



# Technical Documentation

## IKT – Tools & Project-library

Applied Programming, 100 credits  
Upper secondary engineering-program, T4  
Spring Semester 2018

Mentor: Martin Larsson

## Contents

1	Introduction.....	1
1.1	Developers.....	1
1.2	Purpose.....	1
2	Target Group and Needs .....	2
2.1	Target Group.....	2
2.2	Client Needs.....	2
2.2.1	Database.....	2
2.2.2	PDF Upload & Download .....	3
3	Service Description.....	4
3.1	Usage .....	4
3.2	Implemented Functionality .....	5
4	Architectural Design .....	6
4.1	System Overview .....	6
5	Interaction Design .....	8
5.1	Layout & Structure.....	8
5.1.1	Logo .....	10
5.1.2	Colors.....	10
5.1.3	Font.....	11
5.2	Views.....	11
5.2.1	First log-in view .....	11
5.2.2	Start-page view .....	13
5.2.3	Create views .....	14
5.2.4	Edit views.....	15
5.2.5	Detail view .....	16
6	System Design .....	17
6.1	Database design.....	17
6.2	MVC .....	17
6.2.1	Models.....	18
6.2.2	Controllers.....	18
6.3	Login.....	21
7	Implementation.....	22
7.1	ASP.NET Application .....	22
7.1.1	Entity Frameworks .....	22
7.1.2	Paged List MVC.....	22

7.1.3	Testing .....	22
7.2	Design implementation .....	22
7.2.1	Bootstrap .....	23
7.2.2	jQuery .....	24
7.2.3	jQueryUI .....	24
7.2.4	pdfJS .....	25
7.2.5	Google Fonts.....	25
7.3	Limitations .....	25
8	Tools & Frameworks.....	26

# 1 Introduction

*IKT* is an ASP.NET application for storing various projects and tools that have been used in school. The program is created for teachers, to share experiences of tools or projects with other coworkers. But also for all teachers to see which class is currently working with which tools. To make sure only teachers can enter the site a login from Jönköpings Kommun based on Microsoft's login-system is used. This login is customized with some constraints that makes it inaccessible for students. All information is stored in a database and is easily accessible from the website.

This documentation contains descriptions of which functions has been implemented in the application and how they works. As well as which external tools, packages and frameworks have been used. Lastly it also mentions why particular design choices have been made.

## 1.1 Developers

Back-end	Front-end
Christoffer Ekström	Edvin Fagerström
Daria Lewandowska	Joel Hultqvist
Marcus Perhamn	Max Skogward

## 1.2 Purpose

The purpose of this work was to develop a working application for storing data about school-projects and tools that have been used in some way. Likewise, to learn how to work in a developer teams, work towards a customer's requirements and follow the basics of a scrum-based project.

## 2 Target Group and Needs

This chapter is describing the specific target group along with the customer needs and requirements on the IKT-application.

### 2.1 Target Group

The target group for *IKT* is mainly teachers and eventually other personal at *Erik Dahlbergsgymnasiet* in Jönköping. But it could also be used in other schools, though only in Sweden since the app language is Swedish.

All the teachers are performing different projects and using diverse tools in various classes. Owing the fact that it's very hard to remember all the tools you're using with which class, they needed something that can keep track on those things. For instance, the projects that have been accomplished as well as the tools that several classes are using. Additionally, *IKT* is a tool for sharing information with each other too. That means that all teachers can see which projects have been made and redo them with new/different classes.

There are many kinds of teachers at *Erik Dahlbergsgymnasiet* since it's a pretty big school, which implicate many different technical and computer skills. Some teachers may only know some basics on how to use an application, seeing that the application had to be done in a way that everyone can understand.

### 2.2 Client Needs

The teachers need a program that can keep track of all projects and tools. Since so much is changing in the school world and a big amount of external tools are used on a daily basis, it's really hard to remember everything by yourself. To ease their daily work, they need an application that can store those things for them.

To sum up the application must handle input of projects and tools and store those in a database, make it possible for all users to view the information, personal login and eventually PDF-file upload.

#### 2.2.1 Database

The database role in this project is to store projects and tools which users put in. The data should be easily accessible for all users to read and eventually change. To make sure that no unwanted data is stored in the database, constraints in every input-field are in place to prevent that. There is no sensitive information stored in the database therefore it does not have to be extra secure. Even though an authentication system is implemented, which means you have to log in to view and/or change the data.

### 2.2.2 PDF Upload & Download

When a project is made it often ends with a PDF-document where all information about it is saved. To ease the access for that PDF it is possible to upload it to the site and show it in the webbrowser. It is possible to download the PDF-file from the browser.

### 3 Service Description

This chapter describes which services *IKT* currently provides.

#### 3.1 Usage

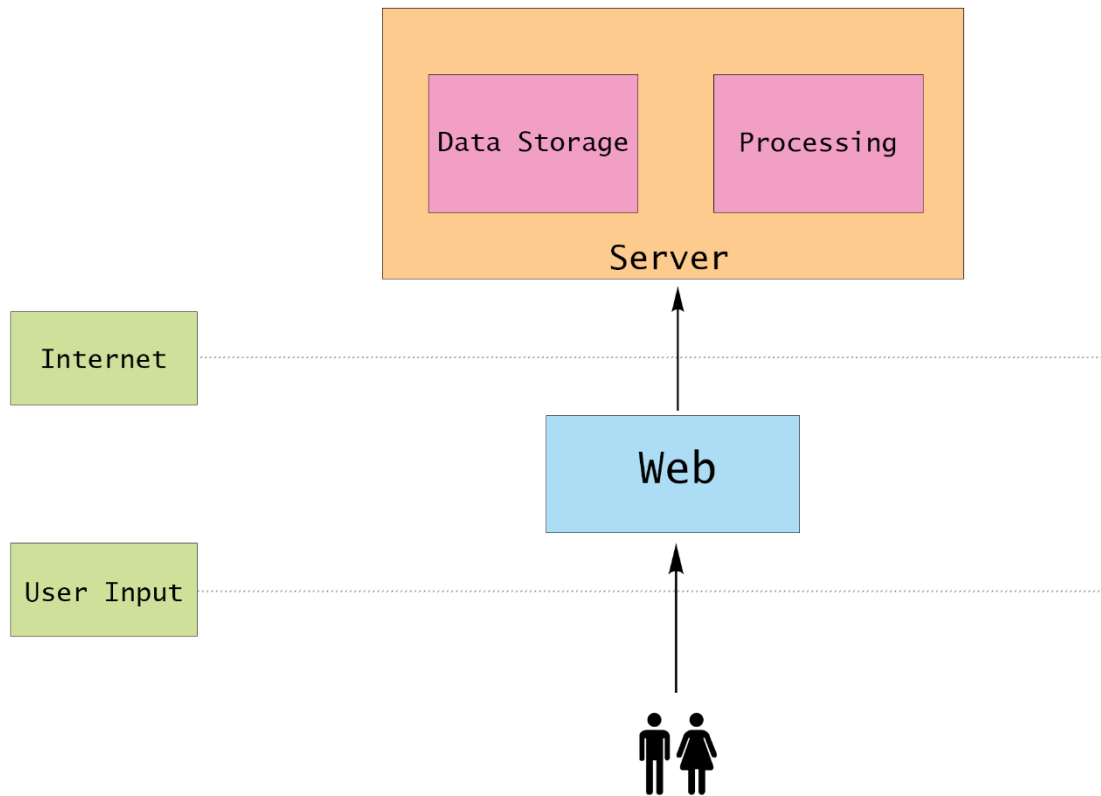


Figure 1 - Communication of the application

The application is web-based, it is possible to access it from any webbrowser. To make it more flexible the website is customized for mobile phones and tablets as well. When a user chooses a task, for example viewing a project. The task is sent to the internet and communicates with the server as the figure 1 shows. Then it returns a response to the user, in this case it would be the view of the chosen project.

## 3.2 Implemented Functionality

*IKT* offers following implemented functionality for all platforms.

- Login & Logout via Jonkoping Kommuns login-system.
- Viewing all projects or tools.
- Creating new projects or tools.
- Editing existing projects or tools.
- Removing projects or tools.
- Adding yourself as an user of a tools or contributor of a project.
- Searching all tools and projects.
- Filtering tools and projects by subject and grade.
- Sorting tools and projects alphabetically or by date.
- Uploading a PDF-file as a complement to a project.
- Viewing project and tools on several pages.



## 4 Architectural Design

This chapter will explain the architecture of the system on an extensive level to give a deeper understanding on how and why the application is designed as it is.

### 4.1 System Overview

*IKT* is a server-client architecture. It only consists of one client though, the web. The implementation is based on the MVC (Model, View, Controller) design pattern. The pattern's basic idea is separating data and logic from the representation and user interface as it appears in figure 2.

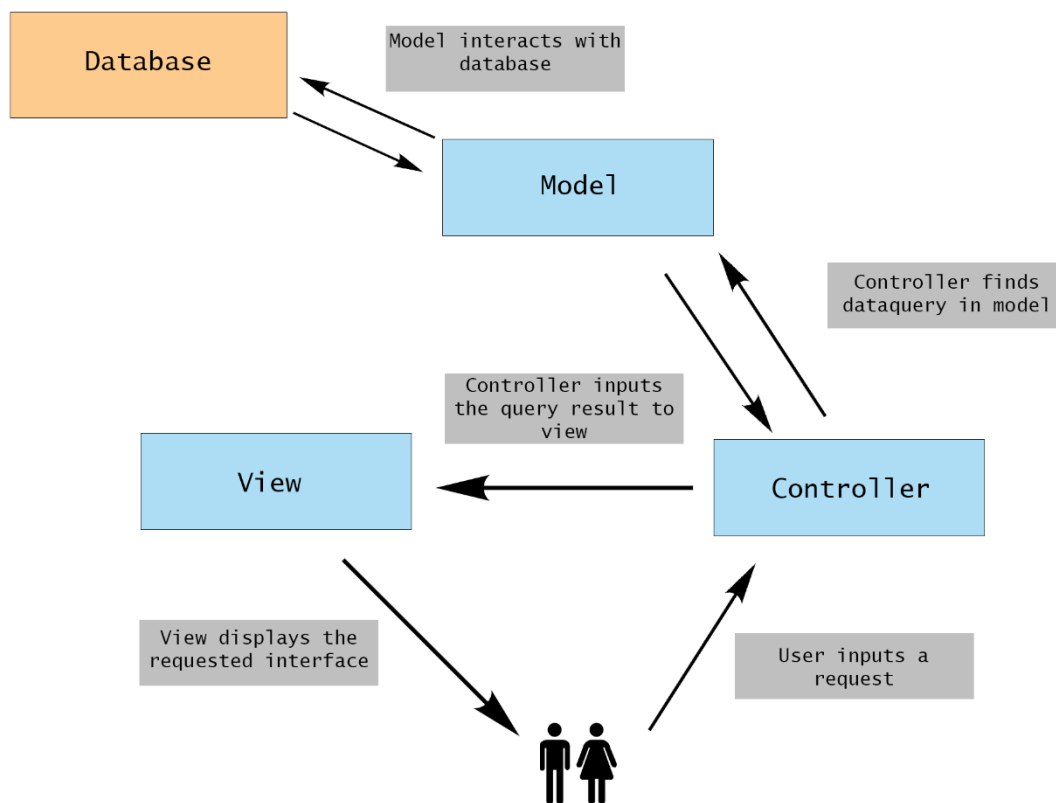


Figure 2 – MVC Model

The view contains lists of projects and tools, information about those and other information for the users. It also contains buttons, input-fields and several other elements that makes it possible to add and modify data. When a user interacts with those in the graphical interface, the controller will handle it and modify the model which the data relies on. The model will communicate with the database. Finally when the controller gets its response from the model it will send it back to the view, and it will be appear on the users interface.

The advantage of using this kind of design pattern is that the user interface can be changed or replaced easily while logic still can remain the same.

## 5 Interaction Design

This chapter is about how the graphical and interactive parts of the application are formed and designed.

### 5.1 Layout & Structure

A framework is a mockup of the design and layout of the website, which is used for getting a general sense of what the final product will look like. They tend to look very barebones, and don't have many details or functions. But none of that is needed, when all you need is just some boxed that mark where a menu or slideshow will be. The frameworks used for this project are attached as figure 3, 4 and 5.

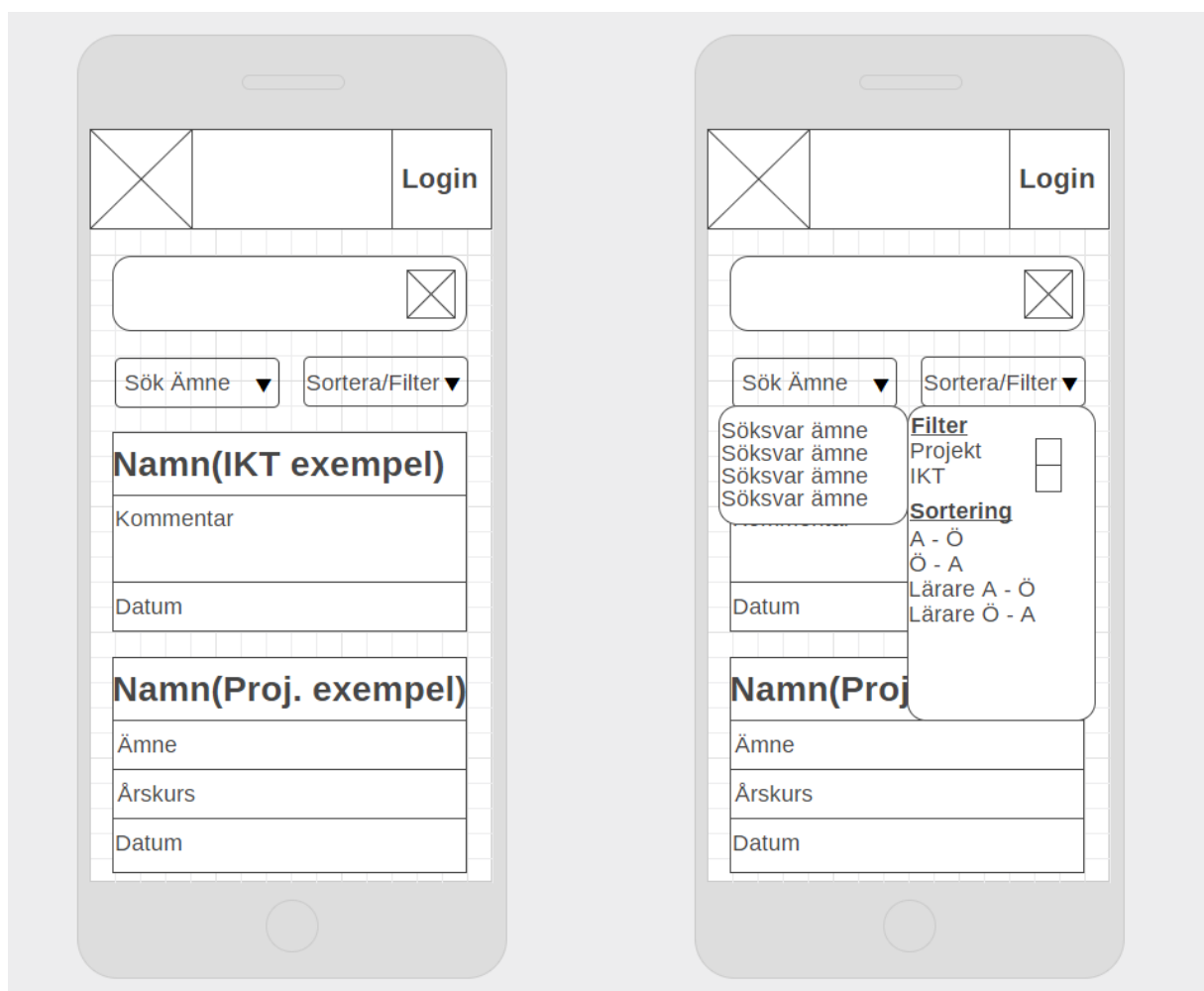


Figure 3 – Frameworks Mobile Device

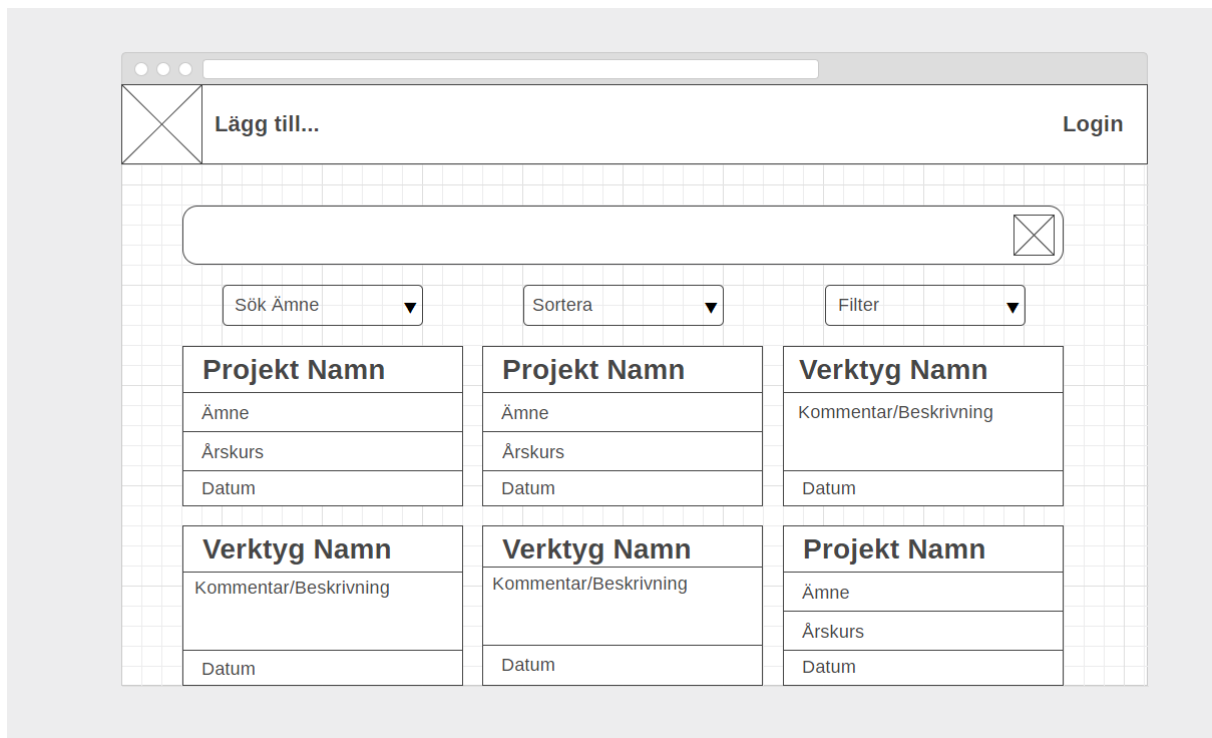


Figure 5 – Framework start page

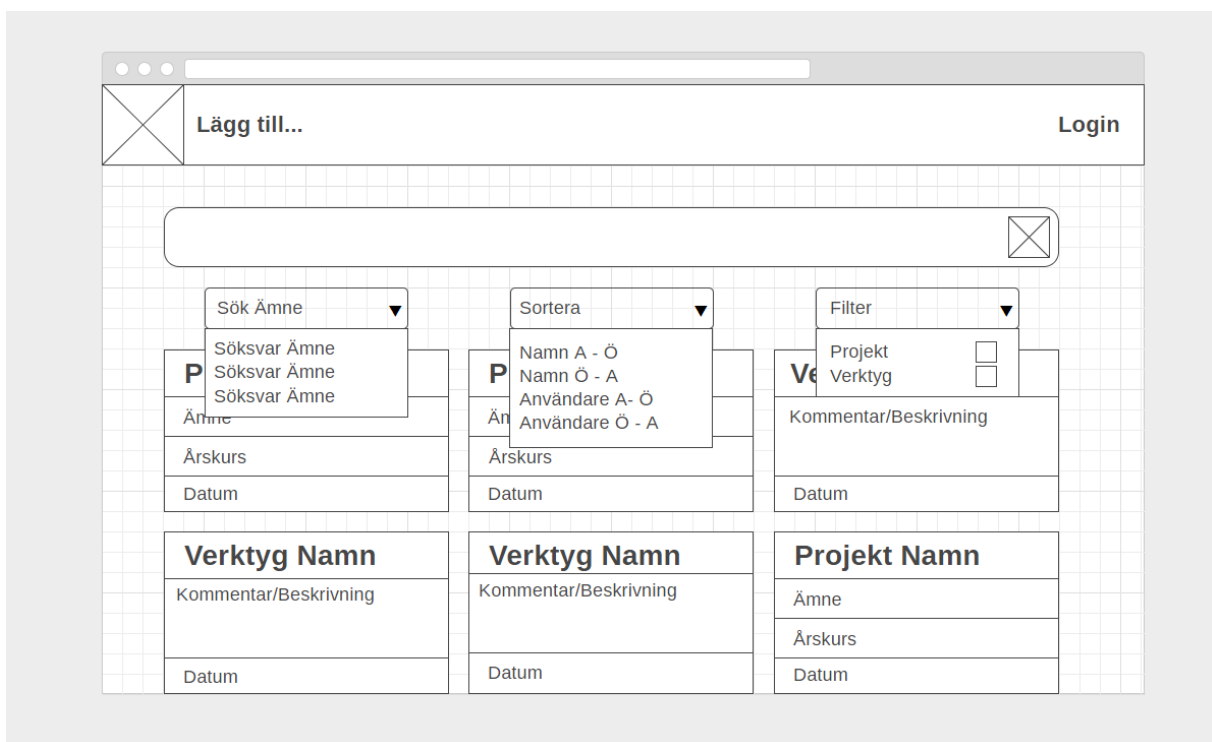


Figure 6 – Framework start page with dropdowns

The final design is way more refined and polished than the initial framework. More details, effects and colors. Instead of everything looking like white boxes, there is now color that gives the site some excitement and shapes that look gentler.

### 5.1.1 Logo

The logotype shows the face a cogwheel and the name of the system, see figure 7. The colors of the logotype make up for the bright colors in the chosen color scheme.

A cogwheel was used in the logotype due the fact that the cogwheel is often a symbolize for productivity when paired correct with others. Where in this projects perspective "the others" are the users and the cogwheel in the logotype represents the system.

When the user and the system then interact with each other it creates a perfect symbiosis.



Figure 7 – Logotype

### 5.1.2 Colors

The color scheme chosen is three gray scale colors with two light blue/green ones, see figure 8. The user interface of the system needed to have a clean look and that also includes the colors of the page so that is why two simple color scales were used.

The color green symbolizes mainly renewal, energy, harmony, safety and growth and that is what the system stands for and that is why this color was the main one.

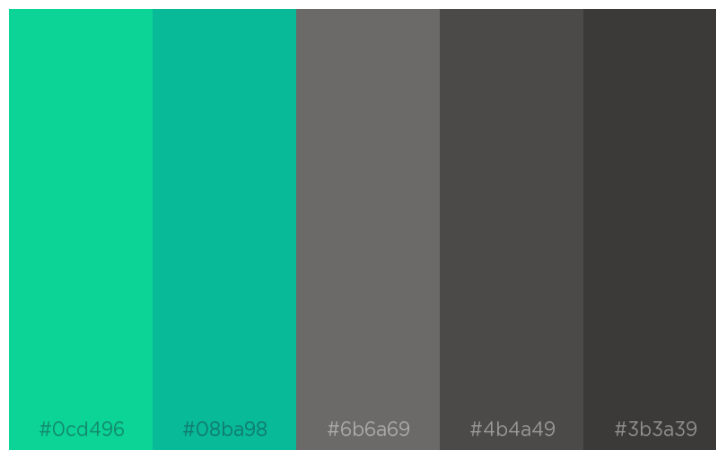


Figure 8 – Color Scheme

### 5.1.3 Font

The font chosen to this project was the "Poppins", it is a geometric sans serif typeface. The font "Poppins" have been very popular since it first was invented by Satya Rajpurohit and Peter Bil'ak in 2009. Poppins is one of the new comers to this long tradition. With support for the Devanagari and Latin writing systems, it is an internationalist take on the genre.

Many of the Latin glyphs (such as the ampersand) are more constructed and rationalist than is typical. The Devanagari design is particularly new, and is the first ever Devanagari typeface with a range of weights in this genre. Just like the Latin, the Devanagari is based on pure geometry, particularly circles.

Each letterform is nearly monolinear, with optical corrections applied to stroke joints where necessary to maintain an even typographic color. The Devanagari base character height and the Latin ascender height are equal; Latin capital letters are shorter than the Devanagari characters, and the Latin x-height is set rather high.

Thin  
*Thin Italic*  
Extra-Light  
*Extra-Light Italic*  
Light  
*Light Italic*  
Regular  
*Regular Italic*  
Medium  
*Medium Italic*  
**Semi-Bold**  
***Semi-Bold Italic***

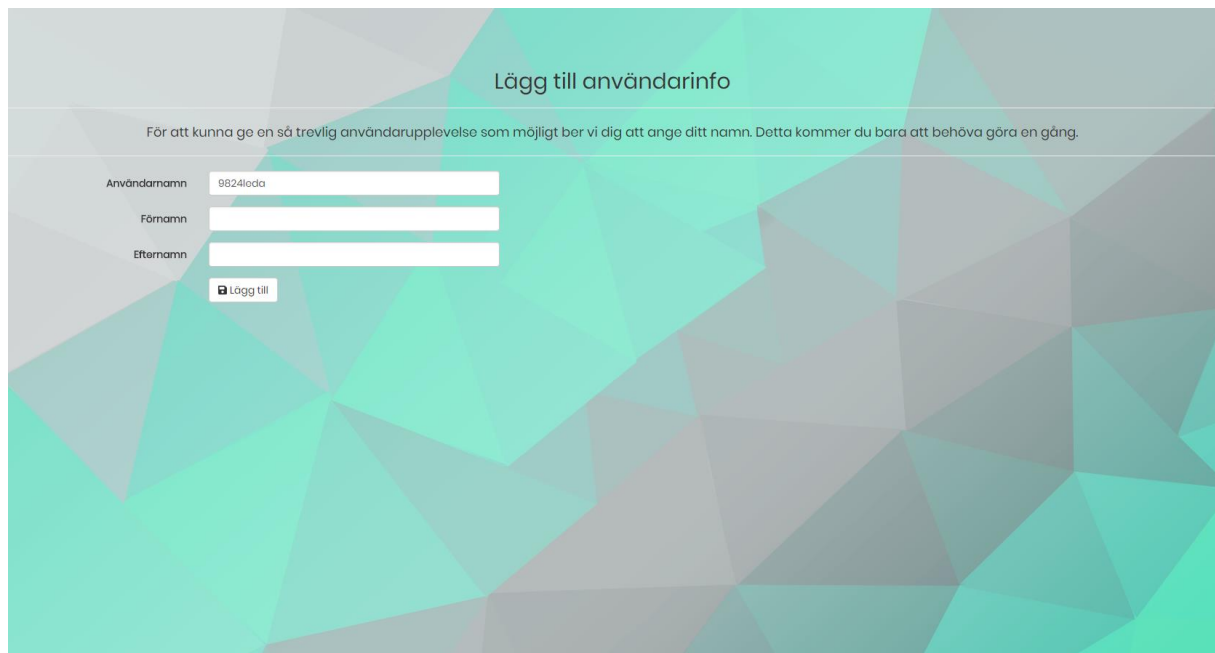
Figure 9 – Font

## 5.2 Views

This section is about the views look. How the views are presented and explaining the ulterior motive of the design.

### 5.2.1 First log-in view

When a user enters the website for the first time he is redirected to a view where he has to enter his first and last name, see figure 10. This is necessary to make it feasible for the application to save the users name when he is adding new projects or tools or editing existing ones in the database. The name is also used to welcome the user and to see whom is working with which project or using which tools, see figure 11 and 12.



Lägg till användarinfo

För att kunna ge en så trevlig användarupplevelse som möjligt ber vi dig att ange ditt namn. Detta kommer du bara att behöva göra en gång.

Användarnamn

Förnamn

Efternamn

Figure 10 – First Login View



Dokumentärfilm

Projekt

<b>Ämne</b> Svenska	<b>Beskrivning</b> Ett arbete inom svenska 1 där eleverna spelar in egna dokumentärfilmer kring ämnesområdet språksociologi.
<b>Årskurs</b> 1	
<b>Lärare</b> Lars Svensson Marcus Perhamn	
<b>Klasser</b>	
<b>Datum för utförande</b> 2017-12-08	

Figure 11 – Detail view where the names are outlined

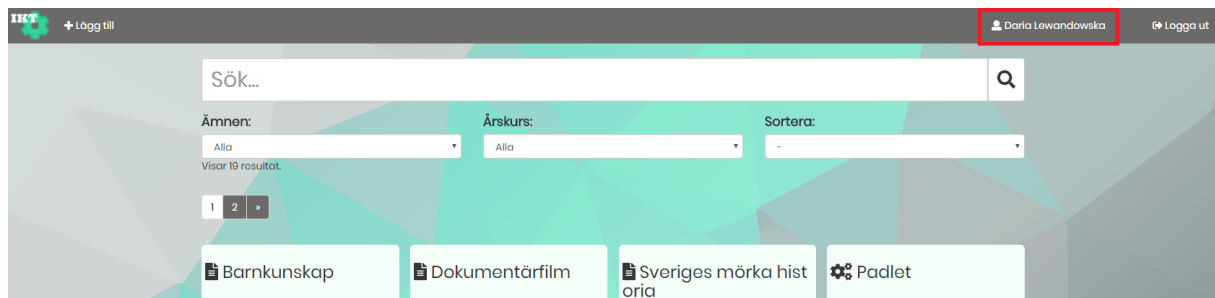


Figure 12 – Start page where name is outlined

### 5.2.2 Start-page view

The design's main thought was to reduce the amount of noise on the screen and try to get it clean and easy for the eyes. It was pretty much agreed right away that it was going to be a card grid layout for the search results, 3 x 3 grid.

It was discussed which content should be included on the result cards, to make the user have a good understanding of what the project or tool was about before going to the detailed view, before clicking in on it.

There was decided to be a big title in the top middle of the card, and some smaller chunks of text about:

- Which grade the project/tool was made for
- Which subject the project/tool was made for
- An icon to display if it is a project or tool

It was discussed where the placement of filter and dropdowns should be and what should be written on them. There were ideas like having them in a side panel, but in the end, they were placed right above the card grid in a row of 3.

The search bar was supposed to be the biggest core part of the site visually, so we made it bigger than the rest of the text bars on the front page. Naturally the search bar was placed at the top of the website so it is the first thing the user looks at when gazing from top to bottom. See figure 13 to see the view of the start page.



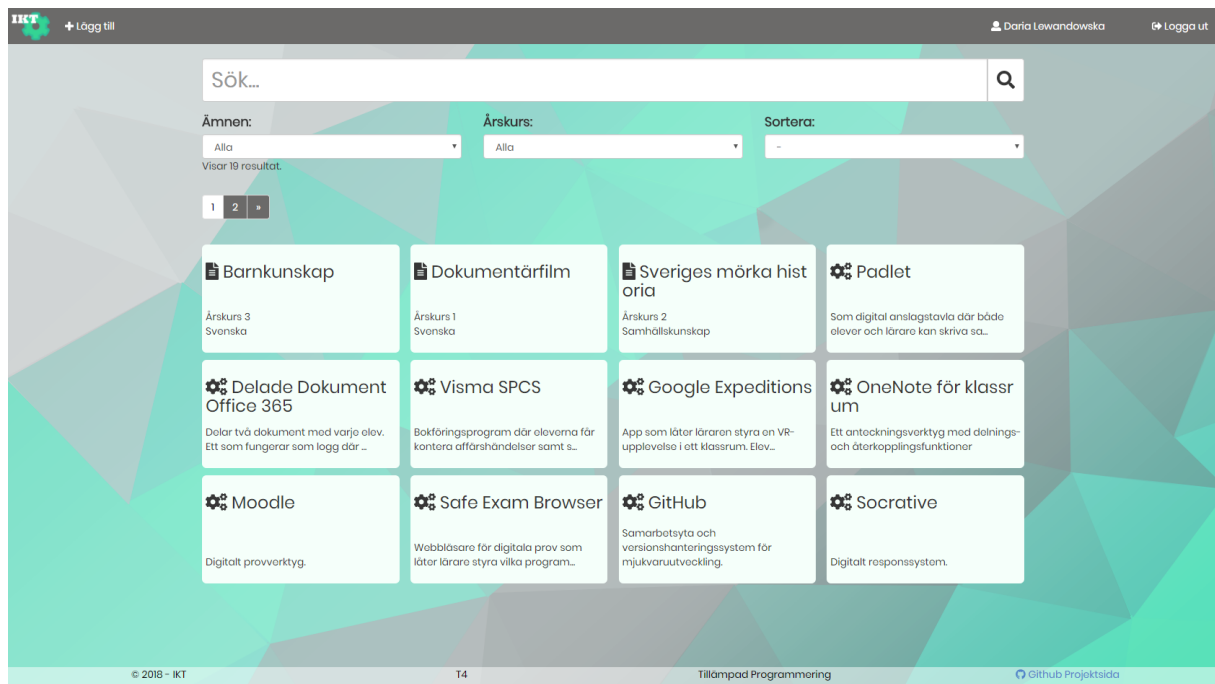


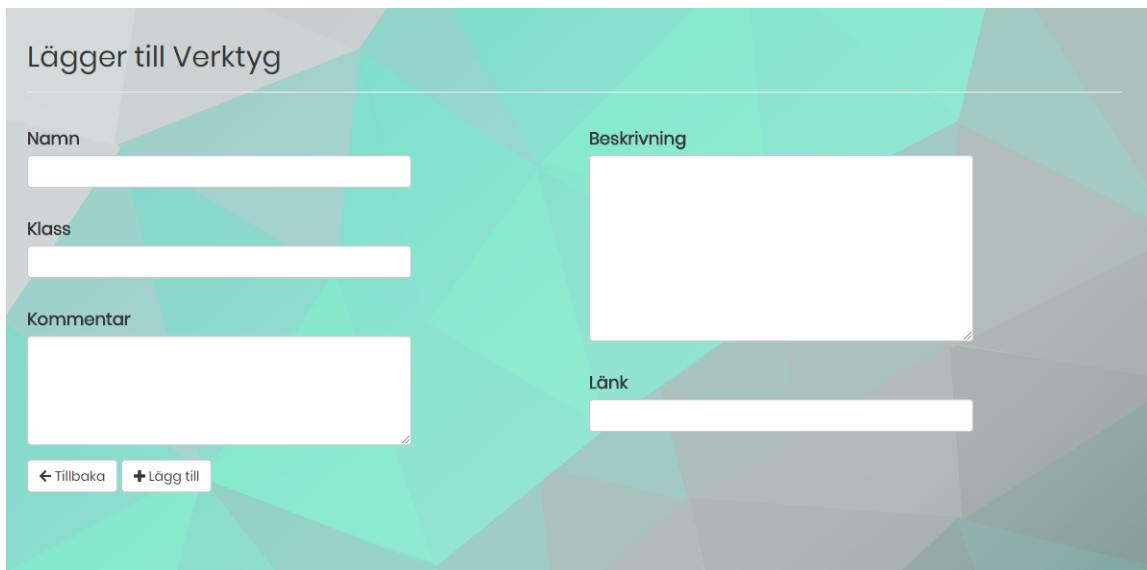
Figure 13 – Start page

### 5.2.3 Create views

The create views were made so that you could add projects and tools. Easily understandable with a text field for all the necessary information, and also the option to add a PDF file if needed.

The idea was to make users feel at home and familiar with the design we made, instead of feeling estranged and confused. Quite straightforward, while still feeling welcoming.

Figure 14 – Create view for projects



Lägger till Verktyg

Namn

Klass

Beskrivning

Länk

Kommentar

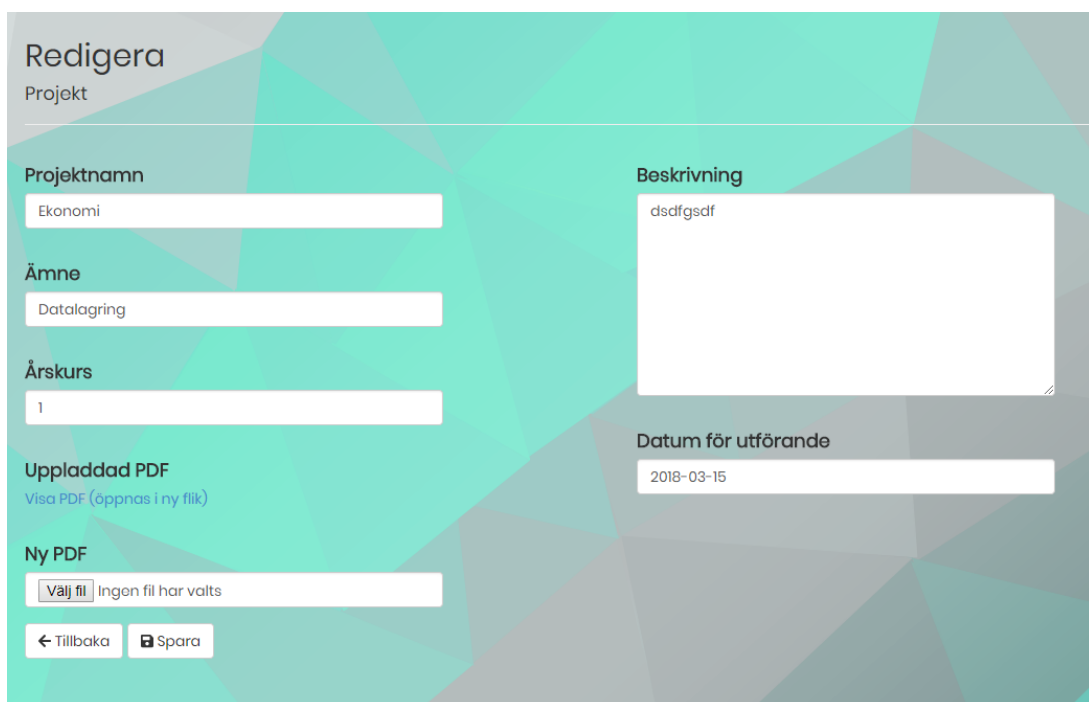
← Tillbaka + Lägg till

Figure 15 – Create view for tools

#### 5.2.4 Edit views

The editing window was made to make changes to a certain project or tool without going in to the database manually.

The design is built on a simple formula nothing fancy, yet it is easily understandable and does the job excellent. There should be no doubts about how it works when a user enters the edit window for the first time. It should follow the standard most websites use so the user should feel familiar with it right away.



Redigera Projekt

Projektnamn

Ämne

Årskurs

Beskrivning

Datum för utförande

Uppladdad PDF

Ny PDF

← Tillbaka Spara

Figure 16 – Edit view for projects

### 5.2.5 Detail view

The detail view provides a more detailed view of a project or tool the user can find, see figure 17.

In details the user gets to see who made the project, who is using it and if the user selected a project he can even get a preview of the first page in a PDF-file if one is connected to the project, just for convenience. That is to say, the user gets some more information presented in the details view about a project or tool which he chooses to read more about, see figure below.

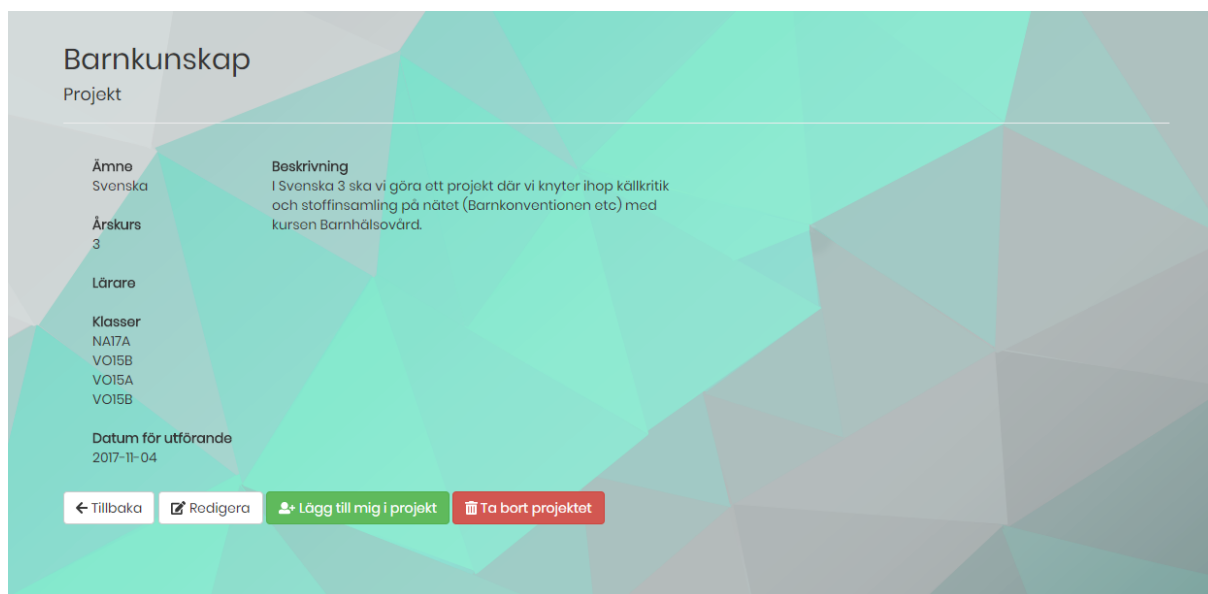


Figure 17 – Details view for a project

## 6 System Design

This chapter goes into detail on how the system is designed and build.

### 6.1 Database design

The database is designed according figure 18.

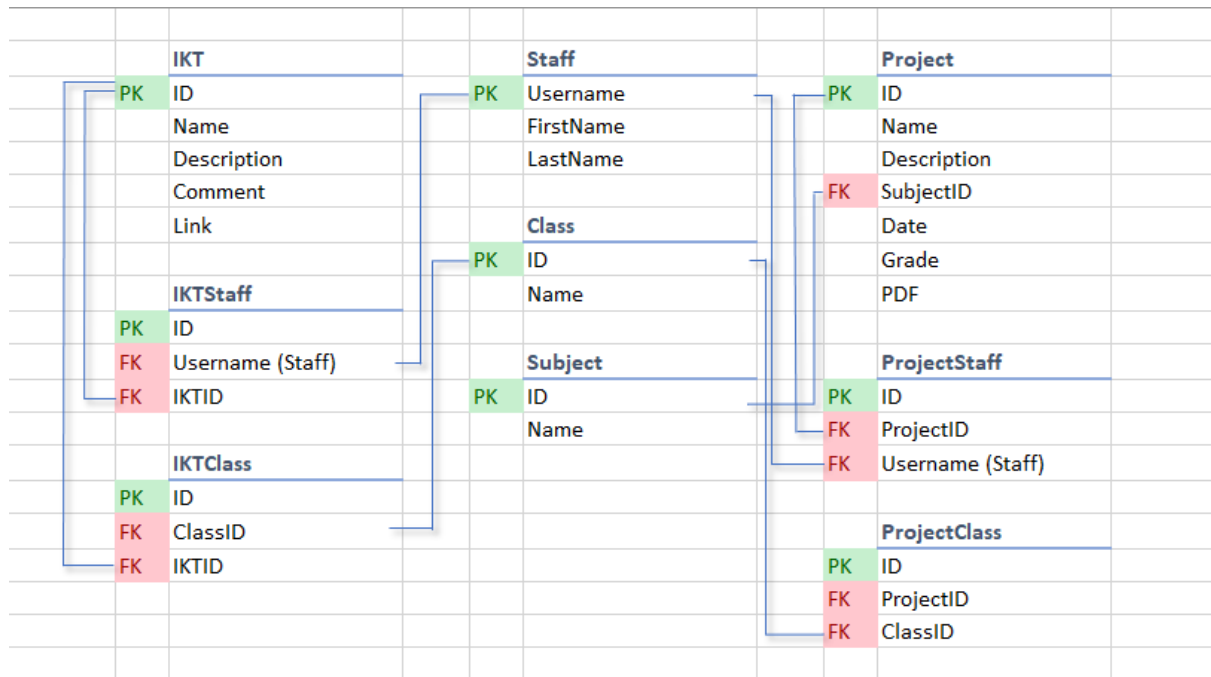


Figure 18 – Database Design

PK stands for Primary Key. FK stands for Foreign Key.

The lines between keys are showing how several tables are connected to each other.

### 6.2 MVC

The application is built upon MVC design pattern. As earlier mentioned the patterns basic idea is separating data and logic from the representation and user interface. It consists of three components, model, view and controller. Model manages data, logic and rules of the application. View is the output of the information that the model holds. The third component, controller, accepts input and converts it for tasks for either model or view. It's almost like a communicator between model and view.

### 6.2.1 Models

The models that are existing in *IKT* are following.

- Ikt
- IktMetaData
- IktClass
- IktStaff
- IktSubject
- Project
- ProjectMetaData
- ProjectStaff
- SearchItem
- Class
- ClassMetaData
- Staff
- StaffMetaData
- Subject
- SubjectMetaData

The models that start with *Ikt* are associated with each other, they are building up the tool-object. Same with project, the ones that start with *Project*, are associated and are together constructing a project-element. Subject, Class and Staff are models that are connected to both *Ikt* and *Projects* owing the fact that those inputs are mutual for both tools and projects. The MetaData-models are subclasses that sets constraints for the superclasses, for instance sets fields to required or sets a max-length for a field. How the models are related to each other is made clear in chapter 6.1 Database Design.

### 6.2.2 Controllers

A controller is a listener for actions, when an action appears the controller catches it and modifies the model for output. Then it converts the response, send it to the view so the requested information is shown.

The following controllers are implemented and their basic tasks described below in Table 1.

Controller Name	Tasks
AccountController	Login and Logout
ClassesController	Showing, creating, editing and removing classes.
FetchDataController	Converting list of subjects or classes to JSON-format. No views.
HomeController	Showing start page, contains all the functions for sorting, filtering and searching.
IktsController	Showing, creating, editing and removing tools. Also adding and removing staff from tools.
ProjectsController	Showing, creating, editing and removing projects. Also adding and removing staff from projects.
StaffsController	Showing, creating, editing and removing staff.
SubjectsController	Showing, creating, editing and removing subjects.

Table 1 – Controllers and their tasks.

#### 6.2.2.1 Searching

The search-function is built in HomeController since the start page is where the search is happening. The search is quite complex seeing that it finds matching items in both the tool-table and the project-table in the database. In order to make it work several tables are joined together. Those are Staff, Subject and Class for the reason that the search to be complete and search for almost all items in the tables. There are basically two searches are done with those joins, one in projects and one in tools. When both searches are done the two lists are put together to be shown at the website as one full search, also containing all filters and sorting that the user might have chosen.

#### 6.2.2.2 Filtering

The filtering is also implemented in the HomeController. It is possible to filter the items on grade and subjects since that is what was most relevant and useful to put a filter on. The idea behind the filtering is to simplify finding objects and make the application more user-friendly.

#### 6.2.2.3 Sorting

Sorting is placed in HomeController as well as the other functions. The user of IKT can sort items by four sorting algorithms. Those are alphabetical order, alphabetical order descending, newest item and oldest item. This is also a function that has been implemented to improve the user-friendliness on the website.

#### 6.2.2.4 JSON

In order to make the convenient autofill-input box on the creating pages, see figure 19. It was necessary to create something that could make the data (class-names and subject-names) available for other parts of the project, for instance for using in a JavaScript-file. In order to make it possible a new controller was created (FetchDataController), only for converting the data. In FetchDataController the data is collected from the database and converted to JSON-format. Later on it is called on in the JavaScript-file where the autocomplete for the input-field is happening.



Figure 19 – Dropdown with autofill function on create sites

### 6.3 Login

To make the application easy to use for the teachers in Jönköpings kommun, an already existing login system provided by Microsoft and Jönköpings kommun was chosen. This makes it so that the teachers do not have to memorize another password, and makes it easier to expand to other schools if wanted.

When a user clicks log in, he or she is presented with Microsoft's log in screen where they enter their email address. The log in system then detects that the user wants to use an email from Jönköping, and redirects the user to Jönköpings kommun's log in system, PortWise, where the user logs in with their normal username and password used for mostly all other work-related things.

This system is however open for both teachers and students, so to prevent students from logging in to the application a way to block access for students had to be implemented. This is done by checking the length of the username, and since all teachers have five letter usernames and students have four numbers and four letters, it is easy to determine if the user is a teacher or a student this way.

But since the developers are students, a white-list had to be implemented in the system so the developers can log in and develop the application. There are also a black-list if anyone misuse the system.



## 7 Implementation

This chapter is mentioning which frameworks and tools were used to create the application, both in backend and frontend.

### 7.1 ASP.NET Application

The web application is implemented in C# using ASP.NET MVC. Using SQL Server as database and the Authentication template "Work or School Accounts". By using this template, we do not need to handle the management of accounts since its organized by an organization. Special package Included is PagedList.MVC and the web application is handled by backend.

#### 7.1.1 Entity Frameworks

Entity Framework is used as an object relational mapper. That makes it easier for .NET developers to work with a database using .NET objects. It eliminates the need for most of the data-access code that developers usually need to write.

#### 7.1.2 Paged List MVC

To stop the page of search results from being an endless scroll, the plugin Paged List MVC was used. Paged List automatically creates a paged layout with a specified number of objects per page.

#### 7.1.3 Testing

The software was tested for bugs before released, both by the developers and other persons, like teachers whom are supposed to be using this application later. All the found bugs were added to the GitHub-Issue page and quickly fixed.

### 7.2 Design implementation

ASP.net was used for the project because of how much it helps you along the way. The design was made with HTML and CSS, bootstrap was used as well, for some groundwork like the login page, for the formulas, edit pages, practically everything. PDF.js library was also used.

When the basic layout of the website was agreed on. A canban board was used to set up the implementation tasks like "front page" or "edit page" to start the creation process.

Some of the imagery was handmade in photoshop, a lot of the icons was taken from font awesome, a site that provides good quality free icons.

ASP.NET comes with a navbar that is good looking and feels familiar to most users and a good tool for navigating, just the fact that a navbar exists makes the website much more understandable for users for instance a user knows the login usually is on the right side of the navbar if you press the icon on the left side you get to the front page and so on.

### 7.2.1 Bootstrap

Bootstrap is an open source toolkit for developing with HTML, CSS, and JS. Bootstrap can be very helpful creating powerful front-end designs for websites and web applications. It contains HTML- and CSS-based design templates for typography, buttons, forms, navigation and other interface components, as well as optional JavaScript extensions. Unlike other web frameworks, it concerns itself with front-end development only.

Bootstrap, originally named Twitter Blueprint, was developed by Mark Otto and Jacob Thornton at Twitter as a framework to encourage consistency across internal tools. Before Bootstrap, various libraries were used for interface development, which led to inconsistencies and a high maintenance burden.

After a few months of development by a small group, many developers at Twitter began to contribute to the project as a part of Hack Week, a hackathon-style week for the Twitter development team. It was renamed from Twitter Blueprint to Bootstrap, and released as an open source project on August 19, 2011. It has continued to be maintained by Mark Otto, Jacob Thornton, and a small group of core developers, as well as a large community of contributors.

Bootstrap has been and is being developed continuously and the last release was in the spring of 2018 when Bootstrap 4 was released.

Bootstrap is modular and consists of a series of Less stylesheets that implement the various components of the toolkit. These stylesheets are generally compiled into a bundle and included in web pages, but individual components can be included or removed. Bootstrap provides a number of configuration variables that control things such as color and padding of various components.

Each Bootstrap component consists of an HTML structure, CSS declarations, and in some cases accompanying JavaScript code.

Grid system and responsive design comes standard with an 1170 pixels wide grid layout. Alternatively, the developer can use a variable-width layout. For both cases, the toolkit has four variations to make use of different resolutions and types of devices: mobile phones, portrait and landscape, tablets and PCs with low and high resolution. Each variation adjusts the width of the columns, see example on figure 20.

In this project bootstrap was used for designing and fixing the layout of the page, exactly what is meant to be used for.

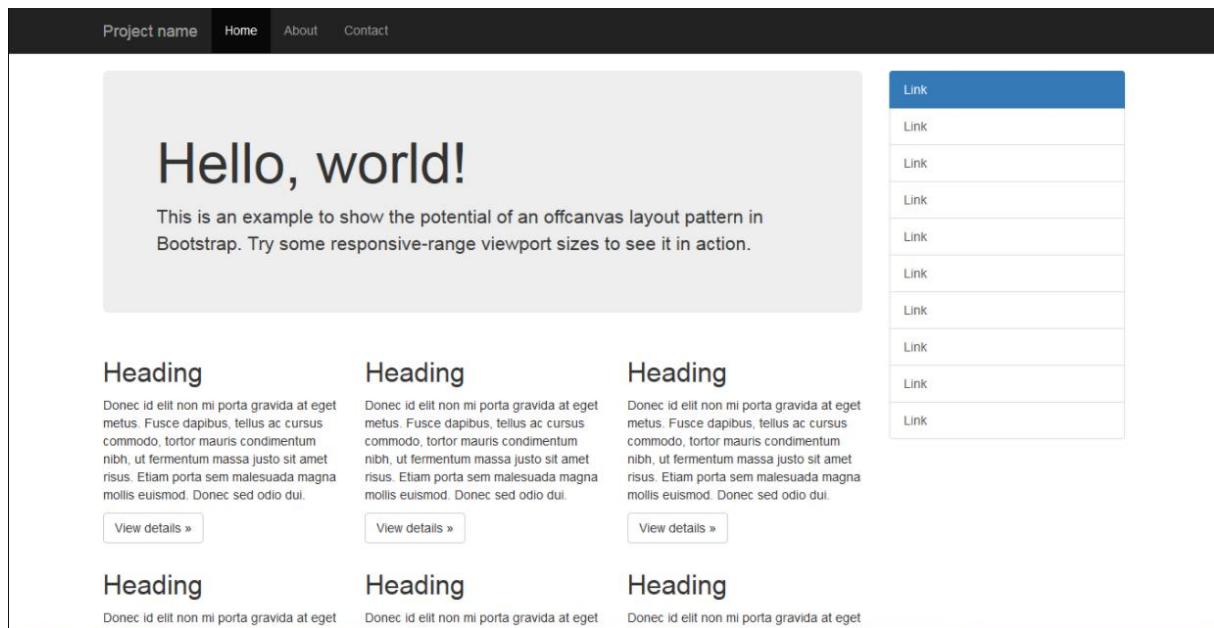


Figure 20 – Bootstrap Layout Example

### 7.2.2 jQuery

jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers. It is free, open-source software using the permissive MIT License.

jQuery's syntax is designed to make it easier to navigate a document, select DOM elements, create animations, handle events, and develop Ajax applications. jQuery also provides capabilities for developers to create plug-ins on top of the JavaScript library. This enables developers to create abstractions for low-level interaction and animation, advanced effects and high-level, theme able widgets. The modular approach to the jQuery library allows the creation of powerful dynamic web pages and Web applications.

jQuery was originally released in January 2006 at BarCamp NYC by John Resig and was influenced by Dean Edwards' earlier cssQuery library. It is currently maintained by a team of developers led by Timmy Willison.

jQuery was used for the Ajax-calls and for some other functions, like setting up a calendar for the create-page.

### 7.2.3 jQueryUI

jQuery UI is a collection of GUI widgets, animated visual effects, and themes implemented with jQuery, Cascading Style Sheets, and HTML. jQuery UI is free and open-source software distributed by the jQuery Foundation under the MIT License, jQuery UI was first published in September 2007.

Two examples of widgets that the system uses are:

- Autocomplete – Auto-complete boxes based on what the user types
- Date picker – Advanced date-picker

#### 7.2.4 pdfJS

pdfJS is a library for showing a preview of a PDF-file in a simple way. It is used in the project because imagine a user is searching for a document the user know exists for hours and hours by downloading a bunch of different PDF files just to open them and have a check if it is the right one. With the PDF preview you can simply see what it is about by displaying the front page and it is much more convenient to search for things when you can see what it looks like before you download it.

Basically, a PDF preview allows better presentation for a document, usually when a user looks for something and wants the best quality results they tend to go by presentation a lot because people who have good presentation tend to have used more effort on the project. People say do not judge a book by its cover, but people do and usually they are right to do so.

#### 7.2.5 Google Fonts

Google Fonts is a library of over 800 libre licensed fonts, an interactive web directory for browsing the library, and APIs for conveniently using the fonts via CSS and Android. The directory was launched in 2010 and revamped in 2011 and 2016.

Most of the fonts are released under the SIL Open Font License 1.1, while some are released under the Apache License, both are libre licenses. The font library is also distributed by Monotype's SkyFonts and Adobe's Edge Web Fonts and Typekit services.

The Google Fonts directory is intended to enable font discovery and exploration, and the service is used extensively with over 17 trillion fonts served.

The font used in this project was found in the google fonts library.

### 7.3 Limitations

The main limitation for this project was time, since the course is assigned 100 credits it means it is lasting 100 hours. There was also a budget limitation, it should not cost anything to build the project, though the developers got all the tools they needed, like for instance computers and software's for programming.

## 8 Tools & Frameworks

Bootstrap 3.3.7	<a href="https://getbootstrap.com/"><u>https://getbootstrap.com/</u></a>
Google Fonts	<a href="https://fonts.google.com/"><u>https://fonts.google.com/</u></a>
jQuery 3.3.1	<a href="http://jquery.com/"><u>http://jquery.com/</u></a>
jQueryUI	<a href="https://jqueryui.com/"><u>https://jqueryui.com/</u></a>
pdfJS	<a href="https://github.com/mozilla/pdf.js"><u>https://github.com/mozilla/pdf.js</u></a>
Visual Studio Enterprise 2017	<a href="https://www.visualstudio.com/vs/"><u>https://www.visualstudio.com/vs/</u></a>

