Aliaksandr Kureichyk

Design and building process of dynamic web application
Ubuoy Imagine Cup 2013 App

Helsinki Metropolia University of Applied Sciences
Bachelor of Engineering
Media Engineering
Thesis
12 May 2013
The purpose of this thesis work was to study and explore the creation process of web application from the planning stages of it, to the end.

The application was built, designed and styled based on new modern theme of windows 8, also known as “metro theme”. There was also an attempt to match the look and feel of the web application as closely as possible to the desktop operating system and application.

Core of the application was built in C# using Visual Studio and styled with CSS, HTML and JavaScript. The application is very complex with simple functionalities and UI. It is responsive and has a nice look on different desktop and mobile browsers. Mobile application version was built for windows phone based on design of the ASPX.NET application.

The idea of the project was to create an auction-fundraiser site where people could be involved in helping humanitarian organizations by selling their skills (time) and items and donating it to the cause. We hope that this project will go live after the final stages of the project are completed. Hopefully this project will also provide opportunities for people in need to find help in everyday tasks as well as specific professional advice and assistance.

In the project, both, the functionalities and code behind were designed so that the application can expand and new ideas can be implemented and added to current application.

Keywords | ASP.NET, CSS3, HTML5, JavaScript, C#, web application
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Terms and Abbreviations</td>
<td>1</td>
</tr>
<tr>
<td>1 Introduction</td>
<td>3</td>
</tr>
<tr>
<td>2 User Interface</td>
<td>4</td>
</tr>
<tr>
<td>2.1 Usability</td>
<td>4</td>
</tr>
<tr>
<td>2.2 User Interface for Windows 8</td>
<td>5</td>
</tr>
<tr>
<td>2.3 Overview of Windows Store Application Wireframes</td>
<td>7</td>
</tr>
<tr>
<td>2.4 Navigations and Controls of Windows Store Application</td>
<td>9</td>
</tr>
<tr>
<td>2.5 Web User Interface</td>
<td>10</td>
</tr>
<tr>
<td>2.6 ASP.NET Web Form Application</td>
<td>11</td>
</tr>
<tr>
<td>3 Planning Process of the Application</td>
<td>12</td>
</tr>
<tr>
<td>3.1 Getting Started With an Idea</td>
<td>12</td>
</tr>
<tr>
<td>3.2 Developing approach</td>
<td>13</td>
</tr>
<tr>
<td>3.3 Use Cases</td>
<td>14</td>
</tr>
<tr>
<td>4 Design Process of the Application</td>
<td>16</td>
</tr>
<tr>
<td>4.1 Introduction to HTML5</td>
<td>16</td>
</tr>
<tr>
<td>4.2 The Power of CSS</td>
<td>18</td>
</tr>
<tr>
<td>4.3 JavaScript Enhancements</td>
<td>23</td>
</tr>
<tr>
<td>5 Developing Process of Application</td>
<td>27</td>
</tr>
<tr>
<td>5.1 Database and Entity Framework</td>
<td>27</td>
</tr>
<tr>
<td>5.2 Master pages and user controls</td>
<td>28</td>
</tr>
<tr>
<td>5.3 Dynamic element creations and loops</td>
<td>33</td>
</tr>
<tr>
<td>6 Ubuoy Application</td>
<td>39</td>
</tr>
<tr>
<td>6.1 Our Team and Application</td>
<td>39</td>
</tr>
<tr>
<td>6.2 Problems</td>
<td>42</td>
</tr>
<tr>
<td>6.3 Solutions</td>
<td>43</td>
</tr>
<tr>
<td>6.4 Result</td>
<td>44</td>
</tr>
<tr>
<td>7 Conclusion</td>
<td>45</td>
</tr>
<tr>
<td>References</td>
<td>46</td>
</tr>
</tbody>
</table>
Terms and Abbreviations

App bar
Windows 8 control that is hidden by default and appear when user swipe the top or bottom of the screen on touch devices or right-clicks it on PC. [1]

ASP.NET
Development framework for building web sites and web applications with HTML, CSS and JavaScript. It supports three different development models: Web Pages, MVC (Model View Controller) and Web Forms. [2]

Buoy
A float placed in the water, used as a marker or to warn of danger and show navigational channel for water vehicles. Lifebuoy refers to a round object designed to keep a person afloat. [3]

C#
Pronounced “see-sharp”. Microsoft programming language. It is object-oriented language used in web application development on the .NET platform. [4]

Charms
Windows 8 navigation panel used to search, share, change settings and more. Five charms: Search, Share, Start, Devices and Setting provides you with different option depending if you are on the start screen or inside an app. [5]

Cascading Style Sheets (CSS)
CSS used to control the style and layout of the web pages and applications. CSS3 is the latest standard for CSS. [6]

HyperText Markup Language (HTML)
World Wide Web’s markup language used for presenting content on the web. HTML5 is newest and latest standard for HTML. [7]
JavaScript

THE Scripting Language of the web. Use in billions web application for validation forms, adding special effects, communicate with the server and more. [8]

Strengths, Weaknesses, Opportunities, Threats (SWOT)

Is an acronym for Strengths, Weaknesses, Opportunities and Threats. By definition, Strengths (S) and Weaknesses (W) are considered to be internal factors over which you have some measure of control and Opportunities (O) and Threats (T) are considered to be external factors over which you have essentially no control. [9]

Team Foundation Server (TFS)

Collaboration platform at the core of Microsoft solution for Application Lifecycle Management (ALM). [10]

Visual Studio

Is a software development environment (also known as an Integrated Development Environment or IDE), It's used primarily by Software Developers to build Software products, websites and Utilities. Visual Studio was introduced by Microsoft in 1998 and has seen many evolutions. [11]

Windows 8

Is latest version of Windows operating system developed by Microsoft. It is used on personal computers, tablets, and many other digital devices. Windows 8 represents a major departure from previous versions of Windows because it is based on the Metro design language, which facilitates a user interface similar to the ones found in mobile phones. [12]

Windows Store

Windows Store is the online marketplace for Windows 8 and Windows RT users to purchase and download apps. [13]
1 Introduction

Creation of web application has two main stages: planning and building. In this thesis work, both of them as well as the theory behind them will be discussed.

Building a dynamic web application from scratch requires a lot of time. Working in a team of skilled engineers, helps to get the application ready in a shorter period and provides an opportunity to learn from each other. This project’s application was submitted onto Microsoft competition “Imagine Cup 2013”. We had four participants and one mentor. Each of us was assigned a specific task according to our degree specialisation. Our application was built using “Visual Studio 2012” and TFS preview which provided us an opportunity to work on the same Project and change files that were stored on the server. We were able to check-in and check-out changes and compare them if the same file had been worked on by two different persons.

The name of the application is Ubuoy. It consists of three elements: Ubuoy is the acronym of the word you, making the application personal and inviting a user to get involved personally in changing the world and participating in charity. Ubuoy is pronounced /bɒi/ (English UK) or /buː.i/ (English US). [14] Buoy or a lifebuoy refers to a round object designed to keep a person afloat. The idea of the application is to help charitable organisations to fund projects, emergency cases and help needed after disasters, wars and terrorist acts around the world. The last element behind the name Ubuoy is word buy providing the opportunity for people to buy and sell skills (their time) and items and donate directly to selected project of a specific organisation.

The application was coded in C#, ASP.NET, HTML5, CSS and JavaScript. Powerful tools like Skype and Dropbox helped to share our work process remotely. Skype feature Share Screen was used to view another member’s monitor and comment or help on the progress. After that the UI templates and wireframes were created. UI designer often worked together with developers of the code behind creation to clarify the structure of HTML elements and helped in the conversion process of HTML to code behind dynamic creation of those elements.

Next, the two main stages of building a dynamic web application that is responsive, user interactive and designed to look and feel like a Windows 8 desktop application will be introduced.
2 User Interface

2.1 Usability

The purpose of System Usability is to make system to fit minds and bodies of the users in context. [15, 7] In *Multimodal Usability*, authors N. J. Bernsen and L. Dybkjær compare system usability to pair of shoes. [15, 7] It is obvious, that when a person buys a pair of shoes he wants to fulfil two goals: look good and purchase comfortable pair of shoes. The shoe companies will try to make shoes that are usable and would be worn comfortably. If they succeed, they will sell better. Usability has to fit bodies and minds of the user, just as shoes have to fit your feet, but also your style and fashion.

Computer systems belong to fifth generation technology (GT). First GT items were manipulated by human body, like door knobs and shoes. Second GT automated human power used in water and wind mills. Third GT automated human manipulation, for example weaving. Forth GT automated process control in thermostats and cybernetics. Fifth GT automating people more generally, which makes the usability more special. [15, 7]

Usability concept was born out of the need to simplify things for the user. Term “user friendly” became the new goal in the usability engineering. Companies like Apple and Microsoft been working on developing and perfecting operating system with idea of simplicity. [16, 3-4] Without usability engineering and considering the needs of end-user, the system may not do what user wants it to do and it will affect the user’s ability to use the system. Most of the time user will adopt to a difficult system if he has to use it, but if he has no need, he will stop using it for good. [16, 13] Developer’s job is to create a system that would be exactly what the user wants and needs, and this job is not easy. Gathering information about the user and using already-done research will help us to create a system which is not only easy for user to use but will also be what he expected.

At the moment web site usability development is perfected every year with new standards and solutions. Developers are quite familiar with usability optimization for a desktop versions. To present same result on other devices developers choose either own, mobile version of the site or develop a desktop version that is responsive and adopts
the style and content of the site to any resolution or device. Both ways are effective and have their own advantages.

2.2 User Interface for Windows 8

User interface is the most important part of the system or application. UI creates the connection between human and computer. When user interacts with an application, he only sees the UI in a form of icons, buttons and text. User does not see the code behind nor does he see the login of the application. Those lines of long code, methods and logic is hidden from the user. Imagine if the UI controls were not graphically presented to the user but instead would have the code of it in the programming language it is written. We would have UI that is too complicated to use. Since we designed our application to look like Windows 8, we can see how a button in the app bar is presented to the user (Figure 1) and in the code behind (Figure 2). [17]

Figure 1. The graphical look of the Delete button in the Windows 8 app bar.

Figure 2. The Code of the Delete button in the Windows 8 app bar.

The code of the button has so much more than just a look. Button’s look should always speak to the user of the main function for it. The example above clearly shows an icon and a label, which makes it more clear for the user to know what the button does in the application. Coder sets the icon and the label for the button along with the functions.
The only thing the user has to do is to learn what the button does (usually application have a manual) and use it depending on his need. Some usability functions are hidden from the user, just so an unexperienced user would not delete something by accident. Many times user gets a pop-up dialog to confirm an action he is performing.

Store applications on Windows 8 have brought a brand new look to applications. Applications can be downloaded and installed from one location *Windows Store* and the core structure and look differ only based on template and wireframe. There are ways to customize the default icons, but usually designers and developers change only the name and the labels for the elements. This makes UI of the app the same and the user knows the location of the app bar, charms or navigation menu.

![Figure 3. Screen shot of Windows 8 Professional Start menu.](image)

Windows 8 operation system has a grid styled tiles to navigate to your application. They are more than just shortcuts to launch your applications. The tiles can have live feeds from an application and most important summary info from the app (Figure 3). For example, news headlines and the picture can change, showing you the highlights of the new event. The user can turn on or off live tiles feature of the application. If the option is not available, that only means that the app developers have not implemented it in the application.
2.3 Overview of Windows Store Application Wireframes

All Windows 8 applications are designed in a similar way. Basic application has a grid system with three-dimensional matrix of group, group detail page and item page. Developers code the content they wish to display in the grid (Figure 4).

![Grid App](image)

Figure 4. Windows 8 group view of grid app template. [18]

The grid system makes it easy to populate content of the items, news, projects in the same format and group them as needed, for example in categories.

The grid view design gives a nice and clean look to the application. Figure 4 shows a basic grid view of the groups. The basic layout usually has a header for the group with a link-like function, tiles with the title and maybe subtitle of the item and the image as tile background. This template can be styled and modified to fit the colour and style of the theme depending on the design.

Group details page usually displays one item in a bigger format. Larger image and a more detailed description of the item and list-view print for the rest of the items. [19, 129-130]
When creating a Windows Store app in Visual Studio, these templates are available by default and most of the code is generated by the system.
2.4 Navigations and Controls of Windows Store Application

For more than 20 years people had to use basic navigation in windows applications. Menu items like File, Edit, View and About were default Microsoft navigation and function controls using typical dropdown list control.

In Windows 8, Microsoft introduces Application bars. The top bar is used for the page navigation, using thumbs and text. It uses larger icons and text and helps to navigate between different locations of the app. The bottom bar then uses command type icons, like filter, add and delete. The weather application on Windows 8 is a good example when both bars are used. See Figure 7

![Figure 7. Screen shot Windows 8 weather app.](image)

Windows Store applications also use charms for special navigation of options and settings. Main charms are Start, Search and Settings. Start will always take you to your computer start menu where the app tiles are found. Search is multi-purpose. If you use search option while you are running Windows Store app, by default you filter and search the app’s content. You can change the area of search to your folder, files or previously made search areas without leaving the search charm. Settings charm controls the settings of the app if you opened it while using an app, or the settings of your PC if you opened it from the start menu. Share and Devices charms are not used as
often as the other three, but they function the same way. If you open Share charm from start PC will state that there is nothing to share and you should run an application to do so. The Devices charm on the other hand will try to list you available devices you can add and use with the app or with the PC. For PC you can use an external monitor as a second screen. Figure 8 illustrates the look and location of charms on Windows 8.

![List View & Charms](image)

Figure 8. Windows 8 Charms. [18]

2.5 Web User Interface

Web UI has changed a lot in the past 20 years. Site owners try to personalize user experience on their site or web application. With years of experience and research developers and designers are taking into consideration user’s ability and what is practical while keeping the style of design look good and eye-pleasing.

Nowadays the most important thing for developers is to design a site where content can be easily updated or dynamically changed. These sites always use some other languages in the combination with HTML and CSS. Very popular platforms and programming languages for dynamic sites are Java, PHP and .NET. Most of the platforms use some kind of database to store site’s content and to help the update or upgrade process.
2.6 ASP.NET Web Form Application

Microsoft ASP.NET is a powerful, modern technology for web application development. Web Forms in ASP.NET is UI elements that connect user to the application and give the look of it on the front-end. Web Forms have similar capabilities as Windows Forms. They also have properties, methods and events for controls that are placed inside of them. These UI controls render in a specified markup language, e.g. HTML. In Visual Studio you can drag-and-drop these controls from the toolbox and style them as needed.

Web Forms are built with two parts: the visual part (ASPX file) and the code behind file, where class, methods and element’s logic is defined. There are two types of controls used to create UI in ASP.NET Framework: Web Form controls and HTML controls. HTML controls resemble the actual HTML elements that you could use on any .html page. All tags and attributes apply for the HTML controls. Any HTML element can be set to run as HTML control by adding attribute (runat="server") to the HTML tag.

\(<input type="text" id=txtFirstName size=25 runat=server>\)

Web Form Controls are created and run on the server like HTML controls and get rendered by the system to appropriate HTML. For example, a Panel, would get rendered as div tag. [20] Figure 9 illustrates Web forms as part of ASP.NET.

![Figure 9. Web Forms as part of ASP.NET. [20]](image-url)
3 Planning Process of the Application

3.1 Getting Started With an Idea

Most likely the debate about what defines the quality of the design in the web site will never stop. Some are more confident that stylish and classy graphics are an important part for a site to function. Others think that most important thing is that the site is easy in use. For others, information presented on the site is the most important attribute. In reality, none of those attributes can have a priority number. A site with great graphics but poor functionality and information is as useless as the site with amazing content but hard to read and poor design.

Planning the site should always answer most important question: “What is the purpose of it?” When you know the purpose of the site, you can answer many other practical questions concerning graphical look or structure. Theme of the site gives clear definition of sites content, but purpose of the site will reveal its functional capabilities. [21]

Every product on the web has strengths, weaknesses, opportunities and threats. Listing and addressing these attributes in the very beginning of the project will help to build a good foundation for the site. This process is also known as SWOT analysis. Information collected using SWOT analysis is very important in planning process; it shows the possibilities and reasons of why the idea might fail or succeed. [22, 38]

When the purpose and capabilities of future product are clearly defined, developer or developers start planning the look and functionalities of the site. Usually first sketches are either hand-drawn rough representations on a piece of paper or board, or made with some modern design software with drag-and-drop capabilities. The sketches later serve as the base for coded wireframes, which are previewed on the browser.

Next step in planning is to create a site map. Site map at this stage would be just a list of pages or states the application would have. For example Login, Register, Home page and so on. While creating the map, you will be able to group the pages to different categories and plan the way they will be connected. It is like building a real map in virtual reality.
3.2 Developing approach

Developing approach is a state when you select tools, material and workers for the task. Over the years, developers formed different approaches to build the desired product. Although there are several types available, the best known approaches are Prototyping, Spiral and Waterfall. They all have advantages and disadvantages.

Prototyping is development approach where a small-scale sample is created to give the user a possibility to understand what the final product is capable of and how it will be used. User takes part in the development process. While most prototypes are just mock-ups and get dismissed, some of them have an ability to make it to a working product at the end. This approach leads to fast development of incomplete, but functional applications. It can also give false expectations to the user. Looking on the prototype user might think the application is nearly done, but in reality it is hardly functioning. [23, 3-5]

Waterfall approach to the project provides you sequential phases that can overlap each other, giving room for changes and fixes while the phases overlap. Waterfall focuses on timeframe, budget and implementations at the same time. This is commonly used by less experienced development teams and you can see the progress of the development at different phases. It is a practical approach for those who need to create documentations and reports during the project. [23, 1-3]

Using Spiral approach can be a great advantage for experienced developers as well as unexperienced ones. While it is customized to one project, cannot be reused and is very complex, it gives an opportunity to get a working application with ability to build on it and expand. Each cycle is considered a path travelled around the spiral with progress of development increasing. All though each round repeats four basic steps, it does not begin at a new phase but continues the progress from the previous round. The four basic steps in the round are planning, analysis, evaluation and development. (See Figure 10) This approach breaks the project into smaller segments and lets developers focus on one part of the application at the time. This approach may inherit the ways of the other two approaches for special cases. [23, 6-7]
When an application is complex, a team of expert is required. From the very beginning the team must be up to date and have the latest news or changes on the project. It is important to choose correct tools for development process. Chosen tools will determine the cost and speed of the progress.

Visual Studio 2012 has everything you need to build ASP.NET application when it comes to coding, but you have to consider also tools that will help you to create icons, logos, backgrounds, keeping documentation, file sharing and file storing. It is important to have more than one copy of the project in more than one location at the time. Personally, I recommend to label archived project folders with the date. Make a copy and archive the project folder to another location if working with files using systems like TFS, Dropbox, SkyDrive, where overwriting old files by mistake is so easy.

3.3 Use Cases

Use cases have been very important part of software development for many years. In the planning stages it is very important to picture the system and plan the steps. The
team must be able to visualize possible steps or actions to achieve a certain task in the application or make changes and start planning again.

A use case defines a sequence of actions a system performs that yields an observable result of value to a particular actor. [24]

In other words, use case shows functional and non-functional steps of an actor achieving the goal. Every part of system usage can be broken down to system use cases and planned carefully. In a complex web application planning process most important use cases should take priority. Registration, Login, Logout, Settings are just a few important use case processes in an application. Use cases can also change depending on user role and access level to the application. When planning a system with different roles and access levels, it is important to create a use case for every role. The best way is to start with low-level access roles and move up the ladder to admin. That way you can reuse the use case created, by just adding to them and expanding them.

One thing important to remember is that use case should not be broken down like a problem into smaller parts. You have an actor and you have a goal; you must complete all the steps in one use case diagram. Even in complex web systems a use case always reaches its goal or fails in the attempt.
4 Design Process of the Application

4.1 Introduction to HTML5

HTML is the language of the browsers. HTML is an acronym for HyperText Markup Language. [25, 3] All the ASP.NET elements render into HTML elements by the system. That is why the best way for us to preview our demo pages and templates is to actually write HTML code and then convert it into appropriate web form elements. It might look like double work for some and some people might find it faster to do web form element right away. But if the HTML coder and .NET coder are two different people, it would be the way to go.

HTML has been around for more than 20 years and during this period the coding approaches have changed, but the core structure and tag creation of the language remained the same. At this moment HTML has reached fifth generation and is referred to as HTML5. Some of the new, commonly used elements that this version brought are header, footer, nav, canvas, audio and video.

Video support is a big difference between this version and HTML4. For example if you wanted to play the video on mobile device in the past you had to have Adobe Flash or other third-party plugin. [25, 7] With HTML5 this issue is solved by nesting the video file in the video tag. Canvas support 2D and 3D graphics. Footer and header are specified elements to control and hold those sections only. It helps to separate the code of the page into three or more main parts: header, middle (content) and footer of the page.

At this point, HTML5 is still work in progress, but with the changing speed of World Wide Web, it just has to catch up. [25, 6] Many browsers already support new features and elements of this simplified version of HTML, but some are still a bit behind. The main disadvantage of this version at the moment is the versions of the browser. If user’s browsers are not up-to-date, new elements and tags might not render properly and will most likely suggest user to upgrade or change the browser. Computers or devices with outdated browser will not live forever. Eventually all devices will support this version.

HTML page consist of elements. Each element has either a self-closing tag or opening and closing tags. It is basically a series of containers. When the document type has been defined there is large container html which is defined by opening and closing tags
This container holds two main structural elements head and body. Figure below shows the basic HTML page code. [25, 10]

![Figure 11. Basic HTML page structure](image)

HTML5 simplified declaration of the document type and now document can be declared with one simple line: `<!DOCTYPE html>`, but it always has to be declared before the html tag. Main container of any HTML page is html. All content that browsers render in the web page must be within opening and closing tags of this element. This element is also referred to as root element of the web page. [25, 11]

As noted earlier the head and body are the two primary elements in the page. The content of the head tag is not visible to the user, except for the <title>. The content included in the head tag contains metadata and information about the page, provided to search engines and external agents, as well as code of JavaScript functions or cascading style sheet code. The content inside the body tag is all the visible parts of the page that is rendered and displayed by the browsers. The tag can include various attributes like ID or class and can be styled using those attributes by CSS, at which we will take closer look later in this section. Some other common attributes used in the tag could be lang, which defines main language of the page or onload which loads JavaScript functions when the page has been loaded. The basics of JavaScript are analysed on the following pages.

The advantage of HTML language is that you do not need to be a good programmer to write it. It isn’t that hard, in HTML5 element names define exact purpose of the ele-
ment. A **Head** element represents the head of the document, a **header** holds the header content, a **footer** holds footer content and a **p** element represents the paragraph. In practice, it is probably the easiest language in the world. Anyone can write it and no special software or text editor to do so is needed. It can be done with just a notepad. Although many claim that they can write HTML code just by using notepad, very few actually do so. We have to understand that 20 years is a long time and the developers and the programmers wanted to make their work load lighter. Software like Dreamweaver and Komodo let you code html tags a lot faster and easier. Also it helps to spot mistyping and any unclosed tags.

4.2 The Power of CSS

CSS stands for Cascading Style Sheets. It is a powerful piece of technology that adds the look and feel to the representation of your HTML page. [25, 21] It lets you change text font, colour, and position as well as add animations and shadows to give 3D feel to your text, images or any HTML elements. You can apply changes to the entire site, a single page, group of elements or to a single element, depending on where and how you use CSS rules. Every site on World Wide Web has CSS implemented into HTML code either externally, internally or inline.

In the past pages were styled with HTML attributes or nested tags. Figures below show the difference.

Figure 12. HTML styling.
At first this looks a bit more complicated, but CSS rule can be applied to entire site or single element, where styling using HTML tags and attributes ties the style to a specific element. In case you do need to apply it to a single element, you can do so by using ID and class attributes or inline use of CSS style. This provides you with benefits to control elements on the global scale of the site and change them quicker, instead of changing every single line. Styling with CSS gives you:

- **Ease of modification**: CSS can style all tags or a selected tag in one place by applying one or many rules to it.

- **Advanced design options**: Latest version of CSS (CSS3) provides us with the ability to style a page with pixel-perfect location positioning, rounding of corners, animations, changing style on when you place your mouse over and much more features.

- **Media targeting**: Content presented in the World Wide Web is viewed on multiple types of devices, with different screen resolutions. CSS lets you target those devices and change the look of the content depending if it is viewed on a computer monitor, a phone, a tablet or a TV. [25, 21-22]

All the tags in the pages that are styled using CSS use one or more rules. J. Lowery and M. Fletcher in the *HTML5 24-Hour Trainer* describe that each rule has three parts: the selector, the property and the value. [25, 22] Figure 14 shows that *h1* is the selector, *color* is the property and value is red.
Each rule can have one or multiple declarations. Rules are separated by opening and closing curly brackets and declarations are separated using semicolons. It is also possible to have more than one value for a property in a single declaration. For example,

![Figure 15. Using multiple values with one property.](image)

You can also have group multiple selectors and apply one rule, just by adding the selectors and separating them with a comma. [25, 22] This significantly shortens the code size of your CSS file and the browser will read the rules faster, optimizing the load speed of your pages.

![Figure 16. Using multiple selectors with one rule.](image)
CSS styling is essential in the modern web development. By equipping yourself with the basic knowledge of the CSS technology, you are able to bring “colour” and “style” to your design. Every modern web page developer uses CSS to bring better user experience to the page visitors.

When browsers process style sheets, they follow three principles that apply to declarations and the rules written in the document. CSS code can be placed in one of the three places: external file, in the `<head>` tag of HTML or inline, effecting only that tag. When the conflicted rules located in different locations, declared style inline will be applied, no matter of the declarations made in the other two locations. If it is embedded in the `head` tag, it will overwrite the style declaration written in the external file. Only when the selectors have no styles applied to them inline or embedded will the rules in the external file apply. [25, 23] No matter where the style rules are located the principles will work the same way. These principles are:

- **Cascading**: When two or more identical rules defined with different declarations, the one declared last will be applied. An example is shown below.

![Cascading example in the same location.](image)

- **Inheritance**: When a rule declared to a tag that has other tags nested inside, the declarations made to the dag will apply to the nested tags unless it is specified otherwise. In other words all nested tags will inherit style declarations made to their parent tag. The example below demonstrates that declaration made to body tag will affect other nested element in the page, until those tags are given another declaration.
Specificity: CSS rules can be applied to custom selectors, not just tags. By adding ID or class attributes to the tags we can access and style more specific elements of the page. This principle commonly used when we want to target a specific element or a group of elements. Selector hierarchy of specificity has 4 levels:

- **Inline**: most specific
- **ID**: specifies an element with that ID attribute
- **Class**: specifies elements, which are classified with that attribute
- **Tag**: specifies which tag will get style from declared rule. [25, 24]

Authors, J. Lowery and M. Fletcher demonstrate how the specificity works in practice. Figure 19, shows an HTML code where ID and class are specified [25, 24]. And Figure 20 shows CSS declaration of rule to h1 tag.
But say, you wish to have all \textit{h1} elements green but the once that have \textit{mainTopic} class attributes to be red. All you would have to do is add another rule targeting those tags that have that class. [25, 24] See Figure 21.

If for some reason we need to have particular tag to be another colour, it is easily achieved by adding the CSS rule into HTML code using inline method. [25, 24]

Inline styling is not very welcomed by web developers, just because they consume more time to modify. I found one particular case when they are used more often is in HTML e-mails. Because rendering external or embedded style method is impossible.

4.3 JavaScript Enhancements

What is JavaScript? It is a scripting language that enhances web applications with personalized and interactive content. Enables us to create flashy drop down menus, animated text and images, and changing content or style of the site. JavaScript is sup-
ported by all the major browsers and it is considered to be solution for many site development problems. [26, 1]

JavaScript it interpreted language. It means that it needs something to interpret the code to machine, that understands zeros and ones. When browser “scans” the JavaScript code it uses program that serves as interpreter. This process happens at the time code is executed and it is repeated every time the code is executed, where compiled language has to be converted only once. Compiled languages are Visual Studio and C++. [26, 8] JavaScript is nothing like Java, the only thing that they share is that they are both have Java in the name and they are programming languages.

To create and task JavaScript code you need to have a browser that is equipped with a JavaScript interpreter and a text editor. First browser that had the interpreter was Netscape navigator 2 and was called LiveScript, but because Java was getting popular at the time it was changed to JavaScript [26, 9]

JavaScript is commonly used to interact with user, gather information and validate their activities. Form validation is major use of this technology. Many times we need to check if the user has actually filled in the information in the form field or agreed to the terms that is where JavaScript comes into play. JavaScript is also used for different changes to the page’s content depending on the mouse action or a specified event. Animations, image manipulation, changing and adding CSS code as well as modifying HTML code itself is all within the power of the JavaScript. Advanced JavaScript usage can be found in various web application where small JavaScript applications are at work. Calculators, design tools, games and many more.

To be able to use JavaScript code in a web application you have to either use starting and ending tags `<script></script>` and place the script within these tags or you have to define the location of the script file with extension .js as an attribute of the `script` tag like this: `<script src="myScript.js" />`. [10, 12] This way we include the external script file and browser will run the script from that location. When you include the file you can place the tag within the `<head>` tag or `<body>` tag. Many developers include CSS files in the head before the HTML body is loaded and are faster visible to user and place the JavaScript files at the end, because the manipulations and interactivity with the user happens after the page has been loaded.
A simple script added to the page can look like illustrated in the following figure.

Looking at the code we see that we made one statement in the JavaScript code telling the browser to display our document (html page) with red background. Every line of code made within the `<script>` tag or in an external `.js` file is considered a statement. The code will overwrite the existing style made in the HTML tag. Every statement in JavaScript ends with a semicolon. A statement should fit into one line and not continued onto the next line without the semicolon. There are also cases where you must insert a semicolon to make a statement.

JavaScript handles different types of data. Some of the data would be understandable and visible to us like text and numbers. Other data type would be somewhat abstract, like object data. When we deal with different types of data in JavaScript, we often do not have to specify the type of data used. JavaScript will recognize and defines it for itself, making JavaScript more “forgiving” language.

Data can be stored either permanently or temporarily using variables. Usually stored data for long period of time would be placed in a database, a collection of tables and rows on a local or remote server. When we wish to store data temporarily we use variables. Variables can be of different types just as data.

Declaring variable and giving a name done in a simple JavaScript statement:

```javascript
var myVar;
```
Placing data inside of the variable would be something like this:

\[ myVar = "myText"; \]

As you see in the statement above the text “myText” is assigned to the variable that we created in the previous statement. Variable values can be accessed, changed and inherited with more statements. For example, we can create another variable and inherit the data inside our firstly created variable with the following statements:

\[
\begin{align*}
    \text{var myVar2;} \\
    \text{myVar2 = myVar;}
\end{align*}
\]

This would place the same data inside our second variable. As mentioned above JavaScript defines the type of data by the value placed in the variable. You can manually define the type using statements as well. There are five main types of variables, and they are defined by placing word `new` and the `type name`:

\[
\begin{align*}
    \text{var carname = new String;} \\
    \text{var x = new Number;} \\
    \text{var y = new Boolean;} \\
    \text{var cars = new Array;} \\
    \text{var person = new Object; \[27\]}
\end{align*}
\]

Variable values can be `undefined`, `defined` or `null`. It is possible to convert the data inside of a variable into another type. [27] Variables are a way of storing the data to be used in your code whenever needed. Names of the variable may not include special character like `&` and `%` or use reserved words like `with` or `for`.

JavaScript is a language of endless possibilities and creativity. It help us to keep our websites secure, interactive and up-to-date. Understanding even the basics of JavaScript will enable us to bring new feature to our applications and give our visitors a reason to visit our pages again.
5 Developing Process of Application

5.1 Database and Entity Framework

Database is simply a collection of data arranged into tables and rows, so it can be accessed, managed, and update whenever needed. Most common type of database is the relational database. It is the type that developers use when building dynamic pages or web applications. Relational, because of the relations between different tables that database have. [28, 422]

When we use databases in a dynamic web application, we do quite a lot of manipulations and updating. Four main type of operations used are grouped under CRUD acronym, letters stand for Create, Read, Update and Delete. [28, 428] Sometimes necessary to repeat these operations several times for several tables, rows or even databases, that is when Entity Framework come in handy.

EF is an Object Relational Mapper (ORM), widely used for development of applications where data-oriented functions and features are present. [28, 500] It enables us to take database objects and use them as .NET objects and access them using code. EF can do reverse designing. For example, we can create an object model first and let EF assist us in creating the database structure needed. Using EF is simple. Diagram designer has drag and drop option available for us. When we drag and drop a table from the database into the model, it becomes available to us as a .NET object. If we drag and drop more than one table related to each other, designer will spot the relationships between those tables and add that to the object model. [28, 500] It does most of the work for us giving us access to all the properties of the database as a result.

EF has four development workflows and they depend on two main questions: “New or existing database targeting?” and “Using designer or coding?” We have to answer these questions before we can select right workflow.

When we answer these questions we get following four workflows to choose from:

I want to write code…

✓ I’m creating a new database:
Use **Code First** to define the model, classing and mapping in your code and let EF generate you the database.

- **I need to access already existing database:**
  Use **Code First** to create the model. Define classes and map to existing database. Reverse engineer tools are available.

*I want to use designer...*

- **I'm creating a new database:**
  Use **Model First** to create boxes and lines and then let EF generate you the database.

- **I need to access already existing database:**
  Use **Database First** to create boxes and lines model that EF will map to an existing database. [29]

5.2 Master pages and user controls

Every year and every update software developed to minimize the time coder spend on design layout, fonts and colours of application, so they can focus on the writing of the code instead. Many organizations tend to change the layout and design of their application regularly. Microsoft separated those elements from the code, so the layout and styling changes could be easily applied. ASP.NET offers developers an option of using master pages. [30, 41]

Master pages separate common elements, like logo navigation and header content from the individual content pages. Using master page will give your pages consistency and will significantly cut your work load when changing content style over entire site.

For example, your design of site is made in a way that your logo, navigation menu and footer content does not change on the entire site. Every page has same elements and they are positioned in the same place. With master page we can create common structure for all of the pages in the web site. [30, 43] Then, whenever you need to update a portion of the content, like a footer or add a navigation element, it is enough to change it in one place and we will get it changed on the entire site.
Master pages are defined with file extension .master and structured in a similar way as any other ASP.NET page. They can contain Script, HTML, CSS, they have code-behind files. The only difference is the class they inherit. Master pages inherit MasterPage class, where regular pages inherit Page class. The following code shows a simple definition of the master page we used in our project.

```csharp
<% Master Language="C#" AutoEventWireup="true" 
    CodeBehind=\"uBuoyMaster.master.cs\" 
    Inherits=\"uBuoy.MasterAuthentication.uBuoyMaster\" %>
```

Figure 23. Master page declaration

After defining the master page and its location we can control the style sheets, navigation controls and footer of the site by coding it into this file. We can control the title and head tags of the pages in the code-behind if needed. Content of the pages then placed using placeholder tag named ContentPlaceHolder. [30, 46] You can have one or multiple placeholders in the page. You can also have one or many master pages.

Creating a content page using our master page is simple. We use add item dialog window and select the master page we want to apply. Content page provides us with mark-up for the ContentPlaceHolder that was created and defined in the master page. All we left to do is to place the content we want to display on the page within this placeholder tags. The code below shows how we can place a placeholder within HTML tags.

```html
<div class=\"page-region\">
    <asp:ContentPlaceHolder ID=\"PageRegionContent\" runat=\"server\">
    </asp:ContentPlaceHolder>
</div>
```

Figure 24. Placeholder container for the page content.
Another advantage in using master pages is by changing the master page we can easily change the header of footer of the page depending on our needs. One of the practical examples on using master page was when we needed to create different header and navigation depending on if the user were logged in or not. In our project we use sessions to store the user id and login information. We also used this session id to define which master page content to display. In the content page the path of master page used is defined like this:

```
```

In the code-behind then we can dynamically change the path of the master page file, thus using another master page with a different content when user is logged in. Code-behind would look something like this:

```
void Page_PreInit(object sender, EventArgs e)
{
    if (Session["LoggedIn"] !- null)
    {
    }
    else
    {
        MasterPageFile = "~/uBuoyMasterNA.Master";
    
    
```

Figure 25. Selecting the path of Master Page from code-behind.
Master pages provide us with ability creating consistent content pages throughout the whole application. Using master pages templates help us to modify more static elements of the page such as header, navigation and footer and removes the need of including long list of CSS and JavaScript files to every single content page. [30, 58]

With ASP.NET we can create two primary types of custom web controls: user controls and server controls. Using user controls we group existing controls together to keep code cleaner and separating elements inside a single site. Server controls are ASP.NET controls using .NET Framework and take advantage of design-time support in the IDE (Integrated Development Environment). [30, 329]

User controls are created in a file with .ascx extension and is similar to the ASPX file and can also have its own code-behind. User controls are normally reserved for reuse within particular site. If for some reason the need occurs to reuse control on another site, creating web server control should be the way to proceed.

User interface of the site often contain common functionalities with similar sets of controls. User controls provides a way to combine those controls into a single unit of functionality. [30, 331] They can be dragged and dropped onto multiple pages within the site when UI is created. This is very useful when creating forms or widgets. Figure 27 show how an avatar user control is defined and coded.

Figure 27. User control example.
It is good practice to place all the styling outside of the user controls, for example into a CSS file. This will allow control created to inherit and receive styling from the theme or a page on which it has been used.

With user controls we can create a single form template with several user controls in it. We would then call a specific user control depending on the form id passed in the session. To add several user controls to the page is fairly simple, we just drag and drop the user control needed and Visual Studio will place the needed syntax to our page. Figure 28 displays one of multiple ways how we can use several user controls on a single template and choose which one we want to display or hide using function in the code behind. This shortens our code significantly, separates the controls from our page and gives us ability to manipulate and alter the controls in one place.

Figure 28. Multiple user controls on one ASPX page.

In the user control page we set visibility of the controls to false by default and in the code-behind we can then choose which of these user controls from lines 11-14 we want to display, by changing the value to true. Figure 29 shows one part of code behind using controls inside a “switch” formatted code where for each case we can manipulate and edit the control, page title and more.
It is possible to dynamically load a user control by using `LoadControl` method inherited from `Page` class, which can be useful in situations where variable number of instances added to the page. [30, 353]

5.3 Dynamic element creations and loops

In web development dynamic web site means that the content and elements are dynamically created and updated. Dynamic data in ASP.NET lets you create application by inferring data at run time entities from database and deriving UI behaviour from it. [31]

Dynamic data supports scaffolding, which creates opportunity for developers to generate web pages for every table in the database. Scaffolding elements are easily customized. It is also possible to create new elements that will override the default ones. Dynamic data gets information from the model, which represent entities as CLR (Common Language Runtime) types. [31]

Architecture of dynamic data has three layers:

- **Presentation layer**, contains elements created to display UI of data entities. There are four common elements: `Page Templates` (pages that render data from any database table), `Entity Templates` (used to customize UI for the whole
data entity, for example table or row), *Field Templates* (controls that render individual data fields), *Filter Templates* (controls used for data filtering, enables user filter and select specific data based on column value). We can also create *custom templates* in the presentation layer.

- Data layer, supports data models, such as LINQ to SQL and ADO.NET Entity Framework, enables user to use these types to query data and perform the CRUD operations mentioned in Entity Framework section.

- Data source mapping layer, used to generate data models, such as LINQ to SQL and ADO.NET Entity Framework. [31]

![Dynamic Data architecture](image)

**Figure 30. Dynamic Data architecture** [31]

Dynamic data supports three options for design:

- Use **Scaffolding**. This option will enable you to generate Dynamic data web application with very little or no coding at all. The pages will be generated by ASP.NET Dynamic data and provide you with CRUD capability for each database table. [32]

- Use **Data-bound controls**. Enables you to use ASP.NET controls like GridView and ListView to populate dynamic data and add them to an existing web site. Has several feature and advantages like run time filed validation, pages with CRUD operators, and customization of data fields. [33]
Add **Dynamic data** to an existing or new ASP.NET web site. This option enables you to add scaffolding elements and dynamic data to an ASP.NET web sites. You cannot enable dynamic data for data-bound controls here, but import elements. This type of approach gives you control over dynamic individual controls. [34]

Above options and operations gives you several options to proceed when creating dynamic web site. This option are great when you want to use build-in features and controls and you don’t want to or know little coding.

When creating an application that requires specific controls and HTML elements and do not wish or cannot use **tables** or **view** controls provided by Dynamic Data, we can create own classes and generate creation of ASP.NET elements in code behind using loops and dynamic creation of controls.

Every ASP.NET controller can be created and accessed from code-behind. To create and add new control we must have a container where this control is placed. [35] If container does not exist and we do not wish to create an extra container that will render as an HTML element, we can create **PlaceHolder** and use it as a container to nest needed elements. This will also enable us to dynamically create or change any attribute of the element, like colour, dimensions or text value. Here is an example of creating an ASP.NET control in the code behind:

![Figure 30. Creation of “Label” control in the code-behind.](image)

We then can nest this element inside a PlaceHolder or another existing control. See figure 31. This is useful when we have an x number of elements we want to create.
Widely known practice in dynamic web development is using loops. Loops are used in most programming languages. It is a way to dynamically create array or data and display it to the user. User never sees this creating process and browser source code will display elements created, not the process itself. When we program using C# language in ASP.NET we have four types of loops available to us:

- **While** loop. This loop uses expression and statement to generate result. In the expression we set condition for the loop termination. The statement states the action to be executed until the condition defined in the expression part gets false value. If it does not get a false value, the loop will be infinite. [36]

- **Do something while** loop. Unlike the while loop, this loop will be executed first time even if the condition is false. But the second time the condition must be true for loop to work. This loop is suited for tasks where the action must be done at least once no matter the conditions. [37]
✓ **For** loop. This loops is used when we know how many times we want to repeat the action in the statement of the loop or when we wish to limit the actions to a specific number of times and stop the loop at the specific interval. [38]

✓ **Foreach, in** loop. Foreach is a new type of loop in C#. It is similar to the *for loop*, but this one is used more when working with arrays or collections. For example if we would use *for loop* when we wish to loop an array we would have to

![Figure 33. Do...while loop example.](image)

![Figure 34. For loop example.](image)

![Figure 34. For loop example with set interval](image)
specify the size of array. With *foreach loop* the size is calculated automatically when we specify the object for the loop. [39]

Which loop to use when will always depend on the structure of the application and the result we want to achieve. Creating dynamic web application almost always use custom generated loops in the code section or system generated loops. As always, “custom” means custom created and customizable, giving you more control over the created element or elements.
6 Ubuoy Application

6.1 Our Team and Application

Every year Microsoft host an international competition for students. Universities from all over the world send teams to participate in the Image Cup. This year under supervision of Principal Lecturer, Kari Aaltonen, a team of five students and a mentor was formed to represent the school in the competition. The team started to meet in October, 2012 and worked together for the next 7 months.

Everyone in the team worked very hard to contribute with their knowledge and skills. In a team, the work of everyone is valuable and we can learn and teach each other something. Our team was no different. Below the members of our quite international team are introduced.

Dinesh, Sapkota (Helsinki University of Applied Sciences). Dinesh is from Nepal and was responsible for life-cycle and back-end structure of the application. He coded most of the functions in code-behind, worked with model, Entity Framework and classes for the dynamic data creation. His specialty is C# and ASP.NET programming.

Mahletkifle, Adenew (Haaga-Helia University of Applied Sciences). Mahlet is from Ethiopia. She was responsible for back-end coding, structural design of application, forms, login and registration.

Maimuna, Syed (Helsinki University of Applied Sciences). Maimuna originally from Saudi Arabia. She is an experienced Windows Phone application developer so the Windows Phone version of our web was delegated to her. She was also significantly involved in the app trailer and overview videos as well as presentation and documentation.

Aliaksandr, Kureichyk (Helsinki University of Applied Sciences). Born in the former Soviet Union, Republic of Belarus. Has several years of experience in web design and development. Main responsibly was UI, graphical representation of the web (logo, style, and layout). Speciality: CSS and HTML.

Eskindir, Abdela (graduate of the Helsinki University of Applied Sciences). Originally from Ethiopia. Experienced coder in .NET Framework and Microsoft applications. In
2010 His team participated in Imagine Cup competition and came second in the World Citizenship category. Eskindir was a mentor for the team during the project and helped us with practical issues about the competition and helped us to structure the foundation for the application.

As mentioned in the introduction and the abstract, our application is a charity based service provided via Microsoft tools and systems. Demo of the application was developed for the web and requires a browser and internet access. It supports major browsers on various devices. Windows Phone app also available providing more features like geolocation, toast notification, push notification and use of build in application like calendar and mail client. Applications main purpose is to promote donations to charities and let people get personally involved in worldwide projects.

We wanted to try something new and adapt the new modern look and feel of Windows 8, and used that as base for our layout and design. Using Metro UI CSS framework we designed and styled the application and were able to achieve the look and functions similar to Windows 8.

Metro UI CSS is made with LESS. It makes developing systems-based CSS faster, easier, and more fun. –Sergey Pimenov, Metro UI CSS.

LESS using variables, mixins, operations and functions to extend CSS and make it dynamic. It is a dynamic language for stylesheets. It can run on client-side user all major browsers or the server-side. [40] Other features like app bar and charms navigation were created using JavaScript. This is how the index page looks in the demo of our application:
Index page is the first page users see when they visit our URL address. We wanted to have a latest feeds from four basic modules available to user. News feed, Projects, Skills, and Tasks sections on the front page dynamically generated elements populated from the database and displayed according to their creation date.

If user is not log in he has only three out of five charms available and must register or login to access additional modules and secured pages of the application. Here is how our login and registration forms are presented to the user:
Logged in user will have access to multiple features and will be able to interact with the application. Once logged in he will be able to follow projects, organisations and other users, create tasks and skills to his profile, edit his own information and customise the look of the application, changing background, colours and fonts.

The tiles are animated and can be rearranged using drag-and-drop. They are responsive and to any screen resolution and rearranged depending on the width and height of the screen. All the pages of the applications have only horizontal scroll and the width of the page is dynamically updated depending on the amount of tiles and tile groups generated by the code.

The application is well formatted and organised. Layers and UI are separated and grouped. Code-behind’s functions and elements are commented for future development of this project. We have a short manual and overview video and a trailer to help user get around using Ubuoy application.

### 6.2 Problems

The biggest problems when building the app were time and inexperience to build such a complex application. Although not every feature was implemented in the demo version, we were able to present our demo to the judges with the most import functions. A
major problem was to implement the design of Windows 8 features into .NET framework. After Windows have released their latest operating system, new “modern” theme gradually started to show up all over the web applications and sites. But we did not find a page with all basic features and function to be used on the web. We had to do a lot of research and study code from other developers to learn how they did things, like “dragable titles”, horizontal scroll, and responsive layout. We followed several tutorials, and learned new technologies from scratch.

Experience of one member in one area of technology and process and inexperience in another one slowed down or paused the development process and we were not able to proceed until we would explain and teach each other. Because we wanted to achieve the goals faster and were limited with time. Our test environment was HTML, CSS and JavaScript. We create the pages and manually placed “dummy” data into the code. Our challenge was to convert HTML elements into ASP.NET controls and connect database using model and Entity Framework.

Meeting times and place many times had to be rescheduled because of ongoing studies or work hours. There was always some kind of challenge with moving the project forward. At the first stages of the project we chose the idea, created a name and planed our next move. As the development project went on, we faced coding problems and spent hours on debugging.

6.3 Solutions

Our team was not created of professionals, but rather of people with desire to learn and create. We learned from one another and received great input from our mentor and supervisor, Mr Kari Aaltonen

During the dynamic controls creation we realize that there are HTML tags that have no equivalent ASP.NET tags. We wanted to keep the structure of our tiles as we designed in HTML testing environment, but we faced the challenge when we tried to convert and dynamically create these kind of tags. We found that solution for adding an HTML tag dynamically from code behind can be achieved using Literal control. Literal control is used to display text in the ASPX pages, so by placing HTML code into literal text value attribute from code-behind and then nest it into ASP.Net container. As a re-
sult, we get a page where HTML tags placed in literals would render correctly and be nested into specified containers.

Adding data to database manually would have taken ages. The solution for adding “dummy data” to the database tables was to create forms using Entity framework and models. Then anyone from the team was able to add, edit, and delete data using created forms.

Team meetings had to be done remotely using Skype its Screen Sharing feature. The only problem was that we could share the screen between two team members at the time. Skype reserved the option to share it with the group to Premium users only. Dropbox and emails, helped us to exchange files and keep the whole team up-to-date.

6.4 Result

When we submitted the Ubuoy application demo onto competition it had three main features that Windows 8 have. A page with just horizontal scroll and animated tiles. App bar for function like command to add or edit something in the application. Finally, we were able to implement charms with hide and show features developed using JavaScript. This features plus the look styled using Modern theme of Windows 8 probably the only application on the web at the moment.

Figures 36-39 shows the final result of Ubuoy application. We as a team were very satisfied with the result and have expectations to take this project further and develop it into a working web application, installed on a server.

We are looking forward to the results from the competition officials and hope to be part of the finals in St. Petersburg, in July 2013.
7 Conclusion

The purpose of this work was to study the process of designing and building web application using .NET Framework with latest web development technologies such as HTML5, CSS3, and JavaScript.

After many hours of research and hands-on experience of building an application, our team was able to select the most useful approaches for different stages of creating dynamic application. We implemented new features to the site because we discovered new or easier ways to create them. As a result, we had a great learning experience in planning and in the design and development processes of application creation.

Building an application from scratch requires great amount of time and effort. Creating an application with fixed specifications and features and limited time, leads you to find solutions on the go and use as much research and help as possible. During the process we discovered new ways of working with code, use latest standards of CSS3 and HTML5 which are simplified and provide us with several new options to proceed when developing dynamic application.

Our team learned that by using Visual Studio build-in technologies like Entity Framework, Master Pages and User Controls we can create an application, where it is possible to modify one part of an application without affecting another. We realised that converting HTML elements to ASP.NET elements when possible will give us control over those elements at runtime and that they are easier created in loops, because they require fewer lines of code.

The biggest success of the application is we were able to prove to ourselves that new Modern style theme features used in operating system and apps, can be applied to any dynamic application and be used on the web.
References

1 XAML AppBar control sample [online].
Microsoft; 2013
URL: http://code.msdn.microsoft.com/windowsapps/
XAML-AppBar-control-sample-2aa1cbb4

2 ASP.NET Tutorial [online].
W3Scools; 1999-2013
URL: http://www.w3schools.com/aspnet

3 Buoy. The Free Dictionary by Farlex [online].
Farlex, Inc; 2013
http://www.thefreedictionary.com/buoy

4 C#. Webopedia [online].
QuinStreet Inc; 2013
http://www.webopedia.com/TERM/C/C_sharp.html

5 Windows 8 & Windows RT Tutorial [online].
Microsoft; 2013

6 CSS3 Tutorial [online].
W3Schools; 1999-2013
http://www.w3schools.com/css3

7 HTML5 Introduction [online].
W3Schools; 1999-2013
http://www.w3schools.com/html/html5_intro.asp

8 JavaScript Tutorial [online].
W3Schools; 1999-2013
http://www.w3schools.com/js
9 SWOT Analysis [online].
Management Study Guide; 2008-2013
http://www.managementstudyguide.com/swot-analysis.htm

10 Team Foundation Server [online].
Microsoft; 2013

11 What is Visual Studio? [online].
Whatisvisualstudio.com; 2013
http://whatisvisualstudio.com

12 Windows 8 [online]
TechTarget; 1999-2013
http://whatis.techtarget.com/definition/Windows-8

13 Windows Store [online]
TechTarget; 2011-2013
http://searchconsumerization.techtarget.com/definition/Windows-Store
techtarget.com

14 Buoy. Wiktionary [online].
Wiktionary; 2013
http://en.wiktionary.org/wiki/buoy

15 Bernsen NO, Dybkjær L. Multimodal Usability.
Springer-Verlag London Limited; 2009.


17 Windows 8: Creating Great App Bar Buttons, 2013 [online].
Liberty J.; 2013
URL: http://jesseliberty.com/2013/02/05/
windows-8-creating-great-app-bar-buttons.
18 Free Windows 8 Wireframe Templates for PowerPoint, 2013 [online]. Designmodo; 2013

19 Novak I, Arvai Z, Balassy G, Fulop D.
Beginning Windows 8 Application Development. Wrox; 2012

20 Introduction to ASP.NET and Web Forms [online].
Microsoft; 2013
URL: http://msdn.microsoft.com/en-us/library/ms973868.aspx#introwebforms_topic1

21 Chast I. Planirovanie i dezain veb-saita [online]. Dialektika; 1992-2013


23 Selecting Development Approach [online].
Department of Health and Human Services, USA; Original Issuance: February 17, 2005, Revalidated: March 27, 2008.

24 Use Case-Base Software Development [online].
Haumer P., IBM Rational Software; 2003
URL: http://haumer.net/paper/UC-Based_Software_Development.pdf
Accessed. 27 April 2013.


27 JavaScript Data Types [online].
W3Scools; 1999-2013
URL: http://www.w3schools.com/js/js_datatypes.asp


29 Get Started with Entity Framework (EF) [online].
Microsoft; 2013
URL: http://msdn.microsoft.com/en-us/data/ee712907


31 ASP.NET Dynamic Data Overview [online].
Microsoft; 2013

32 Walkthrough: Creating a New Dynamic Data Web Site Using Scaffolding [online]. Microsoft; 2013

33 Walkthrough: Enabling Dynamic Data in ASP.NET Data-Bound Controls [online]. Microsoft; 2013
http://msdn.microsoft.com/en-us/library/cc488549%28v=vs.100%29.aspx

34 Walkthrough: Create a Web Site that Uses Dynamic Data [online].
Microsoft; 2013
http://msdn.microsoft.com/en-us/library/cc668188%28v=vs.100%29.aspx

35 How to: Add Controls to an ASP.NET Web Page Programmatically [online].
Microsoft; 2013
36 While [online].
Microsoft; 2013

37 Do...while Loops [online].
Microsoft; 2013

38 For Loops [online].
Microsoft; 2013

39 Foreach, in (C# Reference) [online].
Microsoft; 2013
URL: http://msdn.microsoft.com/en-us/library/txw7t8t6%28v=vs.80%29.aspx

40 LESS [online]
Sellier A.; 2010-2013
URL: http://lesscss.org