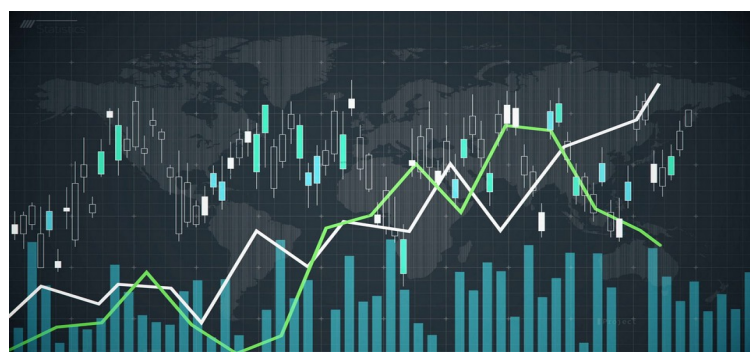


BSc Thesis

Reactive User Interface for Px statbanks

Helgi Poulsen



SERRITGERÐ
Thesis

TØKNIFRÁGREIÐING
Technical Report

UNDIRVÍSINGARTILFAR
Teaching Material

UPPRIT
Notes

NVDRit 2020:01

Heiti / Title ***Reactive User Interface for Px-statbanks -
A Functional Prototype***
Reaktivt nýtslumarkamót til Px
hagtalsgrunnar – Ein funktionel prototypa

Høvundar / Authors **Helgi Poulsen**

Vegleiðari / Supervisor Kári Holm Johannesen, Hagstova Føroya
Torbjørn Andreas Lisberg, Hagstova Føroya
Ábyrgdarvegleiðari / Responsible Supervisor Jari í Hjøllum, Fróðskaparsetur Føroya

Ritslag / Report Type BSc ritgerð, KT-Verkfrøði
BSc Thesis, Science in Software Engineering
Latið inn / Submitted 06. jan 2020

NVDRit 2020:01
© Náttúruvísindadeildin og høvundarnir
ISSN 1601-9741

Útgevandi / Publisher Náttúruvísindadeildin, Fróðskaparsetur Føroya
Bústaður / Address Nóatún 3, FO 100 Tórshavn, Føroyar (Faroe Islands)
☎ ☎ ☎ ☎ @ +298 352 550 ☎ +298 352 551 ☎ nvd@setur.fo

Bacheloruppgáva

Lesandi: Helgi Poulsen

Evni: Reactive User Interface for Px-statbanks – A Prototype
Reaktivt Nýtslumarkamót til Px-hagtalsgrunnar – Eitt Frumsnið
Ábyrgdarvegleiðari: Jari í Hjøllum

Innflatingardagur ^(*): 06.01.2020

() Um ritgerðin ikki er móttikin til avtalaða tíð, verður skrásett, at bacheloruppgávan er ikki staðin*

Víst verður til skeiðslýsingina fyri bacheloruppgávu í útbúgvingini, herundir fortreytirnar fyri at fara undir uppgávuna, umframt Reglur og mannagongdir. Eftir umsókn til Náttúruvísindadeildina kann í undantaksfæri verða vikið frá fortreytunum. Próvtal fyri bacheloruppgávuna verður ikki skrásett, fyrr enn tann lesandi hevur staðið øll skeiðini í síni útbúgvingini.

Bacheloruppgávan kann verða gjørd av antin einum lesandi ella av tveimum lesandi í felag, ið tó ikki viðførir dupult tímatal til vegleiðing ella til útreiðslur í samband við uppgávuna. Eru tvey lesandi, skal íkastið hjá hvørjum teirra kunna eyðmerkjast.

Fíggjarviðurskipti

Áðrenn byrjað verður, skulu møguligar útreiðslur til útgerð o.a. verða avkláraðar, eins og tað skal vera greitt, hvør rindar fyri útreiðslurnar. Er talan um størri útreiðslur, áliggur tað lesandi og ábyrgdarvegleiðaranum at finna neyðuga fígging. Náttúruvísindadeildin hevur eina grundupphædd upp á 500 kr til uppgávuna.

Er talan um uppgávu, sum kemuruttanífrá, rindar viðkomandi stovnur ella fyrítøka fyri útreiðslurnar, ið standast av uppgávuni. Avtala um hetta verður gjørd við Náttúruvísindadeildina.

Uttanseturs vegleiðarar verða løntir sambært tímauppgerð.

Vegleiðing

Tað er uppgávan hjá ábyrgdarvegleiðaranum at finna møguligar medvegleiðarar og próvdómara. Vegleiðararnir, próvdómarin og tann lesandi finna í felag fram til, á hvørjum máli uppgávan skal skrivast.

Umframt fakligu vegleiðingina, skal vegleiðarin eisini vegleiða í, hvussu uppgávan skal setast upp, og hvussu tilvísingar skulu skipast.

Til uppgávuna verða tillutaðir upp til 20 tímar til vegleiðing. Tað talan er um fleiri vegleiðarar skulu teir 20 tímarnir deilast millum hesar. Eru uttanseturs vegleiðarar við verða teir løntir sbt AKF sáttmálanum.

Ritgerðin

Ritgerðin skal skipast sambært vanligum reglum í vísindaligum frágreiðingum og sambært skeiðslýsingini. Fremst í ritgerðini skal vera ein samandrættur av úrslitunum í uppgávuni. Samandrætturin skal vera skrivaður á føroyskum og enskum, og skrivaður soleiðis, at onnur enn fakfólk eisini kunna lesa hann. Samandrætturin skal í mesta lagi fylla hálva A-4 síðu.

Ritgerðin verður skrivað á máli, ið semja verður gjørd um millum ábyrgdarvegleiðara og lesandi, vanliga á føroyskum, einum øðrum norðurlendskum máli ella enskum.

Bacheloruppgáva

Vegleiðandi síðutal fyri ritgerðina er umleið 30 A4-síður, umframt fylgiskjøl. Har tvey fólk eru um ritgerðina, skal hon vera umleið 50 A4-síðir til støddar.

_12.09.2019_____
Dagfesting

_Helgi Poulsen_____
Undirskrift Lesandi

_12.09.2019_____
Dagfesting

_Jari í Hjøllum_____
Undirskrift Ábyrgdarvegleiðari

_12.09.2019_____
Dagfesting

_Kári Holm Johannesen_____
Undirskrift Uttanhýsisvegleiðari

_12.09.2019_____
Dagfesting

_Torbjørn Andreas Lisberg_____
Undirskrift Uttanhýsisvegleiðari

Avtalan verður váttað við undirskrift (omanfyri) ella gjøgnum teldupost til ábyrgdarlæraran fyri skeiðið og útbúgvingarleiðaran.

Abstract

User research performed at Statistics Faroe Islands shows that many users on a novice and mediocre level have a difficult time using the PxWeb interface. Similar findings have been found by other statistical offices, which are using the PxWeb statbank application. This is, therefore, a cross-national problem since users in several countries using PxWeb are facing the same usability issues.

This thesis describes a project on how to develop a functional prototype, which is a reactive UI for Px statbanks, with the software engineering process and project management methodologies.

The functional prototype improved the user experience immensely and gave the people behind PxWeb a new way of thinking on how to move PxWeb forward into the future.

Samandráttur

Brúkarakannningar av markamótinum hjá PxWeb, sum eru gjørdar í Føroyum, vísa at flestu brúkarar á byrjunar støði hava ringt við at nýta PxWeb. Líknandi úrslit eru funnin í øðrum hagstovum, tí er hetta ein altjóða trupulleiki.

Hendan serratgerðin lýsir eina verkætlan, har ein funktionel prototypa verður ment. Prototypan er eitt reaktivt nýtislumarkamót til Px hagtalsgrunnar og var ment við einari ritbúnaðarverkfrøðisligari tilgongd og einum verkætlanarleiðslu námsháttalagi.

Tann funktionella prototypan bøtti brúkara upplivingina ómetaliga nógv, og gav fólkunum aftanfýri PxWeb íblástur til, hvussu tey kunnu menna PxWeb framyvir.

Foreword

I met Kári Holm Johannesen,, head of the statistics at Statistics Faroe Islands, at the conference Kom og Fanga Okkum, that Ráð Teirra Lesandi organized on 02.04.19.

We talked about potential bachelor projects and he introduced me to this interesting project.

We met again on a later meeting and discussed in more detail about the project.

After the meeting I knew right away that this was going to be my bachelor project.

This project had so much potential and could completely redefine the standard PxWeb statbank.

Acknowledgements

I want to thank my family, girlfriend and everyone at Statistics Faroe Islands for being so supportive.

I want to give a special thanks to my supervisors for all the help they gave me during my time doing the project and for presenting the project at the annual Px Meeting 2019 - Armeina.

I also want to thank John Høj Andreassen for giving me the possibility to present the project at Mennarar Meetup on 24.10.19 organized by Mennarar.

At last I want to thank André Miede for providing this `LATEX`classic thesis **template**¹.

¹ URL: <https://bitbucket.org/amiede/classicthesis/wiki/Home>

Contents

Abstract	iii
Samandráttur	iii
Foreword	v
Acknowledgements	v
I FUNDAMENTALS	1
1 Introduction	3
1.1 Background of the Problem	3
1.2 Problem Statement	5
1.3 Addressing the Problem	6
1.4 Thesis Structure	6
1.5 Source Code and Live Demo	7
2 Related Work	9
3 Objective of the Project	11
3.1 functional prototype	11
3.2 Course of the project	12
3.2.1 Hierarchy of Objectives	13
3.2.2 Stakeholders analysis	14
3.3 Project Plan	17
4 Px	19
4.1 Px File	19
4.2 PxSuite	19
4.3 PxWeb	20
4.3.1 PxWeb API	20
5 Technology	25
5.1 JavaScript	25
5.2 React	26
5.2.1 TypeScript	27
5.2.2 VueJS and AngularJS	27
5.2.3 ASP.NET	28
6 The Software Engineering Approach	29
6.1 The Process Framework	29
6.1.1 Communication	29
6.1.2 Planning	30
6.1.3 Modeling	32
6.1.4 Construction	34
6.1.5 Deployment	34
6.2 Process Flow	35
6.3 Process Model	35
6.4 Agile	36
6.4.1 Agile Process Model	37

II	ARCHITECTURE	39
7	System Design	41
7.1	Main	42
7.1.1	Passing New Data From One Component To Another	43
7.1.2	Context	44
7.1.3	Routes	44
7.2	ListMenu	45
7.2.1	Network Error 429	45
7.2.2	Caching	46
7.2.3	Menu Tree JSON Object	46
7.2.4	react-simple-tree-menu	47
7.3	Selectors	47
7.4	TablesData	48
8	User Interface Design	49
8.1	Mobile Phones/Tables	51
9	Prototype Functionality	53
III	ANALYSIS	55
10	Presentations	57
10.1	Mennara Meetup	57
10.2	International Px Meeting 2019 - Armenia	57
11	Benefits of the Prototype	59
11.1	Removing the Three Steps	59
11.2	Reactive	59
11.3	Modern look	59
12	Analysis of the Success Criteria	61
12.1	Remove postback after user interaction	61
12.2	Make the UI Responsive	61
12.3	Using Modern Components	61
12.4	Reduce the Number of Clicks	62
12.5	Make the UI user-friendly	63
12.6	Finding the Right Data Faster	63
IV	SUMMARY	65
13	Findings	67
14	Conclusions	69
15	Future Work	71
V	APPENDIX	73
A	Appendix Px	75
A.1	A Query Example	75
A.2	Query Result	77
B	Appendix Agile Manifesto	79
C	Test Environment	81
C.1	Continuous Integration	81
C.2	End to End Testing	82

c.3 Unit Testing	82
BIBLIOGRAPHY	83

List of Figures

Figure 1	List	4
Figure 2	Selectors	4
Figure 3	Result	5
Figure 4	Sketch of the Functional Prototype	5
Figure 5	List	9
Figure 6	Selectors	9
Figure 7	Result	10
Figure 8	A tripartite of the objective	13
Figure 9	Hierarchy of Objectives: Objectives above the dotted line while deliveries below the dotted line.	14
Figure 10	Influence/Participation Matrix	16
Figure 11	Project Plan for Reactive UI for Px-Statbanks	17
Figure 12	PxWeb - Table: land_oyfj.px	23
Figure 13	Communication Link	25
Figure 14	Communication	30
Figure 15	Component Diagram	31
Figure 16	Sequence Diagram of Px UI	32
Figure 17	Sequence Diagram of Selectors	32
Figure 18	Activity Diagram of PxUI	33
Figure 19	Deployment Diagram	34
Figure 20	Evolutionary Process Flow	35
Figure 21	The Prototyping Paradigm	35
Figure 22	SCRUM	37
Figure 23	Class Diagram	41
Figure 24	With/Without Main component	42
Figure 25	Changing state	44
Figure 26	relationship between the three states	46
Figure 27	fo_vital_md.px	48
Figure 28	PxUI	49
Figure 29	PxUI Mobile	51
Figure 30	Px UI Functionality	53
Figure 31	Menu and Selectors	61
Figure 32	Result	62
Figure 33	Click Test	63
Figure 34	Test Driven Development	81

List of Tables

Table 1	Defined success criteria	14
Table 2	Advantages and Disadvantages of stakeholders	16

Listings

Listing 1	Database-ID	21
Listing 2	H2	21
Listing 3	LEVEL UO	22
Listing 4	LEVEL UO/UOo1	22
Listing 5	TABLE-ID	23
Listing 6	ListMenu	43
Listing 7	DbSelector in Main	43
Listing 8	DBSelecort	43
Listing 9	customPath	44
Listing 10	excat path	45
Listing 11	Menu Tree Example	46
Listing 12	TreeMenu	47
Listing 13	A Query Example	76

Acronyms

UI	User Interface
API	Application Programming Interface
UML	Unified Modeling Language
CSV	Comma Seperated Values
URL	Uniform Resource Locator
HTTP	Hypertext Transfer Protocol
ISO	the International Organization for Standardization
TDD	Test Driven Development

Part I

FUNDAMENTALS

1

Introduction

National statistical offices disseminate official statistics in statbanks where it is available to citizens and institutions via web interfaces.

All the statistical offices in the Nordic region and in several other countries around the world use an installation of the statbank application, PxWeb, to disseminate official statistics.

PxWeb is lead by Statistics Sweden and developed in cooperation with other national statistical offices, among them Statistics Norway and Statistics Finland.

1.1 BACKGROUND OF THE PROBLEM

User research performed at Statistics Faroe Islands shows that many users on a novice and mediocre level have a difficult time using the PxWeb interface. Similar findings have been found by other statistical offices, which are using the PxWeb statbank application. This is, therefore, a cross-national problem since users in several countries using PxWeb are facing the same usability issues.

One of the main challenges for the users across different sectors is to find the right data in the statbank. The user interaction tends to involve too many clicks to get data and when the data finally is returned from the application it often is not what user is expecting. This often results in users giving up in finding the right data.

PxWeb also uses postback¹ reloads the page content after each interaction, for example when selecting a menu item. This makes the user experience very awkward.

The reason behind the postback is that PxWeb is a multi-page application that consists of several pages with static information.

The user journey in the statbank starts by navigating the folder structure and then finding the right table [Figure 1](#).

When the user has found a relevant table the user has to choose multiple categories from several statistical variables [Figure 2](#) and then click the submit button to get the result [Figure 4](#). If the data is not what the user is expecting the user has to go over this procedure once again.

¹ A postback is an HTTP POST to the same page that the form is on. In other words, the contents of the form are POSTed back to the same URL as the form.
Postbacks are most commonly discussed in relation to JSF and ASP or ASP.NET.

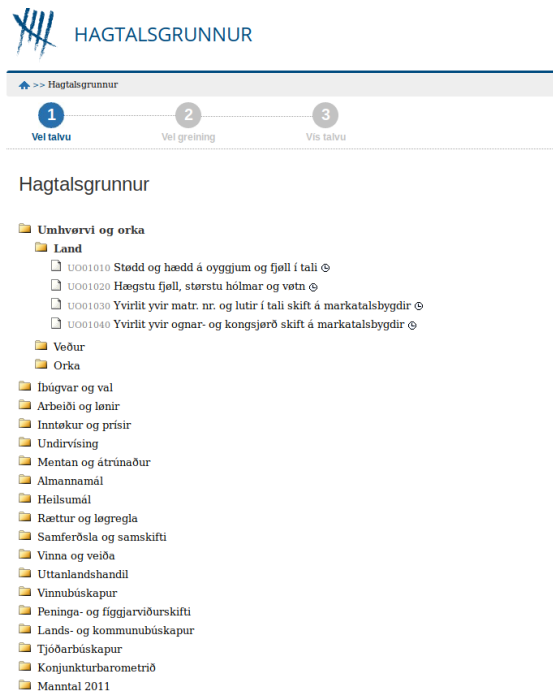


Figure 1: List

The screenshot shows the Hagtalsgrunnur web application with the navigation steps: 1. Vel talvu, 2. Vel greining, and 3. Vis talvu. The main content area is titled "U001010 Stødd og hædd á oyggjum og fjøll í tali". Below the title, there is a section for "Eind : tal" with two tabs: "Vel greining" and "Um talvuna". The "Vel greining" tab is selected. Below the tabs, there is a text box with the instruction: "Vel greiningar og trýst síðani á 'Halt áfram'. Soleiðis velur tú Har greiningin er merkt við * skal minst eitt virði veljast". Below this, there are two columns of selection options. The left column is titled "oyggj/hægsta stað *" and the right column is titled "stødd, tal á fjøllum og fólkatál *". Both columns have a list of options and a "Leita" button. The left column options are: Tilsamans 19 Valt 0, Tilsamans, Streymoy, Eysturoy, Vágar, Suðuroy, and Sandoy. The right column options are: Tilsamans 7 Valt 0, Oyggj í km², Oyggj, hólmar o.a. í km², Oyggj, hólmar o.a. í tali, Hædd í metrum, Fjell í tali, and Fólkatál 01.01.2019. Below the selection options, there is a text box with the instruction: "Tal á valdum puntum er: 0 (mest loyvda talið er 100.000) Skuggjavisning er avmarkað til 1.000 røð og 50 teigar". At the bottom, there is a button labeled "Talva - Uppseting 1" and a button labeled "Halt áfram".

Figure 2: Selectors

The screenshot shows the Hagtalssgrunnur web application. At the top, there's a navigation bar with three steps: 1. Vel talvu, 2. Vel greining, and 3. Vis talvu. Below this, there's a search bar with 'Vis talvuna' and 'Um talvuna' buttons. There are also dropdown menus for 'Rættu og rokna', 'Geym talvu sum', and 'Talva - Uppseting 1'. Below these are buttons for 'Rættu', 'Geym', and 'Rökna'. There are also buttons for 'Talvasetningar' and 'Geym fylgispurning'. The main content area shows a table with the title 'U001010 Stædd og hædd á byggjum og fjöllum í tali'. The table has two columns: 'Oyggj, hölmur o.s. í km²' and 'Tíðarnar'. The data row shows '1.396,4'.

Figure 3: Result

In ISO 9241² usability is defined as the effectiveness, efficiency and satisfaction with which specified users achieve specified goals in particular environments. In this case this means users finding the right data in the statbank.

Effectiveness is the accuracy and completeness with which the users can achieve specified data in PxWeb. Efficiency is the resources expended such as time or number of clicks in relation to the accuracy and completeness of getting the right data. Satisfaction is the comfort and acceptability of the work system to its users and other people affected by its use.

From a usability perspective it seems that there is a lack of effectiveness and efficiency in the PxWeb application. This requires a new approach to get and display the data from the statbank application.

1.2 PROBLEM STATEMENT

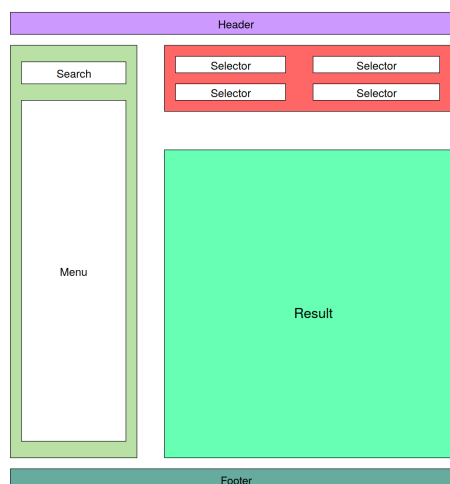


Figure 4: Sketch of the Functional Prototype

In this thesis I aim to develop a functional prototype that is a reactive UI for Px statbanks. A functional prototype is a prototype that behaves and act like a live product.

² URL: <https://www.iso.org/search.html?q=ISO%209241>

This prototype should improve the effectiveness and efficiency of the interface, and exploit and utilize the API in the PxWeb statbank. I aim to establish a direct communication between interface and the API in order to get data, manipulate data and displaying data instantly.

In this approach the user will instantly be shown data when the user has chosen a category from a statistical variable. This differs significantly from the current approach where the user has to choose multiple categories in several statistical variables and then click the submit button to get the data.

This new approach significantly reduces the number of clicks and the time used to get data.

It will make UI more intuitive and easier to use.

1.3 ADDRESSING THE PROBLEM

The problem will be addressed by using the right technology and the software engineering approach such as the software methodology³, Agile and the five framework activities⁴.

Project management techniques will also be used to give a clearer understanding of the project and its purpose.

1.4 THESIS STRUCTURE

This thesis is divided into three parts. Fundamentals, Architecture, Analysis and Summary.

Fundamentals contains six chapters. An introduction about the project, research about related work, an explanation about objective of the project, what kind of technology is used and why it's used and an explanation about the software engineering approach used in this project.

Architecture contains three chapters. An explanation about the reason behind the system design of the prototype, the reason behind the UI design and simplified clarification about the prototype functionality.

Analysis contains three chapters. A telling about the two presentations performed of the prototype, an analysis of the prototype and the benefits of the prototype.

Summary contains three chapters. The findings, the conclusion and possible future work.

The Test Environment chapter [Appendix C](#) was intentionally moved to appendix, since I didn't create a testing environment for the prototype, given the pressure of finishing the prototype on time. The chapter explains what kind of testing environment I wanted to create.

³ A software methodology is a set of related activities that leads to the production of software

⁴ communication, planning, modeling, construction, deployment

1.5 SOURCE CODE AND LIVE DEMO

The source code and a live demo of the functional prototype can be found here.

Source Code: **PxUI**^a

Live Demo: **DEMO**^b

^a URL: <https://github.com/hagvarp/pxui>

^b URL: <https://hagvarp.github.io/pxui/#/>

2

Related Work

Before starting the project I wanted to see if any of the statistical offices had done something similar as this project aimed to do. After searching through most of the European statbanks, I could not find any statistical office that did not use the standard PxWeb template.

Some statbanks, like the Danish statbank have changed the style, but the functionality is still the same as the standard PxWeb where the user has to go through the three steps/pages to get a result.

The screenshot shows the StatBank Denmark interface. On the left is a 'SUBJECTS' sidebar with categories like Population and elections, Living conditions, Education and knowledge (selected), Culture and National Church, Labour, income and wealth, Prices and consumption, National accounts and government finances, Money and credit market, External economy, Business sector in general, Business sectors, and Geography, environment and energy. The main area is titled 'EDUCATION AND KNOWLEDGE' and contains a 'show all...' link and a list of data series under 'Population by status of education'. The series include 'Educational attainment' (with sub-series HFUDD010, HFUDD015, HFUDD020, and KMSTA007) and 'Upper secondary education'. The 'Educational attainment' series are described as: HFUDD010: Educational attainment (15-69 years) by region, ancestry, highest education completed, age and sex (2006-2019); HFUDD015: Educational attainment (15-69 years) by region, highest education completed, socioeconomic status, industry, age and sex (2009-2018); HFUDD020: Educational attainment (15-29 years) by region, ancestry, highest education completed, receiving education, age and sex (2006-2019); KMSTA007: Population 1. January (15 years+) by parish, member of the National Church and highest education completed (2016-2019). There is also an 'Archive' link.

Figure 5: List

The screenshot shows the 'SELECTORS' section of the StatBank Denmark interface. It displays four selector panels: 'REGION (10)', 'ANCESTRY (1)', 'HIGHEST EDUCATION COMPLETED (1)', and 'AGE (1)'. Each panel has a 'More options...' link and a list of available values. The 'REGION (10)' panel lists regions like All Denmark, Region of Southern Denmark, Capital Region, etc. The 'ANCESTRY (1)' panel lists 'Total' and 'Persons of Danish origin (descendants)'. The 'HIGHEST EDUCATION COMPLETED (1)' panel lists various education levels from 'Total' to 'The technology area, mechanical engineering and production (T8T)'. The 'AGE (1)' panel lists age groups from 'Age, total' to '65-69 years'. At the bottom, there is a 'Number of selected data sets for the table: 1' and a 'Select max. 10000' limit.

Figure 6: Selectors

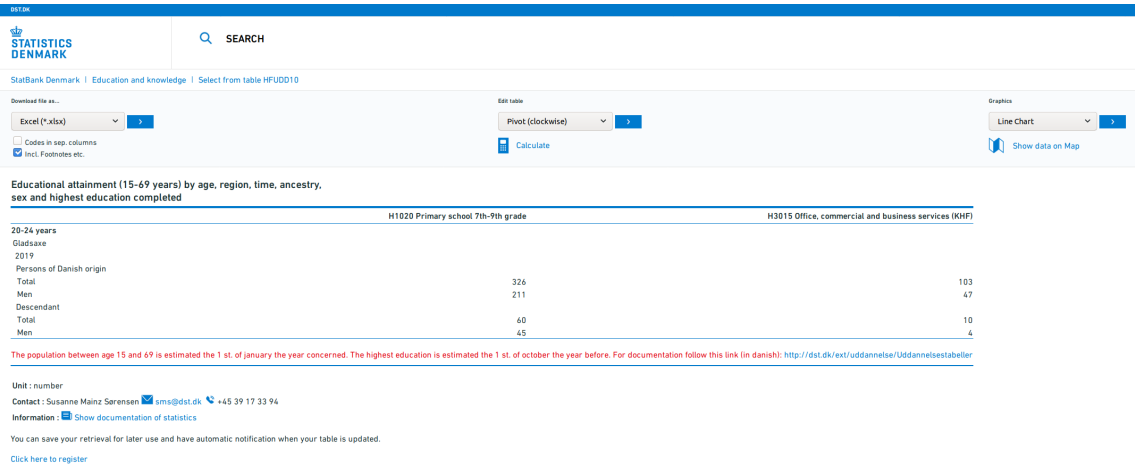


Figure 7: Result

This made it clear that I could not use any related work from other statistical offices as a reference and also making this project unique, concerning Px, and demanding.

3

Objective of the Project

The main objective is to create a functional prototype that is a reactive UI for Px statbanks. I want to make this clear, since there are many differences between a prototype and a final product.

By definition in software technology, the term prototype is a working example through which a new model or a new version of an existing product can be derived. In this case a new version of Px-statbank.

The main reason for creating a prototype that is a reactive UI for Px statbanks is to evaluate and test the concept. Is it possible to create a reactive UI for Px-statbank? If so, how is the user experience?.

Answering questions like these, we can determine with other statistical offices, if a final product should be created or not.

3.1 FUNCTIONAL PROTOTYPE

Cambridge dictionary:

prototype (proh-tuh-tahyp) *Noun* • the first example of something, such as a machine or other industrial product, from which all later forms are developed.

The reasoning of making a prototype and not a finished product is that the project is a proof of concept¹.

There are many good reasons to make a prototype. Four main points are [311, p. 17]:

- To understand.
- To communicate.
- To test.
- To improve.

Prototyping is a great way to understand the problem with PxWeb. It's also great way to illuminate alternative problems that should be solved instead.

In addition, prototyping gives the possibilities to solve the problems in different ways.

¹ A proof of concept (POC) is a demonstration, the purpose of which is to verify that certain concepts or theories have the potential for real-world application.

Prototypes are also a great communication tool for making ideas visible and testable. With prototyping it's now possible to show the stakeholders in this project an specific example. Thus making the communication easier, because now all stakeholders can see the idea, instead of just having it in their head. This puts everyone on the same page where they can give feedback on the approach to solve the problem.

Prototypes are also great for testing, so that ideas can be improved up on form user feedback.

The book *Sketching User Experiences* [3, p. 140] gives also a great example of the sketch to prototype continuum²

SKETCH	PROTOTYPE
Evocative	—> Didactic
Suggest	—> Describe
Explore	—> Refine
Question	—> Answer
Propose	—> Test
Provoke	—> Resolve
Tentative	—> Specific
Noncommittal	—> Depiction

The arrows emphasize that this is a continuum, not an either/ or proposition.

It show how a sketch raises a question: Is it possible to create a reactive UI for Px statbanks? the prototype delivers an answer.

There are many different kinds of prototypes, from paper prototypes to evolutionary prototypes. A list of the most common prototypes can be found at **22 Types of prototype**³.

The decision for making a functional prototype is that we want the functionality to imitate PxWeb as close as possible and the prototype to be fully usable. Meaning that it shall use the API in the same way as PxWeb and get the same results, but more effectively and efficiently.

Functional prototype only focuses on the functionality and not the design itself. Like how the UI is styled[14].

3.2 COURSE OF THE PROJECT

To make the objective of the project as clear, realistic and acceptable as possible, project management techniques will be used to give everyone involved a clear picture of what the objective is and how to reach the end line in the best and safest way possible.

² a coherent whole characterized as a collection, sequence, or progression of values or elements varying by minute degrees.

³ URL: <https://simplicable.com/new/prototypes>

To set the course of the project, the objective will be split up in three parts[12, p. 89].

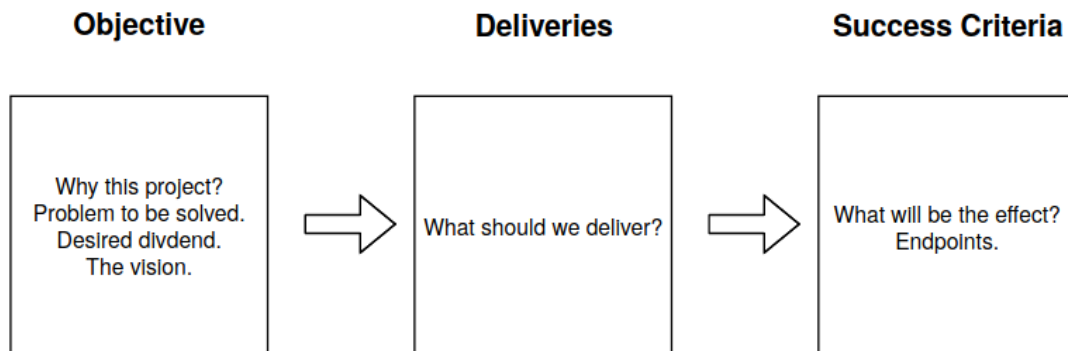


Figure 8: A tripartite of the objective

I will show a clearer picture of this in [Section 3.2.1](#)

3.2.1 Hierarchy of Objectives

The hierarchy of objectives is great tool that gives a clear connection between what I want to achieve and what I shall deliver. By doing it together with Statistics Faroe Islands, we can exclude doubts and misunderstandings about what the project is about.

SMART[12, p. 109] was used as a guide for setting the objectives in HOO⁴, SMART is a acronym used to guide the development of measurable goals.

- **Specific**
Deliveries must be specifically formulated so that they can be used to plan according to.
- **Measurable**
Deliveries that can't be measured aren't in reality deliveries. It is important to be able to measure the delivery so that it is possible to follow up on it and see how far we have come.
- **Acceptable**
Deliveries should be accepted by everyone involved.
- **Realistic (but ambitious)**
Deliveries should be realistic, so that a solution can be seen. On the other hand the deliveries should be ambitious, so that the possibility for failure is present.
- **Time Scheduled?**
Deliveries are time Scheduled so that they can be delivered on time.

HOO is created by asking How to create a reactive UI for Px-statbanks? By optimizing the UI. How to optimize the UI? By following this step we get a tree. So by asking how, we go down the tree and asking why, we go up the tree.

⁴ Hierarchy of objectives

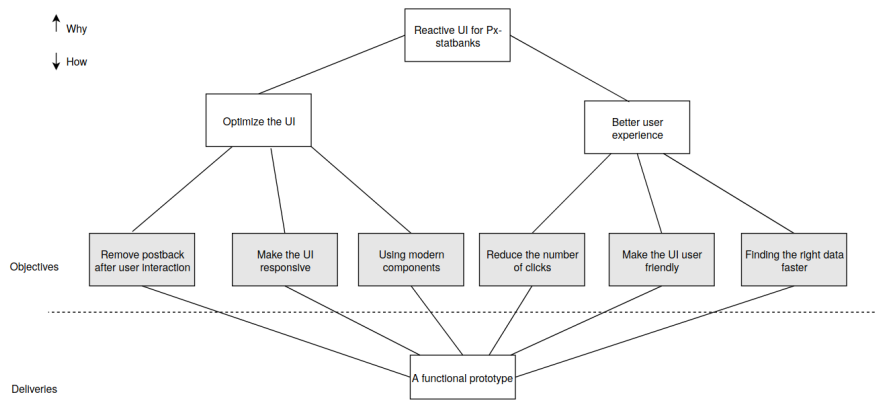


Figure 9: Hierarchy of Objectives: Objectives above the dotted line while deliveries below the dotted line.

By looking at [Figure 9](#) we can establish what the objectives and deliveries are:

The project is being carried out in order to:

- Remove postback after user interaction.
- Make the UI responsive.
- Using modern components.
- Reduce the number of clicks.
- Make the UI user-friendly
- Finding the right data faster.

The deliveries are:

- A New functional prototype

We can now define the success criteria for the project:

Objective	Success Criteria
Remove postback after user interaction	Using technology that can create single-page applications
Make the UI responsive	The user should see instant result of his actions
Using modern components	The web application shall only contain modern components
Reduce the number of clicks	The number of clicks shall be reduced by 40%
Make the UI user-friendly	User feedback of the prototype should be 75% positive
Finding the right data faster	A search functionality shall be implemented

Table 1: Defined success criteria

3.2.2 Stakeholders analysis

“Without stakeholders no project.”[2 12, p. 114] So specific is the book “*Power i projekter & portefølje*” when it comes to stakeholders.

The stakeholders are everyone that are involved or get affected by the project or the project results. This means that some stakeholders will be more than

willing to help and support the project, while others, that feel that the project may give a negative affect on them, may work against it.

By ensuring the success of the project, managing stakeholders becomes a very important factor, especially when it comes to the important stakeholders, because without their satisfaction the project won't succeed.

In the article[18] Rachel Thompson states that there are four key benefits by doing the stakeholders analysis:

- Getting your Project into shape: *You can use the opinions of your most powerful stakeholders to help define your projects at an early stage.*
- Winning Resources: *Gaining support from powerful stakeholders can help you to win more resources, such as people, time or money.*
- Building Understanding: *By communicating with your stakeholders early and often, you can ensure that they fully grasp what you're doing and understand the benefits of your project.*
- Getting Ahead of the Game: *Understanding your stakeholders means that you can anticipate and predict their reactions to your project as it develops.*

The stakeholders analysis is done by following four steps[12, p. 127]:

1. Who are they? (Identifying stakeholders)
2. Who is most important? (stakeholder prioritization)
3. What do they wish? (experienced dividends/benefits for stakeholders)
4. What do we do? (stakeholder management strategy)

3.2.2.1 Step 1

Who are they?:

- Statistics Faroe Islands
- Other statistical offices
- Users

The main stakeholder in this project is Statistics Faroe Islands. Statistics Faroe Islands has the highest interest in this project, since they are the project owners and want to give their users a better user experience when working with PxWeb.

Other statistical offices have also an interest in this project, since the outcome of this project can lead to new changes to PxWeb.

The users are also stakeholders since the prototype is being developed to improve their user experience and the way to find the right results.

3.2.2.2 Step 2

Based on the assessment of influence and active participation of the project, we can split the stakeholders into 4 different categories. This will give us a clearer image of the important stakeholders and what kind of attention needs to be applied to each one of them.

Stakeholders can be [Figure 10](#) into four different categories, depending on their influence and participation in the project.

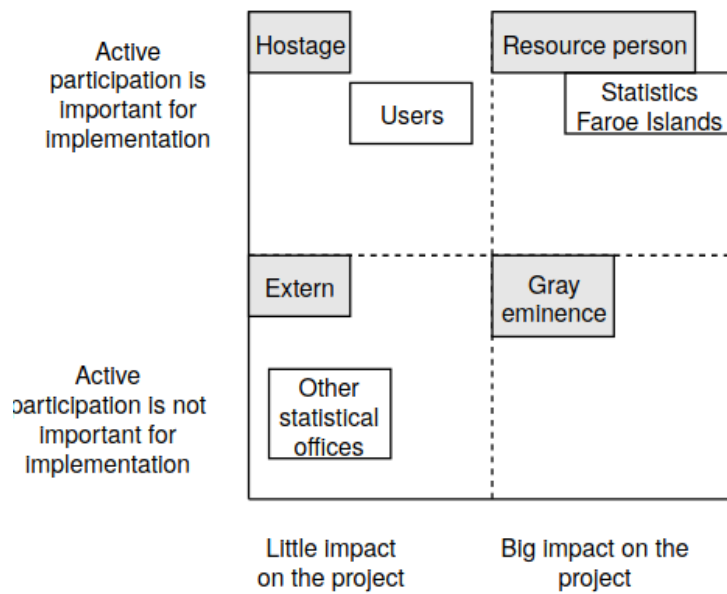


Figure 10: Influence/Participation Matrix

- *Hostage* monitored closely.
- *Resource person* Manage with maximum effort.
- *Extern* Monitor with minimal effort.
- *Gray eminence* keep informed.

In [Figure 10](#) we can see that Statistics Faroe Islands needs special attention, since it is in the resource person category. The users are also very important, since they will be the most affected of the prototype. So gathering user feedback is a key factor in developing this prototype.

Even though this prototype is meant for all statistical offices. Other statistical offices will have little importance and impact on how the prototype will be developed, since there is no collaboration between Statistics Faroe Islands and any other statistical office, concerning this project.

3.2.2.3 Step 3

Experience of advantages and disadvantages of stakeholders.

Stakeholder	Experienced Advantages	Experienced disadvantages	Overall assessment of stakeholder Contribution/position	Managing the stakeholder
Statistics Faroe Islands	The Faroese statbank will be set to modern standards. Less user complaints.	Learn new procedure to use the new statbank.	Very positive over how new technology can be used to make the statbank more user-friendly.	continuously co-operation about the project and updates about how the project is going.
Users	Faster and easier to get the data they are looking for	Learning a new system to get the right data	Very positive about the new prototype, since users have stated their concern and difficulties using the old PxWeb application	Getting user feedback and delivering a easy to understand tutorial on how to use the prototype
Other statistical offices	Their statbank will be set to modern standards	Can be hard to convince them that this is the right direction to go	Positive and interested in the development of the prototype	Show the prototype and it's possibilities in international conferences

Table 2: Advantages and Disadvantages of stakeholders

3.2.2.4 Step 4

A strategy to handle the stakeholders is now set by bringing up the stakeholders analysis with each meeting between myself and my supervisors. This is to see if any of them has changed position or if there are potentially new stakeholders, that were overlooked, as the project progresses.

3.3 PROJECT PLAN

A documented plan is crucial to the success of research projects. Project plans help to stay organized and to keep focus on the project goals. A project plan also improves communication with stakeholders.

With a clear project plan, stakeholders can get a clear picture of what needs to be done to make the project a success. The project plan also a good indicator to show the stakeholders where the project is and where it is heading, this gives them the possibilities to give feedback and make changes before it's too late.

There are many ways to do project plans, some more complicated than others. Many depending on how big the project is and how many people are involved.

Since this isn't a big project involving many people, the project plan is simplified.

Keep in mind that this is an iterative project plan and not a linear project plan.

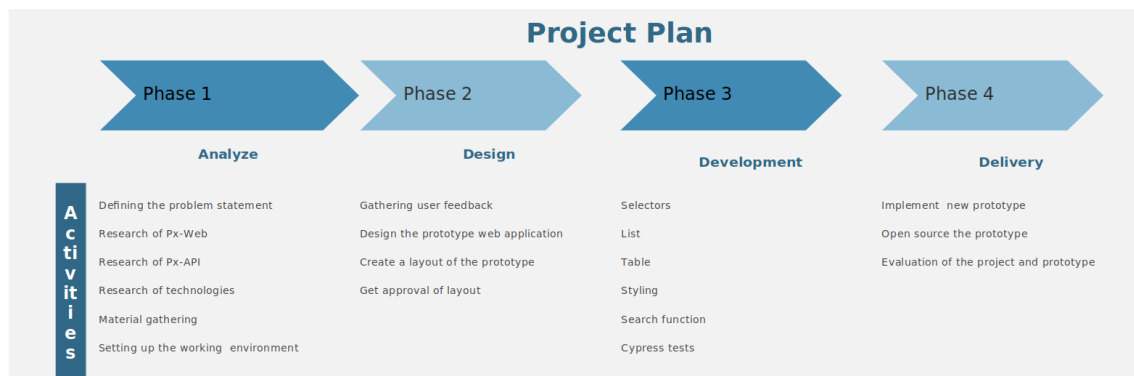


Figure 11: Project Plan for Reactive UI for Px-Statbanks

Figure 11 shows that the project has 4 phases. Analyze, Design, Development and Delivery. With the phases come different activities that need to be performed before being able to the next phase. Each phase and activity has normally a time course for how long it should last, but since this project has many unknown factors for how long each phase and activity should last, they are left out. This also gives more freedom to focus on the whole concept instead of each specific task.

Phase 1 is the starting point. It contains activities that are related to research, material gathering and analysis.

Phase 2 is the initial design of the single page web application. A simplified sketch of component placement.

Phase 3 is the development phase. This phase contains the coding of the components.

Phase 4 is the phase where the evaluation and implementation takes place.

4

Px

Most people outside statistical offices have never heard about Px files.

To get a better understanding about what these Px files are, and what they are used for, I studied at Statistics Sweden's [documentation](#)¹ about PxWeb. Statistics Sweden is the main developer of software that uses Px.

Px is a standard format for statistical files used by a large number of statistical offices[4].

According to the 27th international Px-meeting in Bilbao, Spain 2018, there are 40 countries, 63 international organisations and 21 Swedish organisations that use PxWeb².

4.1 PX FILE

The Px file format consists of a number of keywords. These can be either mandatory or optional. Px files are used in the PX-programs.[4]

Px files are basically CSV files with a space as a delimiter and a whole bunch of metadata at the top[2].

To get a better understanding of the Px file format, I recommend reading *PX-file format specification*[7]

4.2 PXSUITE

The Px Suite consist of a collection of programs developed by Statistics Sweden and Statistics Finland. They offer diverse possibilities to distribute and edit even large statistical tables, and to change structures, combine tables, make calculations and convert into other file formats.[9]

- PxWeb
- PxWin
- PxEdit
- PxJob

My focus will be on PxWeb, since it is the program used for publishing statistics on the web. I will not focus on the code itself, but rather how it looks and feels for the general user.

¹ URL: <https://www.scb.se/en/services/statistical-programs-for-px-files/>

² [Meeting introduction](#)

4.3 PXWEB

PxWeb, formerly known as PC-Axis, is a free web-based table distribution system for statistics producers. It is used to disseminate official statistics.

PxWeb delivers a UI and a API. As stated in [Chapter 1](#) and [Chapter 3](#) chapters, where the UI is the main problem.

Instead of doing work on the PxWeb UI the goal is to completely redefine UI with modern technology. To be able to do that, a clear understanding of the PxWeb API must be established.

4.3.1 PxWeb API

The best documentation about PxWeb API and how to use the API can be found on <http://pxnet2.stat.fi/api1.html>. This documentation shows everything about the API and how to use it. In 2014, Statistics Sweden also made a description about how to use their API *PC-Axis API description*[\[8\]](#)

4.3.1.1 Accessing the API

The way to access the database is done by using URL. The URL is constructed with different components[\[15 8, pp. 14-16\]](#):

PXWEB/API-NAME/API-VERSION/LANGUAGE/DATABASE-ID/LEVELS/TABLE-ID

- API-NAME *defines the root part of the API.*
- API-VERSION *defines the version of the API.*
- LANGUAGE *defines the language of the data retrieval.*
- DATABASE-ID *defines the database where the statistical cubes are stored.*
- LEVELS *defines zero or more levels that show the various divisions in the database.*
- TABLE-ID *defines the identity of the table.*

Here is an example of how the URL components are put together for the Faroese statbank:

- API-NAME: api
- API-VERSION: v1
- LANGUAGE: en

The starting URL will be: <https://statbank.hagstova.fo/api/v1/en>

The response from this url is [Listing 1](#):

Listing 1: Database-ID

```

1 [
2   {
3     "dbid": "H2",
4     "text": "H2"
5   }
6 ]

```

This JSON object gives us the DATABASE-ID:

- DATABASE-ID: H2

Some statbanks, like the Icelandic statbank contain many databases. The Faroese statbank has only H2.

When we add H2 to the starting URL: <https://statbank.hagstova.fo/api/v1/fo/H2> we get the LEVELS:

Listing 2: H2

```

1 [
2   {
3     "id": "U0",
4     "type": "l",
5     "text": "Environment and energy"
6   },
7   {
8     "id": "IB",
9     "type": "l",
10    "text": "Population and elections\n"
11  },...
12 ],

```

[Listing 2](#) is the first menu items. If an object has the type: "l", it means that the object contains a level.

We select the first menu item, that contains a level, and add the ID to the URL: <https://statbank.hagstova.fo/api/v1/fo/H2/UO>

- LEVEL: UO

This will give us another array of objects that all contain level [Listing 3](#):

Listing 3: LEVEL UO

```
1 [
2   {
3     "id": "U001",
4     "type": "l",
5     "text": "Geography"
6   },
7   ...
8 ]
```

As before, we add the first objects ID to the URL:

<https://statbank.hagstova.fo/api/v1/fo/H2/UO/UO01>

- LEVEL: UO/UO01

Now we reached the TABLE-ID. We know this because, in this case, all objects contain type: "t" [Listing 4](#).

Listing 4: LEVEL UO/UO01

```
1 [
2   {
3     "id": "land_oyfj.px",
4     "type": "t",
5     "text": "U001010 Highest peaks and no. of mountains by island",
6     "updated": "2019-11-01T14:14:48"
7   },
8   ...
9 ]
```

- TABLE-ID: land_oyfj.px

So now we got:

- API-NAME: api
- API-VERSION: v1
- LANGUAGE: en
- DATABASE-ID: H2
- LEVEL: UO/UO01
- TABLE-ID: land_oyfj.px

https://statbank.hagstova.fo/api/v1/fo/H2/UO/UO01/land_oyfj.px

Listing 5: TABLE-ID

```

1 {
2   "title": "U001010 Highest peaks and no. of mountains by island",
3   "variables": [
4     {
5       "code": "island/highest mountain",
6       "text": "island/highest mountain",
7       "values": [
8         "999",
9         ...
10      ],
11      "valueTexts": [
12        "Total",
13        ...
14      ]
15    },
16    {
17      "code": "area in km2, no. of mountains and population",
18      "text": "area in km2, no. of mountains and population",
19      "values": [
20        "001",
21        ...
22      ],
23      "valueTexts": [
24        "Island in km2",
25        ...
26      ]
27    }
28  ]
29 }

```

Now we reached the data for building the selectors. We can see on [Listing 5](#), that there are two selectors, since there are two objects.

Statistical Database

- Environment and energy
 - Geography
 - U001010 Highest peaks and no. of mountains by island
 - U001020 Major mountains, islets and lakes
 - U001030 Structure of number of land registrations and lots by the old settlements
 - U001040 Structure of land ownership, private and public by the old settlements
 - Climate
 - Energy
- Population and elections
- Labour and wages
- Prices and income
- Education
- Culture and religion
- Social affairs
- Health
- Justice and police
- Transport and communication
- Economic activities
- International trade
- Business statistics
- Money and credit market
- Public finance
- National accounts
- Business survey
- Census 2011
- Nordic Statistics
- Nordic Health and Social Statistics

U001010 Highest peaks and no. of mountains by island

Unit: number

Select variable About table

Mark your selections and choose between table on screen and file format. Marking tips
For variables marked * you need to select at least one value

island/highest mountain *	area in km², no. of mountains and population *
Total 19 Selected 19	Total 7 Selected 7
<div style="background-color: #f0f0f0; padding: 2px;">Total</div> <div style="background-color: #f0f0f0; padding: 2px;">Streymoy</div> <div style="background-color: #f0f0f0; padding: 2px;">Bjørnøy</div> <div style="background-color: #f0f0f0; padding: 2px;">Vágar</div> <div style="background-color: #f0f0f0; padding: 2px;">Suðuroy</div> <div style="background-color: #f0f0f0; padding: 2px;">Sandoi</div>	<div style="background-color: #f0f0f0; padding: 2px;">Island in km²</div> <div style="background-color: #f0f0f0; padding: 2px;">Island, islets etc. in km²</div> <div style="background-color: #f0f0f0; padding: 2px;">Island, islets etc. in number</div> <div style="background-color: #f0f0f0; padding: 2px;">Altitude in metres</div> <div style="background-color: #f0f0f0; padding: 2px;">No. of mountains</div> <div style="background-color: #f0f0f0; padding: 2px;">Population 01.01.2015</div>
Search <input type="text"/>	Search <input type="text"/>
<input type="radio"/> Beginning of word	<input type="radio"/> Beginning of word

(a) LEVEL

(b) TABLE-ID

Figure 12: PxWeb - Table: land_oyfj.px

In short: [Figure 12a](#) - DATABASE-ID gives us the main menu items and LEVEL gives us the sub folders and tables.

[Figure 12b](#) - TABLE-ID gives us the selectors of the specific table selected. In this case, land_oyfj.px.

4.3.1.2 *Getting Data*

To get data from a specific table, the HTTP POST method is used. The POST method is used to send data to a server to create/update a resource.

To show an example of this is:

URL: http://statbank.hagstova.fo/api/v1/fo/H2/IB/IB01/fo_vital_md.px
Query: [Section A.1](#)
Result: [Section A.2](#)

4.3.1.3 *PxWeb response*

The response of the POST request can be delivered in many formats:

- Px
- JSON
- JSON-stat
- XLSX
- CSV

All depending on what kind of response format you set in the POST request. In [Section 4.3.1.2](#) example the response format is set to textit"px".

5

Technology

One of the main goals of this project is to establish a communication link between the back-end REST API served by statbank application and the new front-end prototype UI.

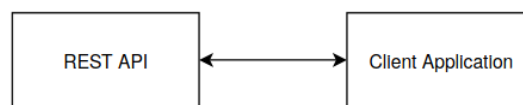


Figure 13: Communication Link

This means that the technology used, has to be a solid front-end technology that communicates well with REST API's. Other than that, the technology chosen has to be accessible for a variety of users since the statbank is intended for the public.

The technology has to be prototype friendly. Meaning it should be easy to work with and fast to create results.

Code maintenance is also an important factor in picking the right technology since this prototype is intended to all statistical offices.

Many technologies have been considered relevant to the project's web interface prototype.

5.1 JAVASCRIPT

JavaScript¹ with related advanced JavaScript frameworks was chosen as the main programming language. JavaScript is one of the worlds most used front-end programming language² supported and implemented in all modern browsers with the support of new versions of the language.

Addition to this, **ReactJS**³ library will be used for building the UI. React is developed and maintained by Facebook.

One advantage of using JavaScript in this project is that JavaScript is not compiled and therefore can be run immediately within the client-side browser. Not only is it an advantage for the users but makes it easy for other statistical offices to implement and try out the prototype UI.

¹ URL: <https://www.javascript.com/>

² **Top 10 programming languages used in web development**

³ URL: <https://reactjs.org/>

The data from the PxWeb API is served in **JSON**⁴. JavaScript is highly compatible with JSON since the syntax of JavaScript Object Notation is based on JavaScript object syntax. It consists of a metadata part and a data part.

Metadata is structured in a hierarchical node tree, where each node contains information about subnodes that are below it in the tree or, if the nodes are at the bottom of the tree structure, the data referenced by the node as well as what dimensions are available for the data at that subnode.

Other programming languages were considered, among them **TypeScript**⁵ and **ASP.NET**⁶.

Note, that it is highly probable that for a finished product another technology would be used.

5.2 REACT

There are many good reasons to choose the React library for this prototype.

React uses a virtual DOM⁷. This means that the actual DOM doesn't get manipulated directly. Instead the virtual representation gets manipulated. React takes care of changing the browser's DOM.

The benefits of using a virtual DOM is that it makes it easier to keep track of changes, since it's difficult to keep track of current and prior state of the actual DOM to manipulate it into a form needed.

It's also faster since modifying the actual DOM is costly and can cause poor performance. On the other hand React's virtual DOM comes with important performance optimizations that makes it very fast.

The virtual DOM uses efficient diffing algorithms, in order to know what changed and updates subtrees of the DOM simultaneously. It also batch updates the DOM.[1, pp. 135-136]

The other reasons are:

- Great for prototyping.
- React is specifically designed to create single page applications⁸.
- It's easy to create interactive UIs.
- React will efficiently update and render just the right components when data changes.
- React component based and allows to write reusable components where each component manages their own state.
- It's easy to setup and debug.

4 URL: <https://json.org/>

5 URL: <https://www.typescriptlang.org/>

6 URL: <https://dotnet.microsoft.com/apps/aspnet>

7 A tree of JavaScript objects that represent the actual DOM

8 A single-page application is a web application or web site that interacts with the user by dynamically rewriting the current page rather than loading entire new pages from a server.

- React has a huge online community.

From personal experience, I also found React easy to learn, compared to other frameworks, so it shouldn't be difficult for another front-end developer to understand the code after learning the basics of react.

5.2.1 TypeScript

TypeScript is a strongly typed, object oriented, compiled language. It is both a language and a set of tools.

TypeScript is a typed super set of JavaScript compiled to JavaScript. In other words, TypeScript is JavaScript plus some additional features and works well with React[19].

The reason for not using TypeScript is that it has a steeper learning curve than JavaScript. It also contains new syntax and can sometimes be a little too complicated to use properly.

With the short time of creating this prototype, I didn't feel I had the time to learn TypeScript and use it to its full potential.

5.2.2 VueJS and AngularJS

VueJS⁹ and **AngularJS**¹⁰ are JavaScript frameworks, unlike React that is a library.

I have no working experience with Vue and Angular, so I had to do some research to figure out if I should use Vue or Angular instead of React.

Their homepages gave a lot of information about each framework, but to get a comparison I read *Angular vs React vs Vue: Which Framework to Choose in 2020*[5] and *Angular vs React vs Vue: Which is the Best Choice for 2019?*[17].

What I gathered from the research was that Angular has a steep learning curve and a good choice for companies with large teams and developers who already use TypeScript.

Vue on the other hand, is relative new and without the backing of a major company like React has with Facebook. It is though simple to learn and flexible.

To end the comparison I cite from the article *Angular vs React vs Vue: Which Framework to Choose in 2020*[5]:

"React looks like a good choice for someone getting started with front-end JavaScript frameworks, startups and developers who like some flexibility. The ability to integrate with other frameworks seamlessly gives it a great advantage for those who would like some flexibility in their code."

I think this fits perfectly for this project and my experience with JavaScript and front-end.

⁹ URL: <https://vuejs.org/>

¹⁰ URL: <https://angularjs.org/>

5.2.3 *ASP.NET*

ASP.NET is a framework, that requires installation and a server, for running web applications. Applications that run on the server are used for processing data that you don't want to user to have access to.

Since this is a client-side prototype project, other technologies, like React that run out of the box, are better suited than ASP.NET.

6

The Software Engineering Approach

To obtain an economically, reliable, and work efficient prototype, the is applied in this project.

Software engineering encompasses a process, a collection of methods (practice), and an array of tools that allow professionals to build high-quality computer software.

According to IEE93a, software engineering is defined as:

(1) The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software. (2) The study of approaches as in (1).[\[16, p. 15\]](#):

6.1 THE PROCESS FRAMEWORK

This prototype is built on the Process Framework that is a collection of five encompassing activities, with the intent to deliver a product in a timely manner with sufficient quality.

- Communication
- Planning
- Modeling
- Construction
- Deployment

6.1.1 *Communication*

Before starting, I held meetings with my supervisors to figure out the objectives and requirements of the project and what kind of technology suited best for the project [Chapter 5](#).

We decided that JavaScript with ReactJS library was the best approach to make this prototype a reality.

In short, JavaScript was picked since is great for front-end projects and ReactJS since it's excellent for building UI specifically for single page applications.

Figure 14a show the minimum requirements that the user should be able to do with web application.

Figure 14b show A simplified version of the layout template for how the single page web application should look like. The layout was created with functionality and component placement for the single page application in mind.

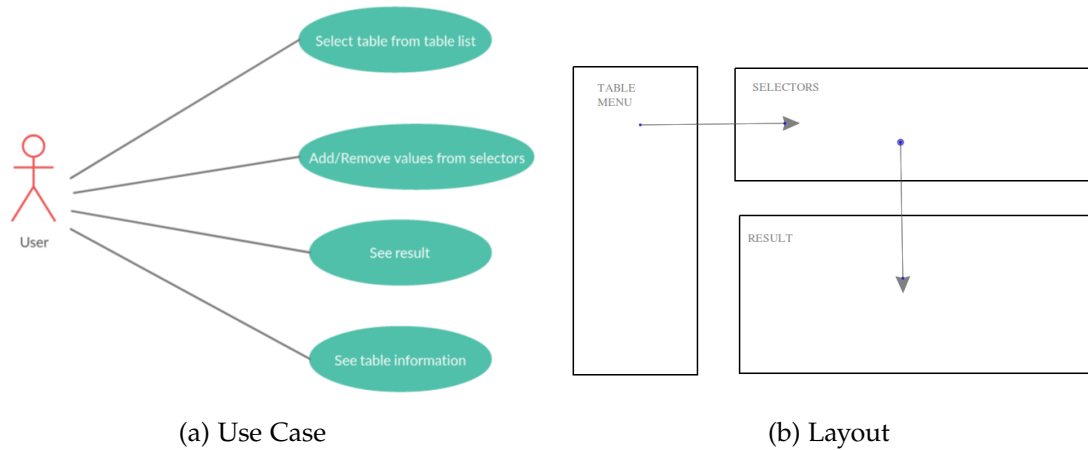


Figure 14: Communication

Figure 14a shows what the user should be able to do in the web application.

They should be able to pick the table, select and remove the values from the selectors and see the results and table information.

Figure 14b shows how the template view of the web application and how the functionality is displayed. A table generates selectors and the selectors generate a result.

6.1.2 Planning

The planning contained creating a simplified component diagram Figure 15 to give a better overview of all the components that needed to be created and their relationship with each other.

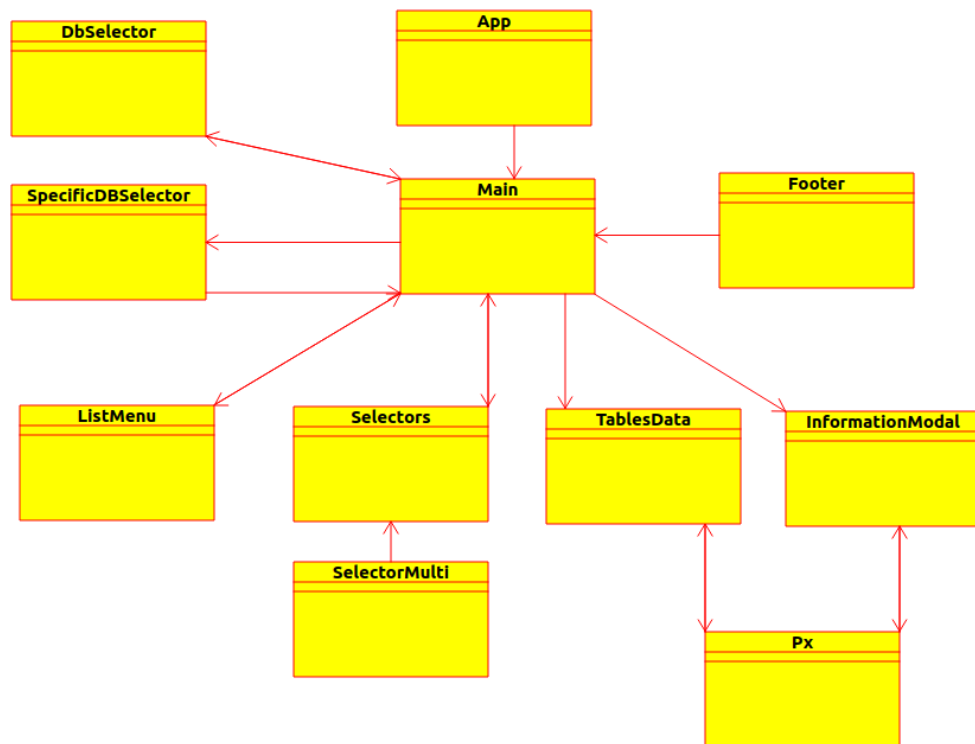


Figure 15: Component Diagram

An step by step explanation on what happens when application runs was also created to put words on how the components work together.

- App runs Main
- Main has a predefined URL selected from DBSelector.
- Main sends URL to ListMenu, where ListMenu generates a menu with tables from the URL data.
- When table is selected, ListMenu sends URL of the table to Main, where Main sends URL to Selectors.
- Selectors generates selectorsMulti's from the data in API URL and creates a JSON object from the values selected in each SelectorsMulti. A POST Request is then made on the URL with the object created. Data from POST Request is then sent back to Main.
- Main sends POST data to TablesData¹ and InformationModal. TablesData and InformationModal use Px to interpret the data and then display it.

The SpecificDBSelector is used when a statbank has many databases like the Icelandic statbank.

By having Main as the main component where all other components send their state to, I have control over when each component updates.

¹ TablesData is the component that shows the result

6.1.3 Modeling

To get a better understanding of the functionality of the prototype I created two sequence diagrams, also know as event diagram, and an activity diagram.

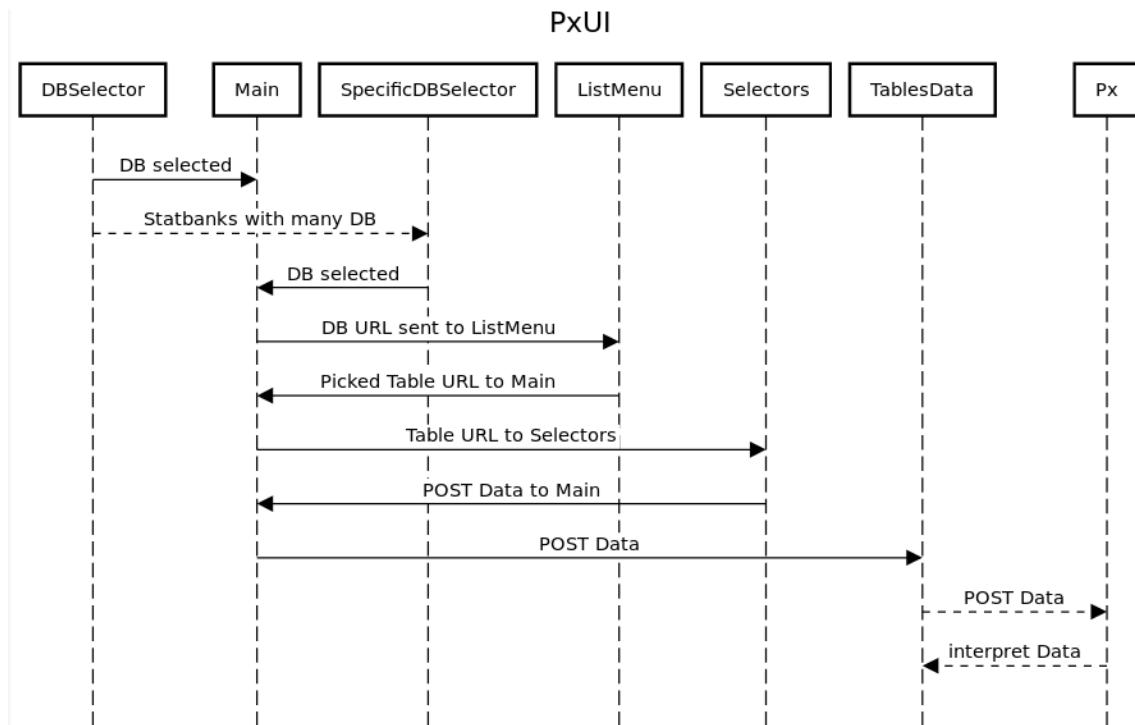


Figure 16: Sequence Diagram of Px UI

Figure 16 shows the same as explained in Section 6.1.2. The sequence diagram shows how the components communicate with each other and in which order they occur.

DBSelector tells Main which database is selected, then Main sends the URL to ListMenu. When a table is selected by the user, the table URL is then sent back to Main and so on.

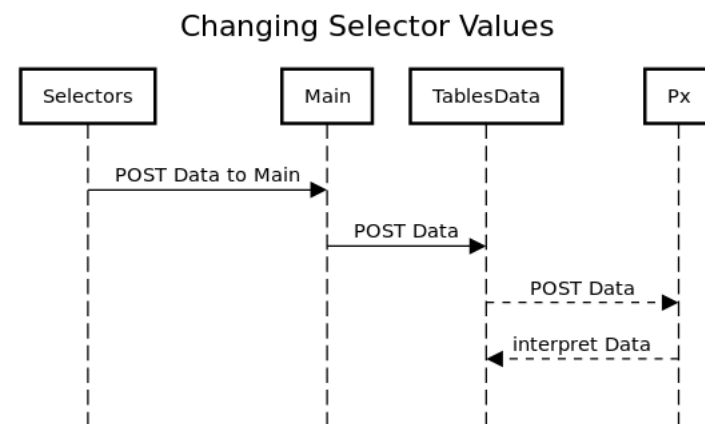


Figure 17: Sequence Diagram of Selectors

Figure 17 show what happens when the user changes values in one of the selectors.

After a value has changed, the Selectors component sends the POST data to Main. Main then sends the data to TablesData, that uses the Px interpreter to interpret the Px data and then displays it as the result on the web application.

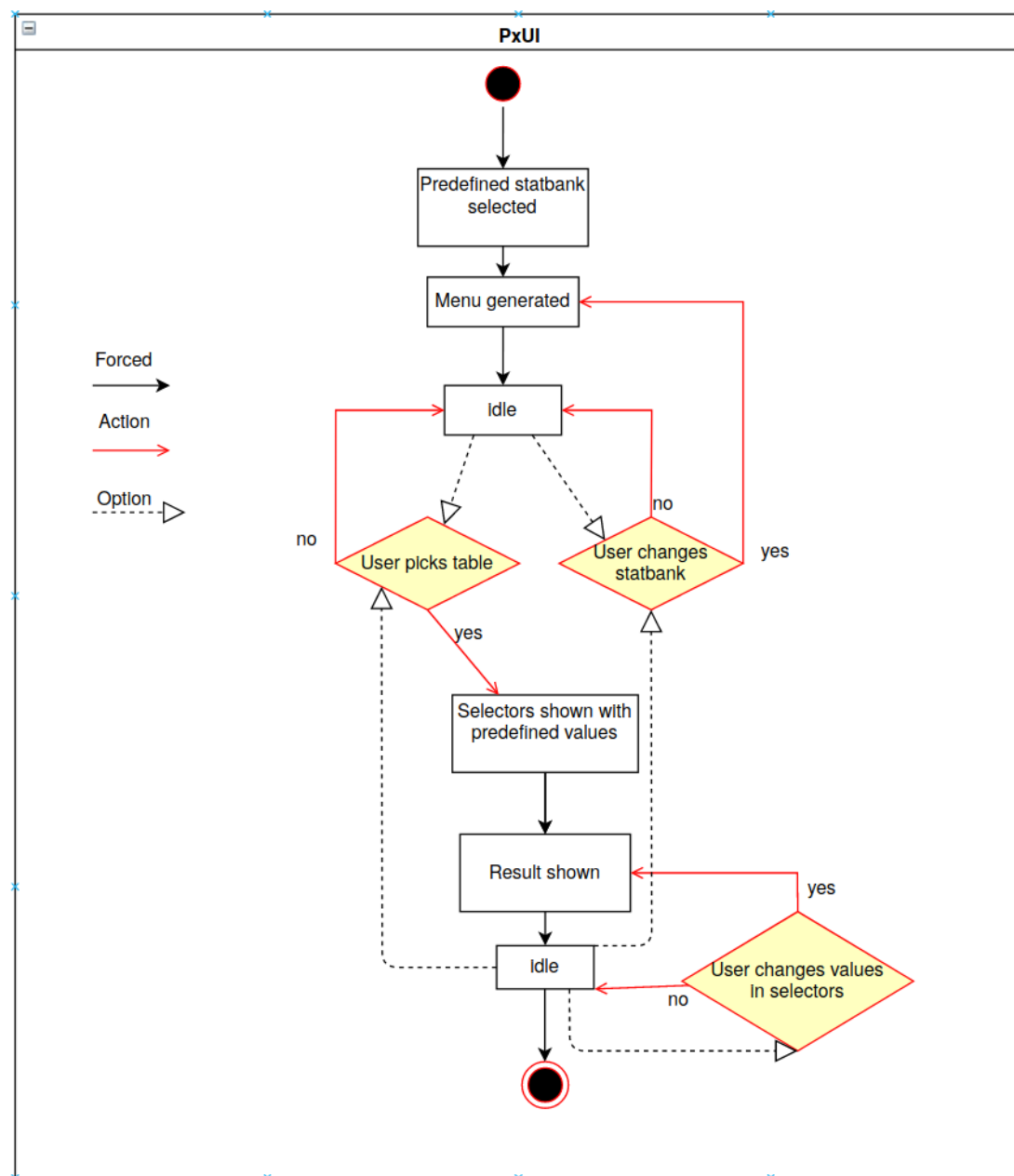


Figure 18: Activity Diagram of PxUI

Figure 18 describe how the activities in the prototype are coordinated to provide a result.

It shows that from the start, a predefined statbank is selected, then a menu gets generated, and the web application goes idle. With the menu generated, the user has two options. Either pick a table or change the statbank.

If the user decides to change statbank, a new menu will be generated, depending on the statbank picked.

If, on the other hand, the user picks a table, the selectors will be displayed and the result from the predefined values of the selector will be displayed.

Now the user has a third option, where it is possible to change the values in the selectors as he pleases and then immediately see the result of the values picked.

6.1.4 Construction

The Construction of the web application was done by developing one main component at a time. The Selectors where developed first. Then the Menu and last Result. The reasoning behind this was getting the Selectors working was be the hardest task and they where the ones that the success of the prototype most depended on.

I was in the fortunate position that Statistics Faroe Islands had already done the a JS Px interpreter for their own homepage, so my job was only to implement it when I had got the Selectors and Menu to work.

Since this is a prototype and time was of it's essence, a testing environment was not created.

6.1.5 Deployment

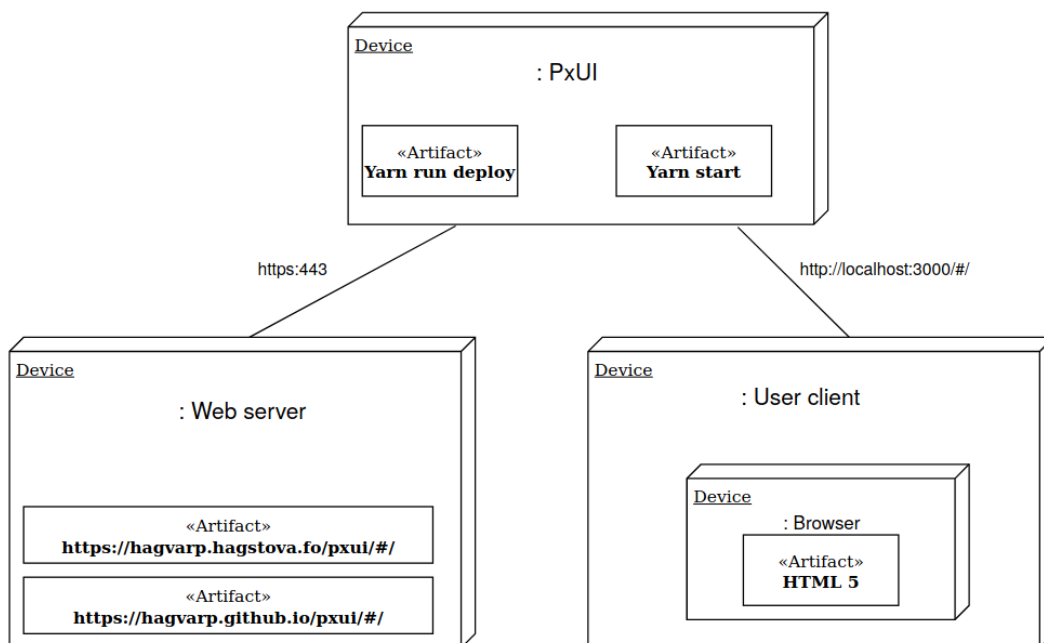


Figure 19: Deployment Diagram

To get feedback and evaluation of the prototype from users and others interested in the project, the prototype is hosted via github pages and made open source.

Figure 19 shows that the prototype can either be seen and used on a web server or run on localhost. Information about setting up the project locally can be found on [README²](#).

6.2 PROCESS FLOW

A important aspect of the software process is the process flow.

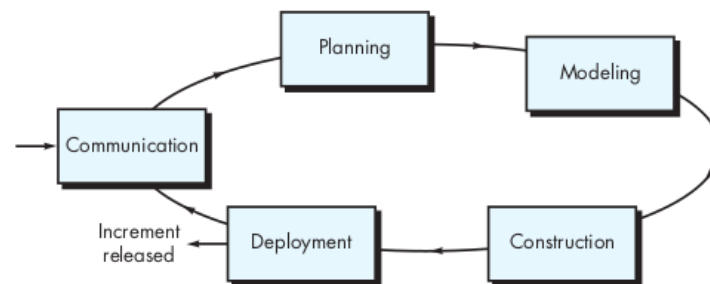


Figure 20: Evolutionary Process Flow

The process flow that was used in this project is the evolutionary process flow [Figure 20](#).

It describes how the five activities defines their relationship with the process and with one another.

This process flow worked best for this prototype, since one component was worked on and deployed at a time.

6.3 PROCESS MODEL

It is encouraged to adapt and follow a process model in software development since it brings order to the chaos that can occur in software development.

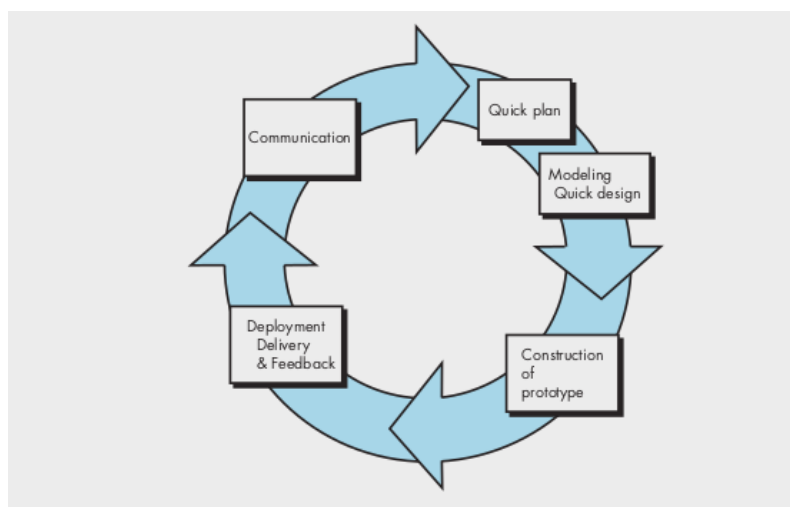


Figure 21: The Prototyping Paradigm

² URL: <https://github.com/hagvarp/pxui/blob/master/README.md>

The prototyping paradigm suited this project the best, since objectives were only generally defined and not detailed requirements for functions and features.

The prototyping paradigm also lets the developer build something immediately and the users get the feel for the actual system and in the same time.

The problems by using the prototyping paradigm are that often software quality and long-term maintainability are overlooked, but since time was a huge factor and the aim to have the prototype presented in the annual Px meeting 2019 - Armenia, this process model was best suited for this project.

6.4 AGILE

Agile is a time boxed, iterative approach to software delivery that builds software incrementally from the start of the project, instead of trying to deliver it all at once near the end.[6]

Agile is best suited for teams, but this doesn't mean that individuals can't adopt the basic components like creating user stories, having a backlog, refactor code and not least having code sprints. Anna Nystrom even wrote a Masters thesis about it and created a process model called *Agile Solo* [13].

I always tried to use the concepts behind agile and followed the 12 Principles Behind the Agile Manifesto [Appendix B](#).

The text above describes my practice. As mentioned in [Section 6.1.4](#), I divided the project into parts. From those parts I created sprints that contained smaller assignments.

The benefits of this is that the project is more organized and it helps with keeping track of the time and course of the project.

I used [Trello](#)³ for the organization of the project.

³ URL: <https://trello.com/>

6.4.1 Agile Process Model

I didn't follow any specific agile process model, since I worked alone. Instead I use my own customized model that was inspired by SCRUM.

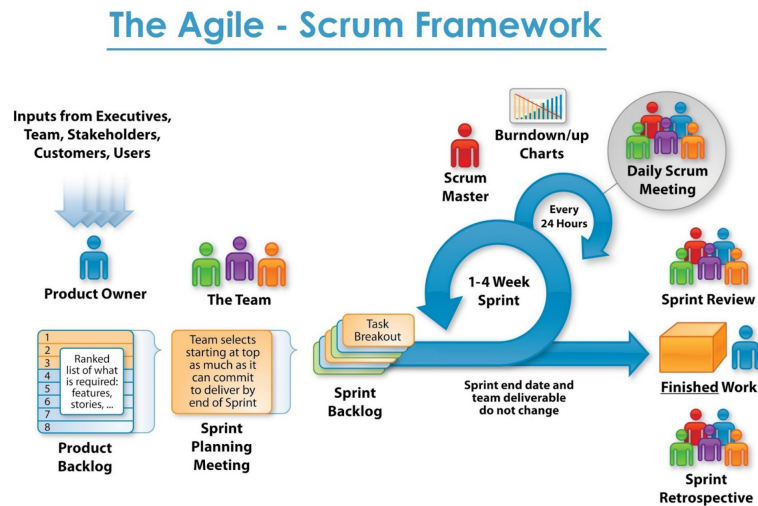


Figure 22: SCRUM

My sprints didn't a fixed number of days, like 1 week. They where more based on the component I was working on from the backlog.

My first coding sprint was the Selectors component. I had a sprint planning meeting with my supervisor Kári Holm Johannesen about the component and about how the PxWeb API worked.

I calculated roughly that the sprint would take two weeks to deliver the Selectors component, since my lack of experience with ReactJS and the PxWeb API.

I then made a task breakdown and started working on the component. My sprints didn't contain a SCRUM master or daily meetings, since I worked on the project alone. It was more me asking for help when I needed it and my supervisors checking up on me a few times a week on how it was going and if I needed help.

When the component was done, I always presented the result to my supervisors for feedback. This was done with a live presentation. If the component was approved the sprint ended and I made a personal review on how the sprint went and what I learned.

Part II

ARCHITECTURE

7

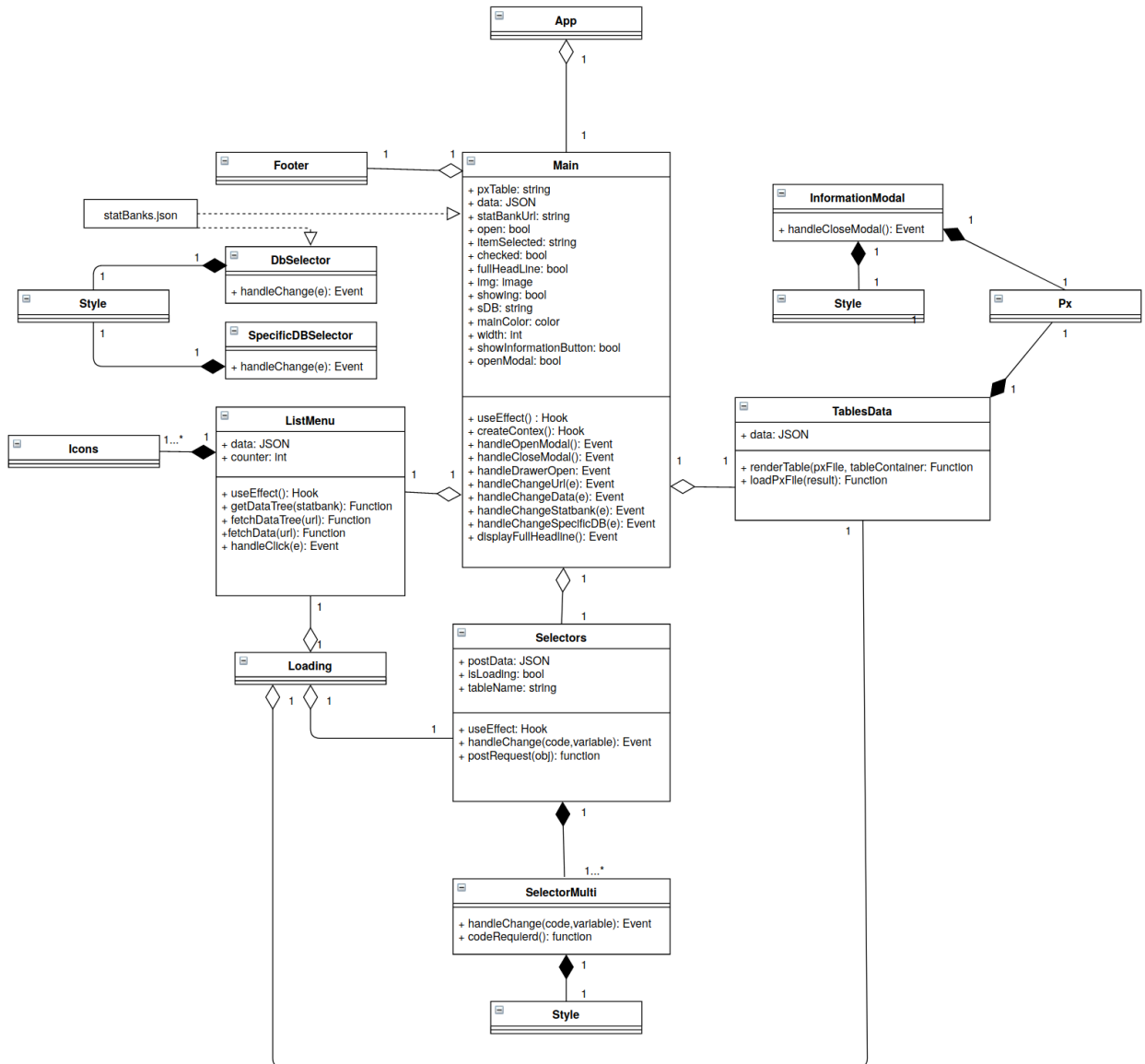


Figure 23: Class Diagram

Figure 23 shows the relationship type between the components, the component attributes and operations. It shows for example, the aggregation between ListMenu and Main. That ListMenu is a part of Main.

If we take a look at the aggregation between Selectors and SelectorsMulti, we see that one to many instances of SelectorMulti can be associated with Selectors.

The web application is built on the design principle separation of concerns, where each component addresses a separate concern. This design principle makes it easier to maintain, make changes or, add/remove components to the application.

If we take a closer look at Figure 23 with human anatomy in mind, we can see that Main component is the brain of the web application, while the other components are the body parts.

The reasoning behind this is to have better control over each component.

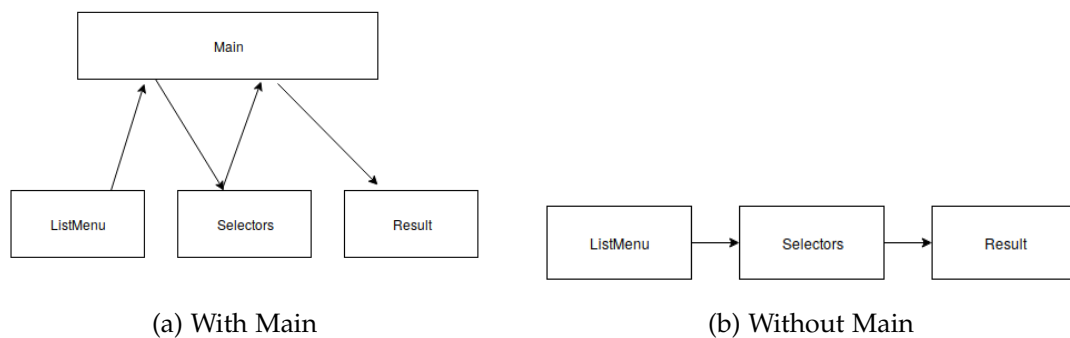


Figure 24: With/Without Main component

Figure 24 shows how it works with and Without Main. If the application became much bigger, the chain in Figure 24b would become bigger and with that harder to make changes to components since they depend on each other.

7.1 MAIN

Main is the core and layout of the web application. In Figure 23 we also see that its the biggest component.

The layout in Main is built by using **Material UI**¹, a React UI framework that is excellent for styling the web application.

By using Material UI and it's documentation, I was able to create a layout for the application promptly and focus on the functional aspect of the web application.

¹ URL: <https://material-ui.com/>

7.1.1 Passing New Data From One Component To Another

When the web application runs the first time the `statBankUrl` state is `statBanks[0].value` -> "https://statbank.hagstova.fo/api/v1/en/H2/". This state is then passed to `ListMenu`, which creates the menu.

Listing 6: ListMenu

```
1 <ListMenu onClickItem={handleChangeUrl} statBank={statBankUrl} />
```

To change the database to Statistics Greenland, the user needs to change the value in `DbSelector`.

Listing 7: DbSelector in Main

```
1 <DbSelector onChange={handleChangeStatBank}></DbSelector>
```

Listing 8: DBSelector

```
1 export default function DbSelector(props) {
2   const handleChange = e => {
3     props.onChange(e);
4   };
5
6   return (
7     <ColorContext.Consumer>
8       {color => {
9         return (
10          <Select
11            className="selectors"
12            styles={Style(color)}
13            onChange={handleChange}
14            closeMenuOnSelect={true}
15            defaultValue={[statBanks[0]]}
16            options={statBanks}
17          />
18        );
19      }}
20     </ColorContext.Consumer>
21   );
22 }
```

When the value is changed, `onChange` is triggered and the event handler `handleChange` is called. `handleChange` then makes a callback to Main with `props.onChange(e)`.

Now In Main, `handleChangeStatBank(e)` runs and sets the new state -> `setStatBankUrl(e)`. This renders `ListMenu` with the new url, since `statBankUrl` has changed.

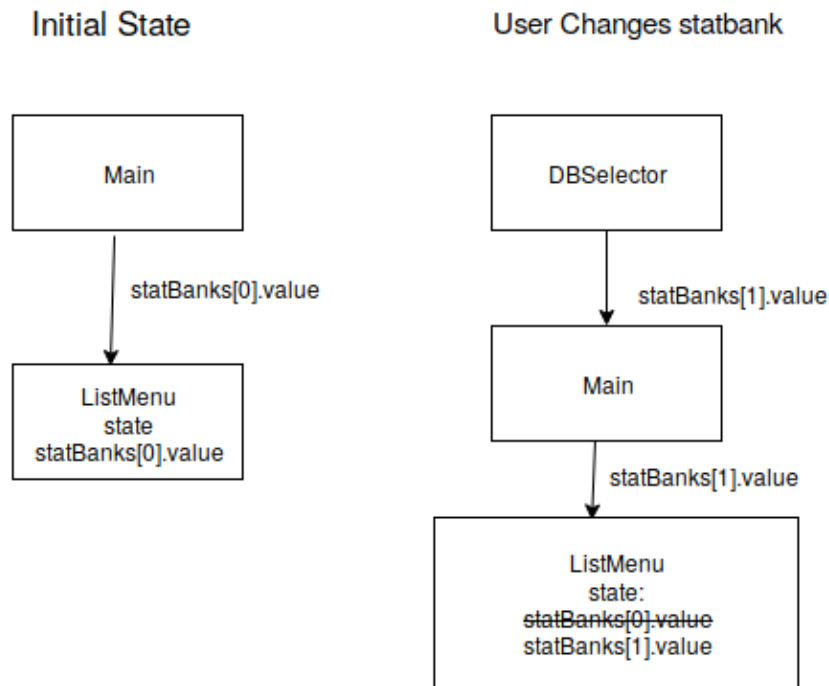


Figure 25: Changing state

Figure 25 shows the initial state of the statbank selected looks like and how it changes when the user changes statbank.

7.1.2 Context

Context provides a way to pass data through the component tree without having to pass props down manually at every level.²

Since I wanted to change the color theme depending on what database was selected, I used context to pass the mainColor to all components.

7.1.3 Routes

Routes were implemented very late in the project, simply because I didn't think it was necessary, but since Hagstovan wants to use this prototype as their mobile version of the Faroese statbank it was required. The reason is that they have their own statbank search engine that finds tables in their statbank. So to link to a specific table is required: https://hagvarp.github.io/pxui/#/IB/IB01/fo_vital_md.px

My approach to making this possible with the little time I had. I made two routes. One with path = "/" : customPath" and the other excatpath = "/" but the problem now is that I have copied the same code twice into both routes, with the only difference in < Selectors > where pxTable is changed:

Listing 9: customPath

² Context

```

1 pxTable={
2   pxTable ||
3   "https://statbank.hagstova.fo/api/v1/en/H2/" +
4   props.location.pathname.substr(1)
5 }

```

Listing 10: excat path

```

1 pxTable={pxTable}

```

7.2 LISTMENU

The major problem with the API is that it doesn't contain an end-point where it can fetch all the menu in one call. See ??

The way PxWeb is using the menu is by making a fetch when the user clicks on a menu item. This method prevents server spamming, but prevents the possibility to search for a specific table, since only the main menu items have been fetched.

To make it possible to for the user to search for a menu item, all the menu items need to be fetched at once recursively.

7.2.1 Network Error 429

Approximate 75 fetches need to be done to obtain the Faroese statbank menu. So if I open 10 browser tabs <https://hagvarp.hagstova.fo/pxui/#/> I make 750 request on the server in a few seconds.

This will put the server under heavy load and in worst case crash it.

To prevent this, most servers have put a limit on requests inside a time frame, some have a big limits while others have smaller limits.

The Faroese³ and Greenlandic statbanks have high limits, so fetching those two menus is not a problem.

The Swedish, Icelandic and other statbanks have low limits on their servers, so trying to fetch their menu with Promise.all() gives Network Error 429 arrives, "Too many request, in to short time".

A promise is an object that serves as a placeholder for a value. That value is usually the result of an async operation such as an HTTP request or reading a file from disk. When an async function is called it can immediately return a promise object. Using that object, you can register callbacks that will run when the operation succeeds or an error occurs[12 15, p. 11].

3 Faroese statbank API:
Limit request: 9999999
Limit timespan: 99999

A promise has three states[15, p. 15]:

- Pending
- Fulfilled
- Rejected

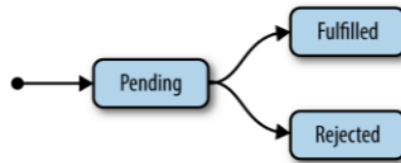


Figure 26: relationship between the three states

The `Promise.all()` function maps a series of promises to their fulfillment values. It accepts an iterable object such as an Array.

The function returns a new promise fulfilled by an array containing the values in the iterable. Corresponding fulfillment values in the resulting array replace any promises contained in the iterable. The new promise that the function returns is only fulfilled after all the promises in the iterable are fulfilled, or it is rejected as soon as any of the promises in the iterable are rejected. If the new promise is rejected it contains the rejection reason from the promise in the iterable that triggered the rejection.[15, p. 23]

It is typically used after having started multiple asynchronous tasks to run concurrently.

I tried using `timeout` with `Promise.all()` to get passed the Error 429, but it didn't work as I wanted. `fetch()` on the other hand worked, but it took a couple of minutes, which was unacceptable.

The core problem in this is how the API delivers the menu. A problem that I think they have overlooked with a easy fix by generating the menu on the server and delivering it with one end-point.

7.2.2 Caching

We discussed caching the menu to prevent server spamming, but we came to the conclusion that it was not necessary, since this is a prototype and the user should be able to get the new data as soon as it was updated.

7.2.3 Menu Tree JSON Object

I had lots of problems with creating the tree JSON object. I couldn't figure out how to nest the nodes in a recursive manner.

Listing 11: Menu Tree Example

```
1 | {
```

```

2      "id": "U0",
3      "nodes": [
4        {
5          "id": "U001",
6          "nodes": [
7            {
8              "id": "U00101",
9              "nodes": [
10               ...
11             ]
12           },
13         ]
14       }
15     ]
16   }

```

After being stuck for a while and feeling I had used too much time on this problem, I asked stackoverflow [Question](#) for help.

In a couple of minutes the user [Maxime Girou](#) solved my problem and showed me how it was done.

From that point I just made the changes needed to make it work in my application.

One thing to note is that I used `promise.all()`⁴ instead of `fetch()` in the recursive function. `Promise.all()` accepts individual promises and gives back a single promise that is resolved when all the individual promises are resolved or rejected.

7.2.4 *react-simple-tree-menu*

`react-simple-tree-menu`⁵ is a great library for building tree menus.

Listing 12: TreeMenu

```

1 <TreeMenu className="tree-item" data={data} onClickItem={handleClick} />

```

At first [Listing 12](#) was used to show the menu, but since I wanted control over the menu, I followed this guide: <https://github.com/iannbing/react-simple-tree-menu/blob/master/stories/index.stories.js>

That made it possible to change the styles, icons and most importantly making the menu items clickable, instead of just the icon itself.

7.3 SELECTORS

The Selectors are generated from the selected Px table data [Figure 27](#). [Figure 27a](#) shows the table `fo_vital_md.px`, that contains a title and four variables, or as I call them selectors.

⁴ `Promise.all()`

⁵ `react-simple-tree-menu`

▼ title:	"IB01010 Fólkatalsbroytingar skiftar á fødd, deyð, flutt og bygd, mánaðarliga (1985-2019)"
▼ variables:	
▶ 0:	{...}
▶ 1:	{...}
▶ 2:	{...}
▶ 3:	{...}
▶ 4:	{...}

(a) Px Table Data

▼ title:	"IB01010 Fólkatalsbroytingar skiftar á fødd, deyð, flutt og bygd, mánaðarliga (1985-2019)"
▼ variables:	
▼ 0:	
code:	"village/city"
text:	"bygd/býur"
▼ values:	
0:	"999999"
1:	"410101"
2:	"410102"
3:	"410201"
4:	"410301"

(b) Variables Data

Figure 27: fo_vital_md.px

for loop is used to create selectorMulti components. Each selectorMulti is then pushed into a array (selectorArray) and displayed when the array (selectorArray) is bigger than 0.

To give the user a instant result from the table selected a query object is created [Section A.1](#) with each selector first value and a POST request with the query object is made on the table.

Each selectorMulti component has an event handler with a callback function. When a selectorMulti changes it's values, the query object gets updated with the new value and a new POST request is made with the new query object.

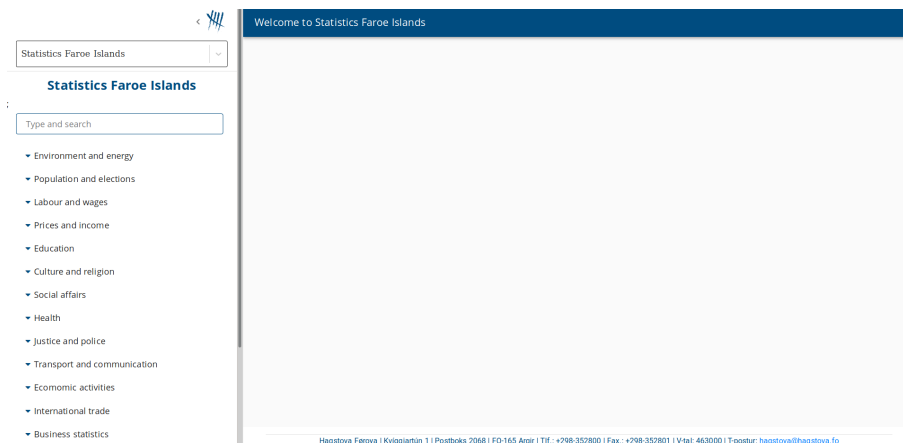
7.4 TABLESDATA

TablesData is the element where the result gets shown. Statistics Faroe Islands had already created a interpreter for Px data [Section A.2](#) and a JS example file. I rewrote and made the changes needed to the example.js file to suite the project and implemented it.

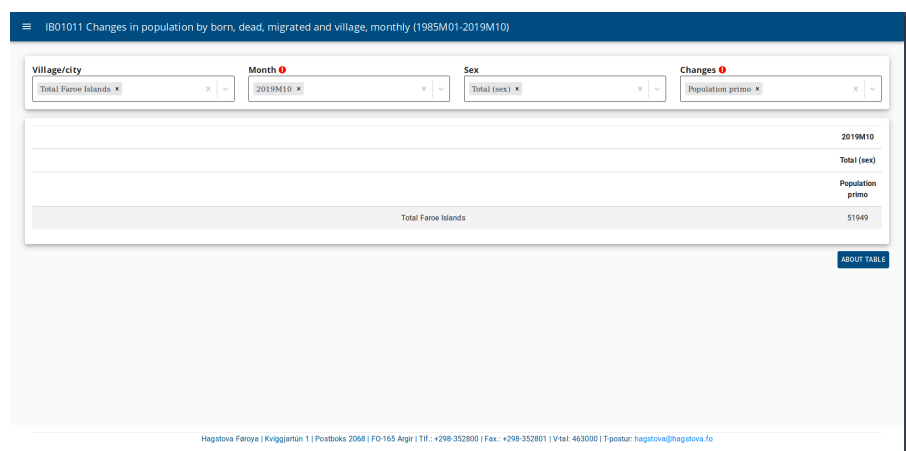
InformationModal works in similar fashion, but here the example.js is not used as a template.

8

User Interface Design



(a) Home Screen



(b) Table picked

Figure 28: PxUI

When it comes to web applications, the UI design is important for the user experience. A great UI design will make people come visit your site while a bad design has the opposite effect.

This is the reason we see popular web pages, like Facebook, change their design time to time. Coincidentally **Statistics Faroe Islands**¹ launched a new homepage they have been working on, while I worked on this project to keep up with today's standards.

¹ URL: www.hagstova.fo

In [Chapter 1](#) I defined the problems with [Statbank](#)² by using ISO 9241 Usability. This standard is the foundation for the UI design:

- Effectiveness is achieved by making the web application responsive in a way that guides the user to his goal.
- Efficiency is achieved by making the prototype a single page application. This reduces the time and number of clicks to get the right data.
- Satisfaction is reached by simplifying the prototype. The prototype only contains useful functionality

To make the web application effective, everything is made responsive. If the user searches for a menu item, only the menu items with those keywords show. Effectiveness is also improved by updating the color theme as soon as the user changes statbank. This is a simple way to let the user know what statbank is selected.

To make the web application as efficient as possible, no steps are used as in [Statbank](#)³. Instead its a single page application, that shows a result as soon as the user picks a table. Also here the responsiveness of the application helps with the efficiency.

The web application only contains the most important components, basically a simplified version of the standard PxWeb. This has given many users great satisfaction, since most users only want to check simple statistics, like how many people live in Tórshavn.

² URL: <https://statbank.hagstova.fo/pxweb/fo/H2/>

³ <https://statbank.hagstova.fo/pxweb/fo/H2/>

8.1 MOBILE PHONES/TABLES

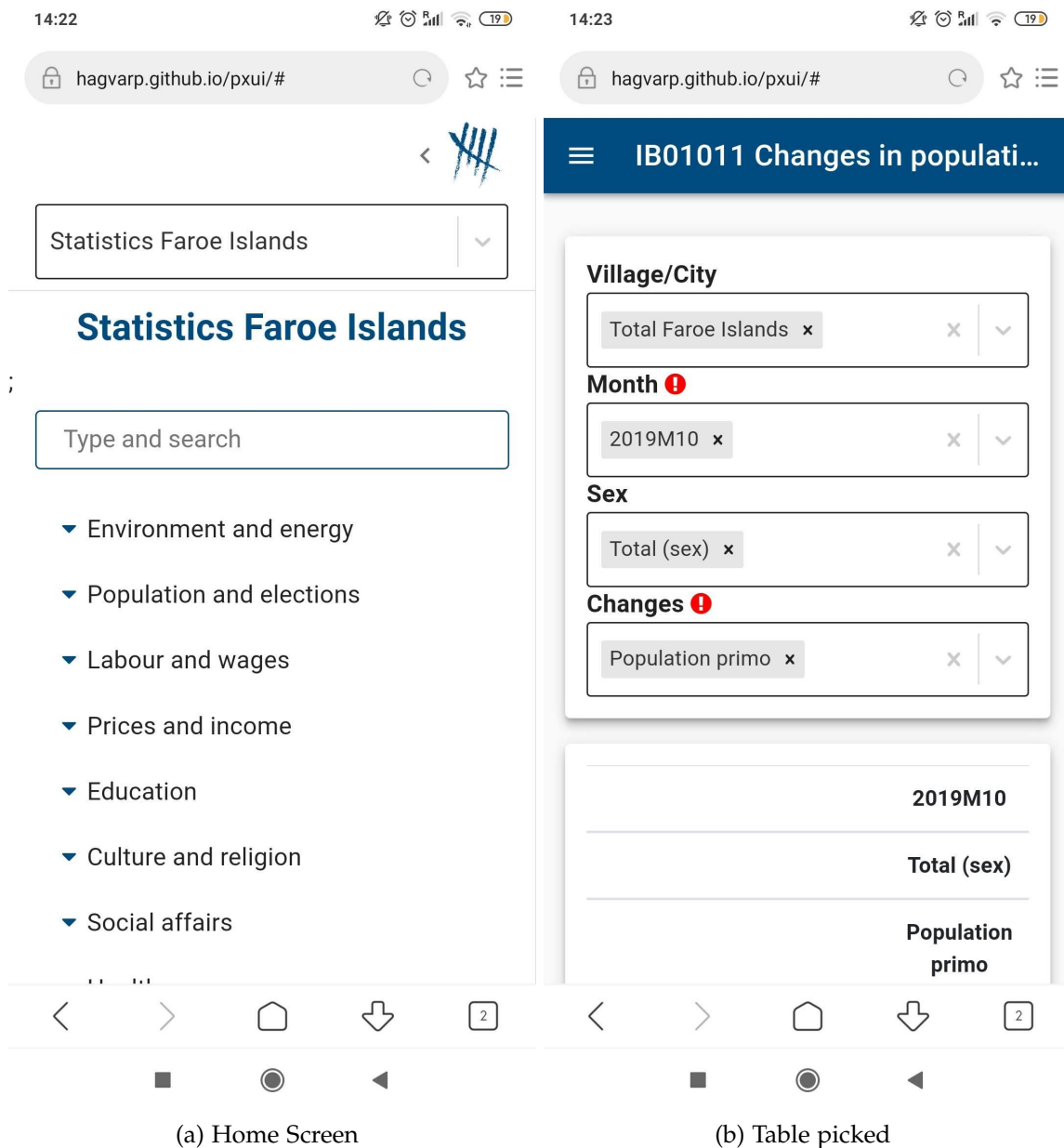


Figure 29: PxUI Mobile

The web application UI design is created in such a way that it works and looks really good on mobile phones and tablets.

To make the drawer [Figure 29a](#) look good on any device, the drawer width is determined by the width of the browser window. If the width is less than 1000px, the drawer will be the width of the browser window, or it will be set to 500px.

An event listener is used for monitoring the browser window width, so if the user flips their phone, the drawer gets adjusted to that width.

The toolbar [Figure 29b](#) also contains an event listener for the headline. At initial state, the headline is not displayed fully. The user can see the full headline by pressing on it. This is done by a boolean and noWrap.

9

Prototype Functionality

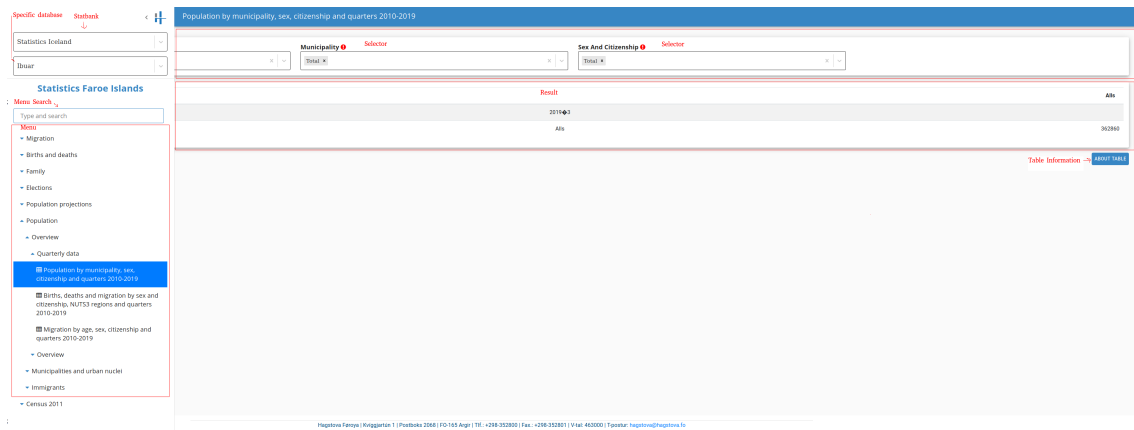


Figure 30: Px UI Functionality

Figure 30 gives a clearer picture on what all the components do. Here is a short description on what each component does:

- First Selector Top Corner, selects statbank.
- Second Selector Top Corner, selects specific database in the statbank.
- Searchbar, searches for specific menu item.
- Menu, for picking table.
- Selectors, for picking the specific data the user wants to see.
- Result, shows the result from the specific table and Selectors selected.
- Table Data Button, shows the information about the specific table selected.

Part III

ANALYSIS

10

Presentations

The prototype was presented on two occasions. Once by me at an event on 24.10.19 called Mennarar - Mennara Meetup¹ and then on 13.11.19 by Torbjørn Andreas Lisberg at the annual International Px Meeting 2019 - Armenia.

10.1 MENNARA MEETUP

My former work colleague John Høj Andreassen from **TechCare sp/f²** asked if I was interested in presenting the prototype at the event.

I accepted without hesitation since this was a great opportunity to get feedback about the functionality of the prototype from other developers and getting experience in presenting the prototype.

The presentation went excellent and the overall feedback was very positive.

After the presentation we had a very beneficial discussion about the menu, since doing so many fetches is never good for the server, but most agreed on that it was the lack of an end-point to get all the menu, that was the problem.

10.2 INTERNATIONAL PX MEETING 2019 - ARMENIA

A personal goal was to have a functional prototype ready for the International Px Meeting 2019 in Armenia, but since I wasn't sure if I would manage to deliver the prototype at that time, I didn't include it as an objective for the project.

The coding and styling went exceptionally good, so on 13.11.19 I had a finished functional prototype that could be presented on the International Px Meeting 2019.

The prototype was a real eye-opener for the other statistical offices because it showed a different way of how the statbank could be utilized. Now with only a few clicks, an instant result appears, compared to the old web application where you have to go back and forth through the steps to pick a new table or change the data you want to be displayed.

It's worth to mention that the prototype was presented in mobile view, where it worked perfectly.

PxWeb works poorly, or not at all on most mobile phones, so that made the prototype even more special.

¹ Meetup for people interested in programming in Faroe Islands, hosted by John Høj Andreassen
<https://www.facebook.com/Mennarar-101542821231917/>

² URL: www.techcare.fo

11

Benefits of the Prototype

The user experience has significantly improved with the elimination of the three steps and with the reactive UI and the modern look.

11.1 REMOVING THE THREE STEPS

By removing the three steps and creating the prototype as a single page application, the user experience has improved immensely.

Instead of guiding the user to their destination as the three steps do, the prototype gives the user full control with a simplified UI.

The prototype makes it possible for the user to pick a table or change values with ease. PxWeb, on the other hand, demands that the user goes back to the first step and then start over.

11.2 REACTIVE

The reactivity of the prototype lets the user know the result from their actions right away. This gives the user a much better feel for the application and makes the experience more pleasant.

11.3 MODERN LOOK

The UI has also got a modern look, which improves the user experience, especially for mobile users.

12

Analysis of the Success Criteria

12.1 REMOVE POSTBACK AFTER USER INTERACTION

The postback has been removed by using ReactJS. The web application does not reload anymore with user interaction.

Example of this can be seen when the user pick a menu item, a table or changes values in the selectors.

12.2 MAKE THE UI RESPONSIVE

The UI is now responsive. When the user changes statbank, the new menu gets instant generated, same goes for picking a table, where new selectors get instantly generated.

The result also gets instant updated as soon as the user changes values in the selectors.

12.3 USING MODERN COMPONENTS

Only new modern components have been used in this prototype.

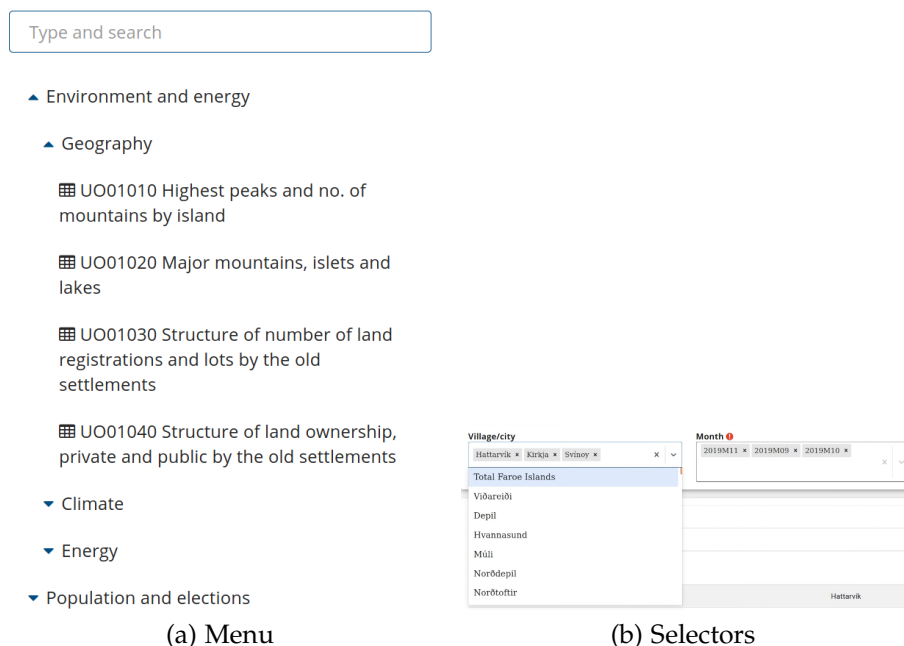


Figure 31: Menu and Selectors

	2019M09	2019M10	2019M11
	Total (sex)	Total (sex)	Total (sex)
	Population primo	Population primo	Population primo
Hattarvík	11	11	11
Kirkja	27	26	27
Svinoy	32	32	32

Figure 32: Result

The menu [Figure 31a](#) has now gotten a modern look with a search bar added, for making it easier for users to find the right data they are looking for.

The selectors are modern style selectors. They work as a drop down menu where you can pick your values.

It's also possible to search in the selector by just pressing on it and typing. It works in similar fashion as the search function in the menu. Its also possible to remove all values by pressing the x in the selector bar.

The selector also grows dynamically as more items get selected.

The result table [Figure 32](#) has also gotten a new modern look.

12.4 REDUCE THE NUMBER OF CLICKS

A test was made to compare the number of clicks of the prototype and PxWeb, to see if the prototype was more efficient than PxWeb.

The aim is to do the same in both applications. Go to a specific table, change specific values, and then select a new specific table.

- See the results from https://statbank.hagstova.fo/pxweb/fo/H2/H2__IB__IB01/fo_vit_md_t.px/.
- adding two villages (Hattarvík, Kirkja) and see the new result.
- Select the new table https://statbank.hagstova.fo/pxweb/fo/H2/H2__IB__IB01/fo_vital_md.px/table/tableViewLayout2/ and see the results from it.

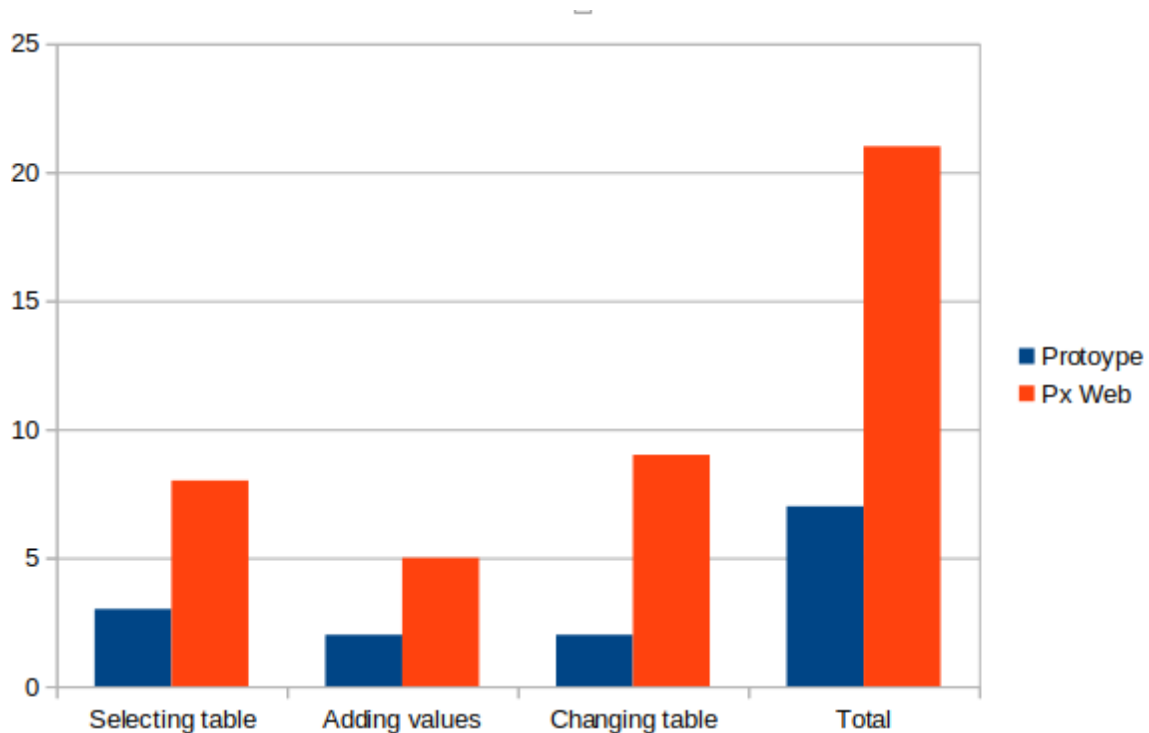


Figure 33: Click Test

Figure 33 shows that the user only needed to perform seven clicks with the prototype while 21 with PxWeb. This is a reduction of 66.6%. This means that the prototype is more now efficient in all tasks.

This is a significant improvement in efficiency, considering how small the test was.

12.5 MAKE THE UI USER-FRIENDLY

There has not been a formed a test group to figure out if the UI is more user-friendly or not, but the feedback from the presentations [Chapter 10](#), Statistic Faroe Islands and everyone has been overwhelmingly positive, so that suggests that the UI is more user-friendly than PxWeb.

12.6 FINDING THE RIGHT DATA FASTER

Users can find the right data faster now, with the search functionality in both the menu and the selectors. '

The predefined values in the selectors also give the user a faster result, which contributes to finding the right data for the user.

The cleaner and easier to use interface also helps the user finding the right data faster, since it removes the doubt about how the get the desired result.

Part IV

SUMMARY

13

Findings

- This project is unique, concerning Px, since I couldn't find any similar projects.
- React is a excellent library for front-end functional prototyping.
- The software engineering process is a excellent tool for building high quality prototypes.
- Project management techniques are excellent for organizing and planning small projects like this.
- Agile is not only for teams. Individuals can also benefit greatly from adopting it.
- By developing a prototype with ReactJS, I was able to improve the effectiveness and efficiency of the interface.
- Using ReactJS also made it easy to make the application mobile friendly.
- Prototyping is a good way to test new technology.

14

Conclusions

This thesis aimed to develop a prototype that would improve the effectiveness and efficiency of the interface and exploit and utilize the API in the PxWeb statbank.

Based on the analysis of the success criteria [Chapter 12](#), that show that the prototype is much more effective, efficient and user-friendly than PxWeb, it can be concluded that the developed prototype, with the help of the software engineering process and project management techniques, is a significant success.

User feedback gathered from the presentations and people working with PxWeb in Statistics Faroe Islands showed that the UI is much more intuitive and easier to use, especially on mobile devices.

Analysis of the success criteria [Chapter 12](#) also states that the number of clicks needed to get a result has been reduced significantly, compared to PxWeb.

A simple task that takes 21 clicks with PxWeb [Figure 33](#), now takes seven clicks with the prototype.

The success of the prototype indicates that the people behind PxWeb should use this prototype as a reference when going forward with PxWeb. Meaning that instead of working and updating the old PxWeb, a new PxWeb with modern technology, like ReactJS, could be created.

15

Future Work

The next step for this prototype is to convert it to a finished mobile product for Statistics Faroe Islands.

This process will include code cleaning, adding a testing environment and improving the UI design.

There are also numerous functionalities that PxWeb contains, that are not included in this prototype, such as visualizing the results with graphs, saving the result, saving the query and swapping rows and columns of the result.

These features could be studied and implemented in future work.

The biggest hindrance for making this a finished product is the creation of the menu tree object [Section 7.2.1](#).

The best way to get passed this hindrance is by establishing communication with Statistics Finland¹, to figure out if is possible to serve the whole menu with one call as a JSON object.

¹ Administrator for PxWeb API

Part V

APPENDIX



Appendix Px

A.1 A QUERY EXAMPLE

Listing 13: A Query Example

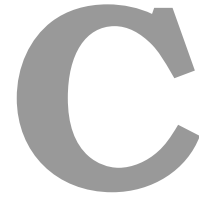
```
1  {
2  "query": [
3    {
4      "code": "village/city",
5      "selection": {
6        "filter": "item",
7        "values": [
8          "999999"
9        ]
10     }
11  },
12  {
13    "code": "year",
14    "selection": {
15      "filter": "item",
16      "values": [
17        "2019"
18      ]
19    }
20  },
21  {
22    "code": "month",
23    "selection": {
24      "filter": "item",
25      "values": [
26        "M01"
27      ]
28    }
29  },
30  {
31    "code": "sex",
32    "selection": {
33      "filter": "item",
34      "values": [
35        "M"
36      ]
37    }
38  },
39  {
40    "code": "changes",
41    "selection": {
42      "filter": "item",
43      "values": [
44        "Popu_primo"
45      ]
46    }
47  }
48 ],
49 "response": {
50   "format": "px"
51 }
52 }
```

CHARSET="ANSI";
AXIS-VERSION="2010";
CODEPAGE="iso-8859-15";
LANGUAGE="en";
LANGUAGES="en","fo";
CREATION-DATE="20000101 09:00";
DECIMALS=0;
SHOWDECIMALS=0;
MATRIX="IB01010_FO_VITAL_MD";
COPYRIGHT=YES;
SUBJECT-CODE="IB";
SUBJECT-AREA="Population and elections";
DESCRIPTION="IB01010 Changes in population by born, dead, migrated and village, monthly (1985-2019)";
TITLE="Changes in population by village/city, year, month, sex and changes";
CONTENTS="Changes in population";
STUB="village/city";
HEADING="year","month","sex","changes";
VALUES("village/city")="Total Faroe Islands";
VALUES("year")="2019";
VALUES("month")="Jan";
VALUES("sex")="Males";
VALUES("changes")="Population primo";
TIMEVAL("year")=TLIST(A1),"2019";
CODES("village/city")="999999";
CODES("year")="2019";
CODES("month")="M01";
CODES("sex")="M";
CODES("changes")="Popu_primo";
DOMAIN("village/city")="village-en";
ELIMINATION("village/city")="Total Faroe Islands";
LAST-UPDATED="20190911 09:00";
UNITS="persons";
CONTACT="Jógvan Bærentsen";
SOURCE="Statistics Faroe Islands: www.hagstova.fo/en";
NOTE="<u>Symbols:</u>## - Nil (exactly zero)# 0 Rounded to zero# . Not applicable# .. Not available or too uncertain# ... Not disclosed, confidentiality/privacy protected####";
NEXT-UPDATE="20191014 09:00";
TABLEID="IB01010";
UPDATE-FREQUENCY="M";
VARIABLE-TYPE("village/city")="M";
VARIABLE-TYPE("year")="M";
VARIABLE-TYPE("month")="M";
SUBJECT-AREA[fo]="Íbúgvar og val";
DESCRIPTION[fo]="IB01010 Fólkatalsbroytingar skiftar á fødd, deyð, flutt og bygd, mánaðarlaga (1985-2019)";
TITLE[fo]="Broytingar í fólkatalinum eftir bygd/býur, ár, mánaður, kyn og til- og frágongd";
CONTENTS[fo]="Broytingar í fólkatalinum";
STUB[fo]="bygd/býur";
HEADING[fo]="ár","mánaður","kyn","til- og frágongd";
VALUES[fo]("bygd/býur")="Alt landið";
VALUES[fo]("ár")="2019";
VALUES[fo]("mánaður")="Jan";
VALUES[fo]("kyn")="Mannfólk";
VALUES[fo]("til- og frágongd")="Fólkatalið fyrst í mánaðinum";
TIMEVAL[fo]("ár")=TLIST(A1),"2019";
CODES[fo]("bygd/býur")="999999";
CODES[fo]("mánaður")="M01";
CODES[fo]("kyn")="M";
CODES[fo]("til- og frágongd")="Popu_primo";
DOMAIN[fo]("bygd/býur")="village-fo";
ELIMINATION[fo]("bygd/býur")="Alt landið";
LAST-UPDATED[fo]="20190911 09:00";
UNITS[fo]="persónar";
CONTACT[fo]="Jógvan Bærentsen";
SOURCE[fo]="Hagstova Føroya: www.hagstova.fo";
NOTE[fo]="<u>Tekn:</u>## - Veruliga null (ikki rundað)# 0 Rundað til null# . Kann ikki koma fyrri# .. Ikki til taks ella ikki álitandi# ... Ikki upplýst, trúnaðarvart####";
VARIABLE-TYPE[fo]("bygd/býur")="M";
VARIABLE-TYPE[fo]("ár")="M";
VARIABLE-TYPE[fo]("mánaður")="M";
DESCRIPTIONDEFAULT=YES;
DATA=
26526
;



Appendix Agile Manifesto

- I Customer satisfaction by early and continuous delivery of valuable software.
- II Welcome changing requirements, even in late development.
- III Deliver working software frequently (weeks rather than months).
- IV Close, daily cooperation between business people and developers.
- V Projects are built around motivated individuals, who should be trusted.
- VI Face-to-face conversation is the best form of communication (co-location).
- VII Working software is the primary measure of progress.
- VIII Sustainable development, able to maintain a constant pace.
- IX Continuous attention to technical excellence and good design.
- X Simplicity—the art of maximizing the amount of work not done—is essential.
- XI Best architectures, requirements, and designs emerge from self-organizing teams.
- XII Regularly, the team reflects on how to become more effective, and adjusts accordingly.



Test Environment

At the start of the project I wanted create a testing environment with continuous integration, end to end testing and some unit testing.

I didn't create any tests given the pressure of finishing the prototype on time.

Tests can also take a long time to write for an developer like me, that has little experience with writing them.

I could implement continuous integration right away, but since the project was on the Faroese statistical offices github page and they didn't have a circleCI account, I decided against it.

I had no intention of using TDD, since I have no experience with TDD.

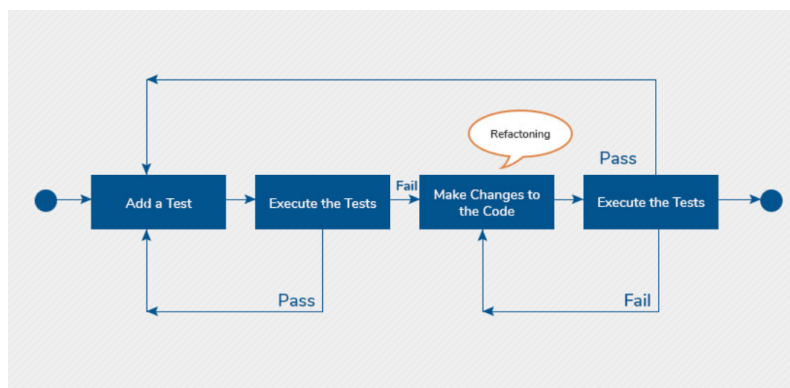


Figure 34: Test Driven Development

I made a plan at the beginning of the project of what the testing environment should contain.

C.1 CONTINUOUS INTEGRATION

I wanted to use **CircleCI**¹ for checking the builds before they got merged into GitHub, because I always wanted the master branch on GitHub to be free of bugs that break the application.

I also wanted to apply some rules so that the code was always formatted and had the same style, so that it was easier to read.

¹ URL: <https://circleci.com/>

C.2 END TO END TESTING

For end to end testing I wanted to use **Cypress**².

Cypress is a excellent framework for end to end testing. Its easy to use, easy to run test and contains a UI that shows the test running.

C.3 UNIT TESTING

For unit testing I wanted to use the framework **Jest**³, which focuses on simplicity.

² URL: <https://www.cypress.io/>

³ URL: <https://jestjs.io/>

Bibliography

- [1] Anthony Accomazzo, Nate Murray, Ari Lerner, Clay Allsopp, David Guttman, and Tyler McGinnis. *Fullstack React*. Fullstack.io, 2017.
- [2] B3LLM4R. *Obscure Data Formats: .px files*. 2014. URL: <https://exversiondata.wordpress.com/2014/06/17/obscure-data-formats-px-files/>.
- [3] Bill Buxton. *Sketching User Experiences*. DianeCerra, 2007.
- [4] CBS. *Statistical programs for PX-files*. 2019. URL: <https://www.scb.se/en/services/statistical-programs-for-px-files/>.
- [5] Shaumik Daityari. *Angular vs React vs Vue: Which Framework to Choose in 2020*. 2019. URL: <https://www.codeinwp.com/blog/angular-vs-vue-vs-react/>.
- [6] Breaking Projects Down. *What is Agile*. 2019. URL: <http://www.agilenutshell.com/>.
- [7] Petros Likidis. *PX-file format specification*. 2013. URL: https://www.scb.se/globalassets/vara-tjanster/px-programmen/px-file_format_specification_2013.pdf.
- [8] Petros Likidis. *PC-Axis API description*. 2014. URL: http://pxnet2.stat.fi/API-description_SCB.pdf.
- [9] Petros Likidis. *Px suite (former PcAxis suite)*. 2019. URL: http://www.stat.fi/tup/tilastotietokannat/px-tuoteperhe_en.html.
- [10] Petros Likidis. *PxWeb API*. 2019. URL: <http://pxnet2.stat.fi/api1.html>.
- [11] Kathryn McElroy. *Prototyping for Designers*. O'REILLY.
- [12] John Ryding Olsson Mette Lindegaard Attrup. *Power i projekter & portefølje*. 2nd. Jurist- og Økonomforbundets Forlag, 2008.
- [13] Anna Nystrom. *Agile Solo*. 2011. URL: <http://publications.lib.chalmers.se/records/fulltext/147143.pdf>.
- [14] Aashish Pahwa. *What Is A Prototype? - Examples, Types and Qualities*. 2019. URL: <https://www.feedough.com/what-is-a-prototype/>.
- [15] Daniel Parker. *JavaScript with Promises*. First Edition. O'REILLY, 2015.
- [16] Bruse R. Maxim Roger S. Pressman. *Software Engineering - A Practitioner's Approach*. Eight Edition. McGraw-Hill Education.
- [17] Sophia. *Angular vs React vs Vue: Which is the Best Choice for 2019?* 2019. URL: <https://hackernoon.com/angular-vs-react-vs-vue-which-is-the-best-choice-for-2019-16ce0deb3847>.
- [18] Rachel Thompson. *Stakeholder Analysis - Project Management Tools From MindTools.com*. 2019. URL: https://www.mindtools.com/pages/article/newPPM_07.htm.

- [19] Tutorialpoint. *Typescript - Overview*. URL: https://www.tutorialspoint.com/typescript/typescript_overview.htm.