mails. Unfortunately, the code was not working and we had to find out why. We tried different e-mail addresses (gmail account), but that did not help much. As it turned out, part of the code was written in a wrong order and we had to switch it. After implementing changes, the e-mail sign up was working, but only for gmail and office365 accounts.
 - The team encountered a problem with the design when we were trying to implement the verification code for the e-mail. We weren't sure whether we should disable the fields after they're filled or should still allow the user to edit them. Another idea was to create a new window with a place for the code, only printing out the data the User entered. That seemed like the best idea and we went with it.

- We also created a database for Danish cities, so that when you enter your ZIP-code, it automatically finds your city. Unfortunately, when we tested it, the formatting turned out to be all wrong, with too many spaces everywhere. We had to work on fixing that.
- We spent the rest of the meeting polishing the details and finishing the SignUp User Story.

5.12 Date: Tuesday, May 5

Time: 9:00

Duration: 4 hours

Agenda: Sprint 2

Meeting progress: Meeting with Michael

- Meeting with the supervisor:
 - Michael gave us some info about our report. He told us to put a table with our strengths and weaknesses into it
 - In general, he said we are on a good path and should carry on with what we have, adding a few more things according to his guidelines
- EditProfile and CreateMember.
 - The user can't click save button if input is empty - Done by Salik with JavaScript.
- We are a little behind according to burn down chart, so we will work some extra in the weekend to finish User Stories
- What we have implemented into the code so far:
 - Logging (error-logging)
 - Documentation (Doxygen)
 - Increased Usability/Security for the forms (edit, create, sign up)
- Team discussed the invoice-subscription User Story.
 - Problem: How to send attachments (pdf-files) through C# + Email
 - Problem: How to generate invoice automatically based on info
- Plan on meeting Customer Thursday 14th May (Will contact him tomorrow)
- Team made a deal on writing some report at home and have something to show each other when we meet next (Thursday 7th May)

5.13 Date: Thursday, May 7th

Time: 12:00

Duration: 2 hours

Agenda: Summary, discussion about the future of the project.

Meeting progress:

- We looked at what we have, merged together stuff we did at home and discussed our progress so far.

5.14 Date: Monday, May 11th

Time: 10:00

Duration: 4 hours

Agenda: Finishing the 2nd sprint.

Meeting progress:

- The invoice User Stories
 - The User Story required changing the logs from C#, which was very difficult.
 - Salik and Nicolai worked together on finishing this User Story due to its difficulty.
 - The plan was to create a template in Excel that only needed filling the gaps and then convert it into PDF file.
 - Unfortunately, the User Stories containing the invoice turned out to be too difficult. We also could not synchronize the software. We are going to inform the customer about our difficulties and ask about any advice. Nonetheless, those User Stories were medium priority in the User Story board, which means it wasn't necessary for us to do them.
 - The User Stories we had to skip were: 8, 9, 10.
- We then put online the local project we held on the computers.
- The rest of the meeting consisted of preparing the project we have so far to show it to the customer on Thursday.

5.15 Date: Wednesday, May 13th

Time: 10:00

Duration: 4 hours

Agenda: Finishing the 2nd sprint, meeting with supervisor

Meeting progress:

- We have internet problems because the school moved our accounts from the old to the new school and therefore taking away our internet access. We had to set aside some time to retrieve Guest logins for the network to solve the problem.
- A bug in the code happened, which temporary removed the functionality of creating members in the database. We have some of the same functionality in "Edit Profile", but something was wrong that we couldn't fix quickly. We had to make a temporary fix which will become permanent if it doesn't fix itself.
- We have contacted the customer and set up a sprint review tomorrow the 14th May.

5.16 Date: Monday, May 18th

Time: 10:00

Duration: 4 hours

Attendees: Salik and Nicolai

Agenda: Fixing errors from Sprint Review, starting sprint 3

Meeting progress:

- We encountered a few errors at the sprint review meeting.
 - Error in the Sign Up function (fixed now)
- Customer asked for extra information on Board members, so you can see the role of the board member.
 - Added it in database, Member model and updated methods
 - Small changes to the design of the GridViews (more spacing, easier to read/navigate).
- Salik has successfully created and edited in a PDF document from server side which is a big step towards our goal in creating invoices.

5.17 Date: Wednesday, May 20th

Time: 10:00

Duration: 4 hours

Attendees: Salik and Nicolai

Agenda: Working on Sprint 3

Meeting progress:

- The team has successfully created the functionality for creating invoice through modifying pdf files with ASP.NET and C#.
 - Problem: Storage problems with Azure Websites. This is something that needs fixing, because we don't want to store unlimited amounts of invoices of the website server but maybe use SharePoint instead.
 - Problem: Not sure how the invoice should look, so we will make an example for product owner which he can give feedback on.
 - Problem: We had some problems with how the price of an invoice should be listed in the database. We decided on going with a 'Decimal' format.
- Next step is to attach invoices to our email functionality in the system and add member info to the invoices automatically.

5.18 Date: Monday, June 1st

Time: 10:00

Duration: 4 hours

Attendees: All team members

Agenda:

Meeting progress:

- We started the meeting from reviewing what we have made so far. We encountered a few problems, but solved them very quickly.
 - There was a problem with Add Member method ???
 - Another problem we encountered was that the login didn't show on a mobile/tablet. We had to rewrite the code so that it detected what device the website was viewed on and if needed redirected to the mobile version of the website. The automatic detection was outside our skill range so we decided to settle on a button that would redirect the user to a mobile version if they clicked it.
- We realized we don't have all the icons we need, so Kasia started making those.
- Salik tried to make the Invoice open in an e-mail window. He managed to view the text but didn't know what to do with a picture we also included in the e-mail.

6 First meeting with costumer - Requirements

LOGON

- MEDLEMMER
- BESTYRELSEN (ADM)

MEDLEMSFUNKTION

- OPRET (ADMIN)
- RET (ADMIN/MEDLEM)
- ANMELDELSER
- GEMME DOKUMENTER (DOKUMENTBIBL.)
- ~~VEDR. MEDLEM~~

OPKRÆVNING

- E. (Abb. fx pr)
- (SEND 1-1)
- Rykne (ved manglende betaling)