

UNIVERSITY OF TARTU
Institute of Computer Science
Informatics Curriculum

Anneli Klamas

**Reinforcement stirrups drawing web
application for Eesti Traat**

Bachelor's Thesis (9 EAP)

Supervisor(s): Manuel Alejandro Camargo Chávez

Tartu 2021

Reinforcement stirrups drawing web application for Eesti Traat

Abstract:

Customers of Eesti Traat often send drawing of the stirrups made by hand. These drawings are not always accurate, which can lead to time and monetary losses if the wrong ones are produced due to the misunderstandings. In response to the above problems, it is possible to find different software solutions. But existing applications can have expensive licensing, require training and specialized technical knowledge, include functionalities not required for the specific purpose, or do not make drawings with right proportions. The result of this thesis is a web application for drawing and sending price inquiries of the standard and custom reinforcement stirrups. Logged in moderators and admins can change the information about price and wires. Admins can also add and delete users. During the development Bootstrap 4, FabricJs, Spring and Vectr were used.

Keywords: Web application, Bootstrap, FabricJs, Spring, Vectr, drawing

CERCS: P170

Veebirakendus rangide joonestamiseks Eesti Traadile

Lühikokkuvõte:

Eesti Traati kliendid saavad tihti käsitsi valmistatud rangide jooniseid. Need joonised ei ole alati täpsed, mis võib põhjustada aja- ja rahalisi kaotusi, kui arusaamatuste tõttu tekivad valed joonised. Vastuseks ülaltoodud probleemidele on võimalik leida erinevaid tarkvaralahendusi. Kuid olemasolevatel rakendustel võib olla kallis litsents, vajavad koolitust ja spetsialiseeritud tehnilisi teadmisi, võivad sisaldada funktsioone, mis pole konkreetseks otstarbeks vajalikud, või ei tee õige proportsiooniga jooniseid. Selle lõputöö tulemuseks on veebirakendus standardsete ja kohandatud tugevdatud rangide joonestamiseks ja hinnapäringute saatmiseks. Sisseloginud moderaatorid ja administraatorid saavad muuta teavet hinna ja traatide kohta. Administraatorid saavad kasutajaid ka lisada ja kustutada. Arenduse käigus kasutati Bootstrap 4, FabricJ, Spring ja Vectr.

Võtmesõnad: Veebirakendus, Bootstrap, FabricJs, Spring, Vectr, joonestamine

CERCS: P170

Contents

1. Introduction	5
2. Background and related concepts	7
3. State of the art analysis	9
3.1 General-purpose tools	10
3.2 Specialized tools	10
4. Technologies	12
4.1 Bootstrap 4	12
4.2 FabricJs	13
4.3 Spring	14
4.4 Vectr.....	15
5. Application architecture and design	17
5.1 Functional requirements.....	17
5.2 Non-functional requirements.....	18
5.3 Client-side solution	19
5.3.1 Drawing shapes.....	19
5.3.2 Sending price inquiries.....	21
5.3.3 Updating data and role-based access.....	22
5.4 Server-side solution	23
5.4.2 REST controllers.....	25
5.4.3 Services	25
5.4.3 Data Transfer Objects.....	26
5.4.1 Database model.....	27
5.5. Tests	28
References.....	30
Appendix.....	34

I. Source code	34
II. Licence	35

1. Introduction

Eesti Traat is an Estonian company that was established in 2005 [1]. The company specializes in manufacturing and selling wire structures for reinforcing concrete buildings. Their product range includes reinforcement nets, reinforcing steel, bi-reinforcement, wall mesh, bars, spacers, tri-reinforcement, trail tiles, steel profiles, and accessories for concreting works. Their main customers are construction companies in the Baltics and Scandinavia. They receive twenty to fifty price inquiries and orders daily [2].

One of the company's main problems when processing the volume of inquiries is the unclear specification by the customer. Most common misunderstandings come when customers draw stirrups by hand; usually, they do not show if measurements are taken from outside or inside of the stirrup. Moreover, sometimes customers want to order something that the manufacturer can not produce. Due to these inaccuracies, the manufacturer must rectify multiple times what the customer needs, slowing down the processing of the orders. There have been cases where the manufacturer produces the wrong stirrups even after elaborating with the customer, which ends up with the entire batch going to waste. In such cases, it is unclear whose fault it is, and the manufacturer must produce new ones assuming the cost and time expended. These situations usually occur two times a month.

In response to the above problems, it is possible to find in the market two types of software solutions: multifunctional and domain-specific. Multifunctional products seek to support the design and measurement of all types of structures the most known are Allplan 2021 and AutoCAD. However, despite the high technical level of these tools, they are inadequate stirrup design due to several reasons. First, their licensing is expensive; second, they require training and specialized technical knowledge; and third, they include functionalities not required for the specific purpose. In the case of the domain-specific products like RebarWeb, Graphico Pro, Autorebar, RGS REBAR, and RebarCAD that are designed to draw explicitly stirrups, the problems are that they need some other software like AutoCAD to use them, or the measurements of the stirrups do not correspond to their proportions.

This thesis aims to develop a web application to design and send price inquiries for reinforcement stirrups. This application seeks to prevent misunderstandings between customers and the manufacturer. The customer will draw his stirrup and see how it looks like; meanwhile, the application will take the measurements according to the manufacturer's requirements. Additionally, once the customer confirms the order, the manufacturer will directly get his

drawing and specifications to type in the bending machine, avoid economic losses due to mistakes and reprocessing.

2. Background and related concepts

Reinforcement stirrups are bent and cut wires that engineers and builders use in concrete buildings to hold reinforcement bars in place. They often use them in beam reinforcement [3]. Figure 1 presents the machine used to produce Stirrups in Eesti Traat. Machines have two parts that operators can change: wire coils and bending mandrel. When the needed wire coil and bending mandrel is placed, then the machine operators can start defining stirrup to the machine (see Figure 2). Every row in the recipe table represents one line that the machine will produce. In the first input column operator establishes the length of the line, in the second column the bending direction, and in the third column the bending degree. With every input row, operators see how the stirrup will look like and can change the input if there is a mistake. Operators can also define arcs by selecting arc and then writing in the first column the length of the arc, in the second column the bending direction, and in the third column the radius of the arc. Usually, they know only the bending direction, radius, and how many degrees should the sector of the arc be. It means that operators need to calculate the length of the arc.

When operators finish inserting data into the machine, they define how many pieces they need and start the machine. The machine will then start pulling the wire, bending it, and cutting it (see Figure 3). Stirrup will then fall on the floor or in the collector.



Figure 1. Example of the stirrup making machine. Schnell Prima 12 [4].

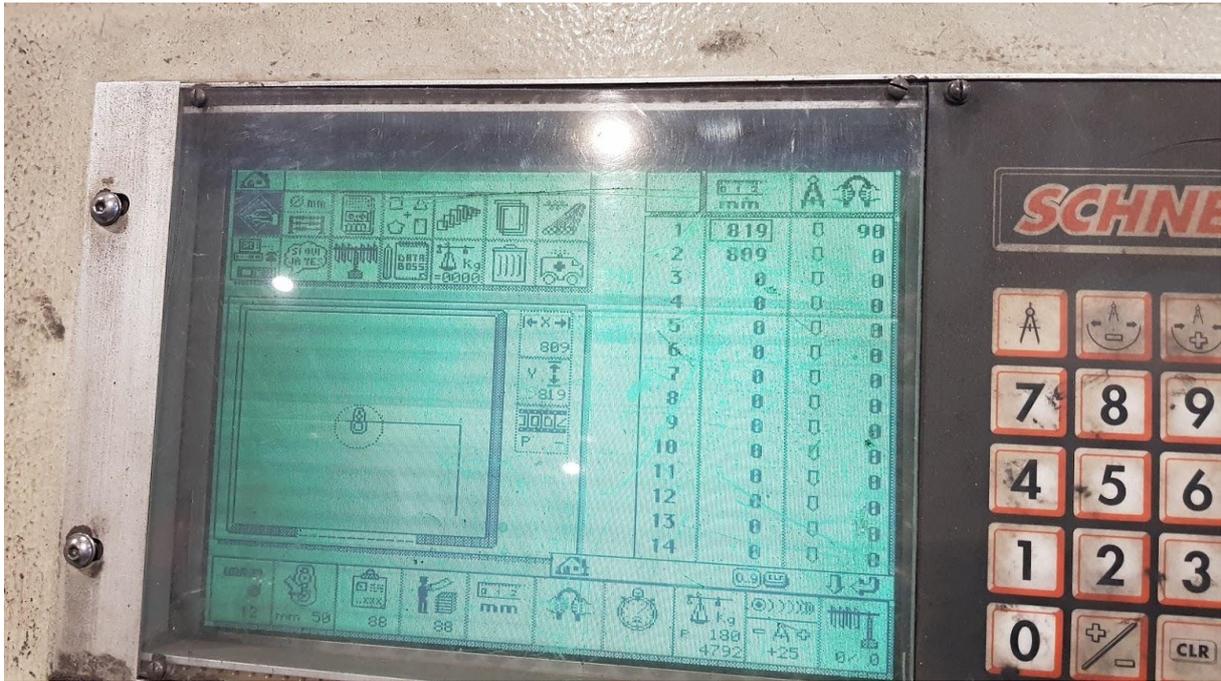


Figure 2. Example of the machine input.

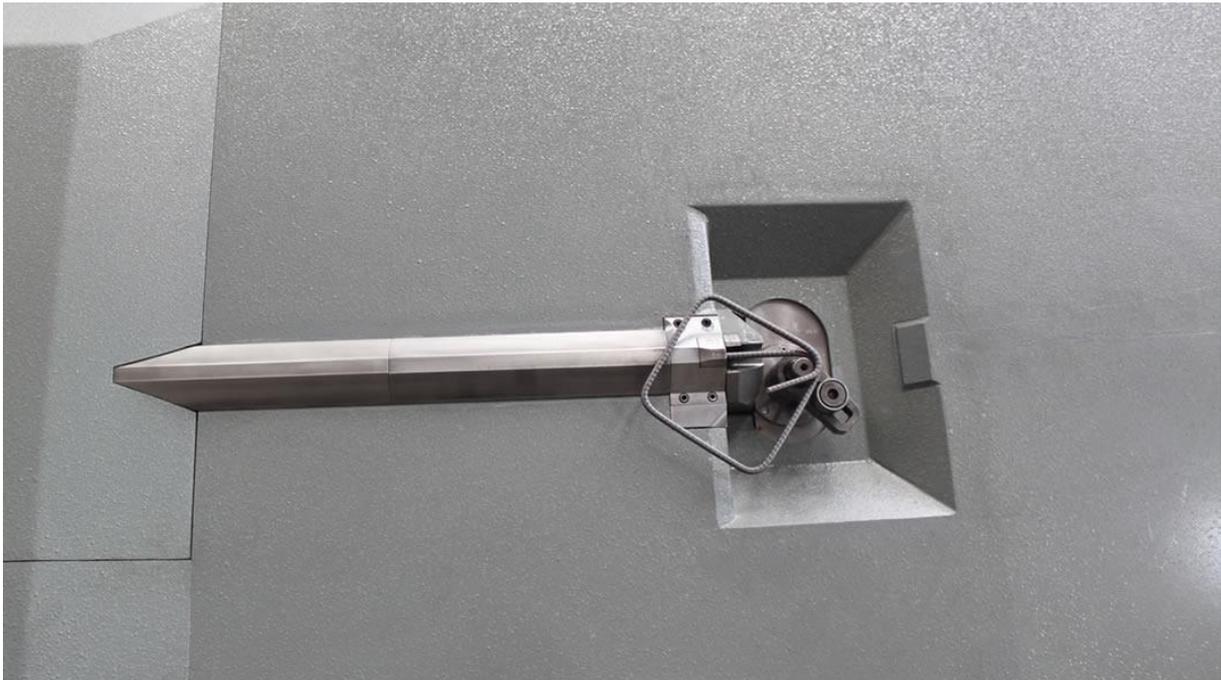


Figure 3. Close-up look of the example of bending and cutting a stirrup [5].

The stirrups producing process starts when customers send drawings of the stirrups, which are usually drawn by hand, to the salesperson, who calculates the price and sends it to the customer. Once the customer confirms the price, then the drawing of the stirrup is forwarded to the secretaries. They will make an accurate drawing with dimensions and give it to the machine operators, who will translate it to the machine by drawing it line by line in the machine software.

3. State of the art analysis

We divided the software applications on the market into general-purpose (GP) tools and Specialized (S) tools for analysis purposes. We compare a total of seven applications: AutoRebar [6], AutoCAD [7], RebarWeb [8], Graphico Pro [9], Allplan 2021 [10], and RebarCAD [11]. Table 1 shows the comparison of their licensing characteristics and drawing capabilities. The *Application* column provides the name of the product. In the *Category* column, we specify if the product is a GP or an S tool. The *Type* column categorizes the application on four main types: Web Application (WA), Installable Program (IP), Plugin (P), and Mobile Application (MA).

Regarding the licensing information, the *Term* column shows how extended the product license lasts: lifetime, monthly, one year, N/A if there is no public information. The *Cost* column shows the price of the license. Regarding the drawing capabilities, the *Templates* column shows if the application supports a wide range of standard stirrups' templates, the *Draw stirrups* if the application can draw customized stirrups, the *3D drawing* shows if the application support 3D shapes, and the *Proportions* if the proportions correspond to the actual dimensions. For these categories, we mark (+) if the feature is supported and (-) if it is not.

S/N	Application	Category	Licencing			Capabilities			
			Type	Term	Cost	Templates	Draw stirrups	3D drawings	Proportions
1	Autorebar	S	P	Lifetime	112\$	-	+	+	+
2	RebarWeb (free)	S	WA	None	None	+	-	-	-
3	RebarWeb	S	WA	Monthly	15€-100€	+	+	-	-
4	Graphico Pro	S	IP	N/A	N/A	+	+	-	-
5	Allplan Engineering Building 2021	GP	IP	N/A	N/A	-	+	+	+
6	RGS REBAR	S	P	Monthly	75\$	-	+	-	+
7	AutoCAD	GP	IP-WA-MA	One year	\$1,690	-	+	+	+
8	RebarCAD	S	P	N/A	N/A	+	+	-	+

Table 1. Comparison table.

3.1 General-purpose tools

General-purpose tools help to do a wide range of things like modeling different constructions, calculating how many and which types of reinforcement are needed, or drawing a wide variety of different shapes. In this category, we find AutoCAD and Allplan. AutoCAD is a 2D and 3D designing software meant for architects, engineers, and construction professionals. Users can use it as a desktop, web, or mobile application. It costs \$1,690 (~1409,50€) per year. It is not very easy to use because it has many options, so it is easy to get confused about which AutoCAD tools are needed [7]. Allplan Engineering Building 2021 is a 2D and 3D designing software for structural engineers. It helps plan which types of reinforcement construction are needed by specifying the reinforcement parameters such as minimum and maximum spacing, minimum diameter or permissible reinforcement ration, and offsets [10]. In general terms, this type of tool is pricy and is meant for big companies and engineers with domain knowledge, making them unsuitable for clients of Eesti Traat.

3.2 Specialized tools

Specialized tools draw only rebars or draw rebars and help with some other small tasks like achieving the schedules. In this category, we find multiple AutoCAD plugins, RebarWeb and Graphico Pro. We analyze two AutoCAD plugins AutoRebar and Viskartech's RGS REBAR. AutoRebar is a tool designed to make drawing rebars easy in AutoCAD. It has a lifetime license that costs 112\$ (~93,39€) [12]. Viskartech's RGS REBAR is a plugin for AutoCAD and ZWCAD, designed to help make 2D rebar detailing and bar bending schedules. It costs 75\$ [13]. Both tools are cheap, but they require an AutoCAD license, which costs \$1,690 per year and makes these tools expensive and therefore unsuitable for Eesti Traat [14].

RebarWeb and Graphico Pro are tools developed by Schnell Software, and their primary purposes are managing, scheduling, and detailing customer orders and the production assignment. RebarWeb is the newer web version of the Graphico Pro. Everyone can use RebarWeb by creating an account and choosing the package that best suits them. In Graphico Pro it is needed to send a price inquiry to the company to get the information about the prices of the application and the process of buying it. In both RebarWeb's not open source packages and GraphicoPro, it is possible to draw your stirrups just by drawing lines and then adding measures over them. However, this feature has a problem; when the user adds measurements to the shape, the proportions of the stirrup do not change, leading to drawings that can not exist in real life (see Figure 4) [8] [9].

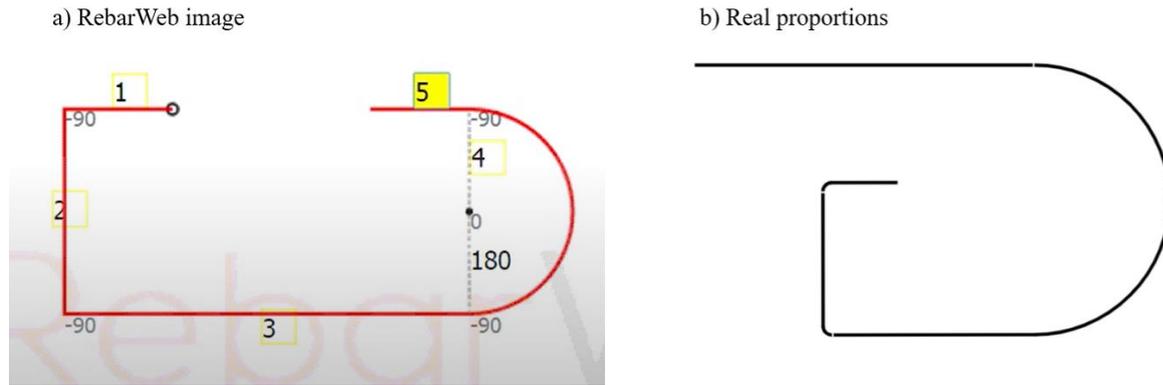


Figure 4. RebarWeb proportions and real proportions comparison [15].

In summary, existing programs are either AutoCAD extensions, which costs \$1,690 per year, or tools where reinforcement stirrups drawing is not the primary purpose. All programs that can help draw customized stirrups are not open source preventing its adaptation to the needs of Eesti Traat.

The cheapest program that can draw customized stirrups is RebarWeb and costs 15€ per month. However, the impossibility of maintaining the correct proportions would be confusing for customers, which would lead to relapse into the same reprocessing problems that the company currently faces.

4. Technologies

4.1 Bootstrap 4

Since Eesti Traats customers use all kinds of devices to access the web page, we choose Bootstrap to make the app responsive. We choose Bootstrap due to its extending use and community support in web development. Bootstrap is a toolkit that helps to make responsive mobile-first websites. It has many designed components, so users do not have to define all components in CSS to have a modern-looking website. It also has its open-source SVG icon library with over 1,300 icons that users can with or without Bootstrap [16] [17].

To make a website responsive, Bootstrap has a grid system (see Figure 5) that uses a series of containers, rows, and columns, which users can combine to make wider columns [18].

span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1
span 4				span 4				span 4			
span 4				span 8							
span 6						span 6					
span 12											

Figure 5. Bootstrap grid [19].

Bootstrap also has five grid options for different screen sizes (see Figure 6). If the container is larger than defined in the grid column, then alignment will not change, but if it is smaller, then the columns will be presented above each other (see Figure 7) [20].

	Extra small <576px	Small ≥576px	Medium ≥768px	Large ≥992px	Extra large ≥1200px
Max container width	None (auto)	540px	720px	960px	1140px
Class prefix	.col-	.col-sm-	.col-md-	.col-lg-	.col-xl-
# of columns	12				
Gutter width	30px (15px on each side of a column)				
Nestable	Yes				
Column ordering	Yes				

Figure 6. Grid options [21].

a) Screen is more than 578px wide

```

<div class="row">
  <div class="col-sm-3" style="background-color:lavender;">.col-sm-3</div>
  <div class="col-sm-3" style="background-color:lavenderblush;">.col-sm-3</div>
  <div class="col-sm-3" style="background-color:lavender;">.col-sm-3</div>
  <div class="col-sm-3" style="background-color:lavenderblush;">.col-sm-3</div>
</div>

```

b) Screen is less than 578px wide

```

<div class="row">
  <div class="col-sm-3" style="background-color:lavender;">.col-sm-3</div>
  <div class="col-sm-3" style="background-color:lavenderblush;">.col-sm-3</div>
  <div class="col-sm-3" style="background-color:lavender;">.col-sm-3</div>
  <div class="col-sm-3" style="background-color:lavenderblush;">.col-sm-3</div>
</div>

```

Figure 7. Example code of responsive columns with different screen sizes [22].

4.2 FabricJs

Since in native <canvas> API, it is impossible to modify defined objects. We needed to find a library where it would be possible. We chose FabricJs because it provides functions for moving, resizing and rotating objects before and after they are drawn.

FabricJs is an open-source javascript library for making drawings. Its most significant advantage is that the user can group all specified objects into one, allowing their manipulation at once. It is easy to scale, move and draw objects with FabricJs's functions. It also has an SVG parser so that drawings can be exported from or imported into the canvas. Drawing shapes with FabricJs is similar to drawing with native <canvas> API. An example of drawing a red square

with native <canvas> API is presented in Figure 8, where after defining the canvas, the other parameters are specified, for example, color. Then the rectangle is drawn [23].

a) Creating red square with native <canvas> API

```
// reference canvas element (with id="c")
var canvasEl = document.getElementById('c');

// get 2d context to draw on (the "bitmap" mentioned earlier)
var ctx = canvasEl.getContext('2d');

// set fill color of context
ctx.fillStyle = 'red';

// create rectangle at a 100,100 point, with 20x20 dimensions
ctx.fillRect(100, 100, 20, 20);
```

b) Creating red square with FabricJS

```
// create a wrapper around native canvas element (with id="c")
var canvas = new fabric.Canvas('c');

// create a rectangle object
var rect = new fabric.Rect({
  left: 100,
  top: 100,
  fill: 'red',
  width: 20,
  height: 20
});

// "add" rectangle onto canvas
canvas.add(rect);
```

Figure 8. Drawing a square with native <canvas> API and with FabricJS [24].

In FabricJS, after defining canvas, the rectangle object is created, and all the parameters like color and measurements are specified in it (see Figure 8).

4.3 Spring

Spring is a lightweight open-source framework that is based on Java platform. Spring is popular due to it being organized in a modular fashion; adding and using extensions on top of frameworks core will enable to develop more than Java applications. Spring has a very well-designed web framework that provides tools and components necessary for developing flexible web applications. The framework is based on Model-View-Controller architecture, which makes it easy to make the developed applications scalable. It is also possible to integrate it with other existing technologies. That is why we built our server-side solution entirely on Spring. We used it to make and change databases, role-based access, and send emails. Since it is the most popular application development framework for enterprise Java, it has many tutorials, examples, and answers to the questions on different websites, making it easy to find information about how to make something with it [25, 26].

Spring is an open-source application that makes it easy to create Java enterprise applications. It also supports Kotlin and Groovy languages. Spring has a big active community because it supports a lot of different application scenarios. It started when Spring Framework came to the market in 2003. Now it has evolved, and Spring is not just Spring Framework but has over 21 other projects. They include Spring Boot, Spring Security, Spring Data, and Spring HATEOAS. Every project has its source code repository, issue tracker, and release cadence [27].

Spring Framework has over 20 different modules that are organized in bigger groups (see Figure 9). Data Access / Integration layer has Java Database Connectivity (JDBC), Object-relational mapping (ORM), Java Message Service (JMS), Object/XML Mapping (OXM) and Transactions modules. In the Web layer it has WebSocket, Servlet, Web and Portlet modules. The Core Container consists of Beans, Core, Context and Expression Language (SpEL) modules. Framework has also Test, Aspect-oriented programming (AOP), Aspects, Instrumentation and Messaging modules [28].

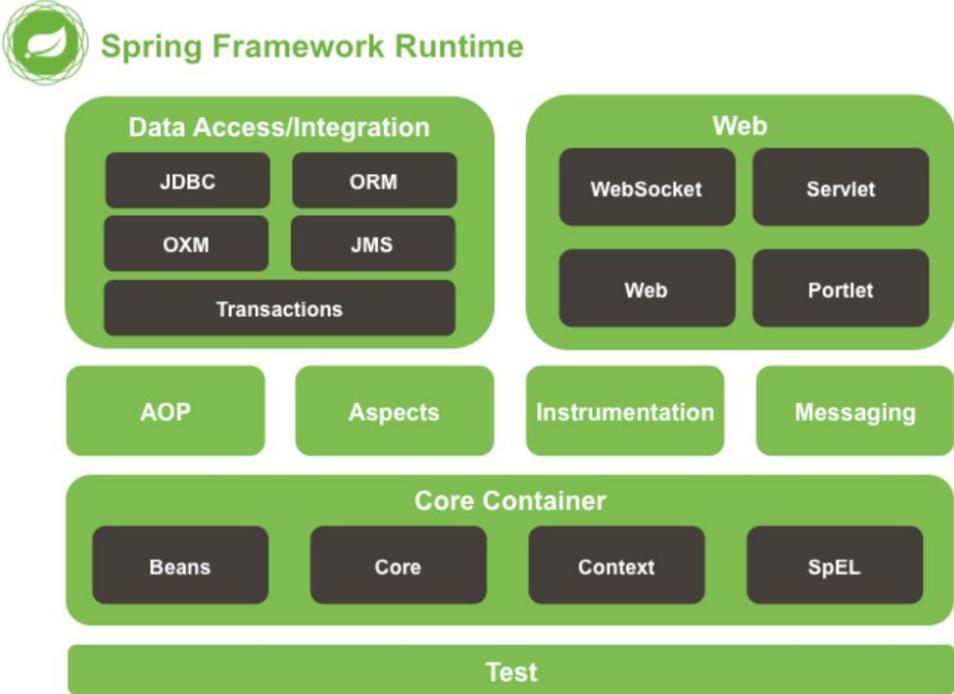


Figure 9. Spring Framework architecture.

4.4 Vectr

Since it is hard to find a circle drawing compass or a stirrup SVG icons that also would be open source, we decided to make all the icons in our web application ourselves. For that, we used Vectr.

Vectr is open source software for creating vector graphics. It is available as a web application and as a downloadable program. With Vectr, users can create their SVG, png, and jpg images by uploading and updating their existing photos, drawing by hand, writing or using different shapes and symbols, such as rectangles, curved lines, circles, stars, and envelopes. The user interface is intuitive and easy to use (see Figure 10). It is unnecessary to make a user account to use drawing functionality, but users can not save their drawings without being logged in. [29]

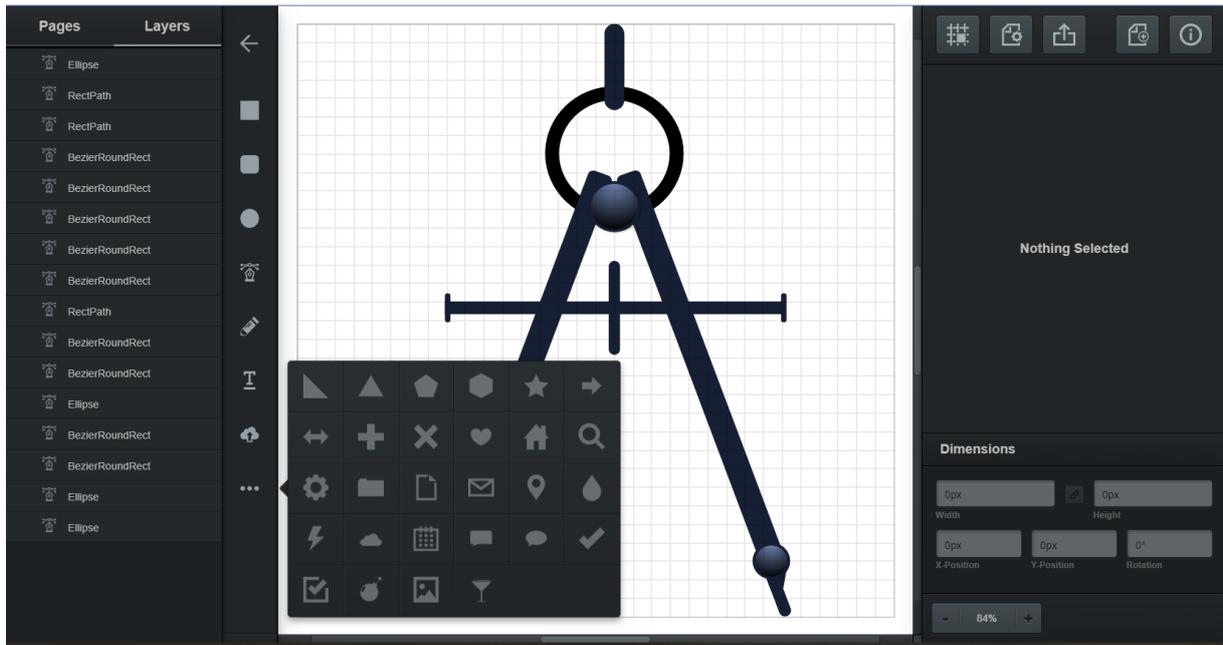


Figure 10. Vectr drawing UI.

5. Application architecture and design

5.1 Functional requirements

We created the functional requirements (Table 2) based on several interviews conducted with one of the board members of Eesti Traad. After making the initial requirements, mockups of the web application were created, shown to the representative of Eesti Traad, which helped us complement the requirements. We also visited the manufactory to get a better understanding of the process of making a reinforcement stirrup.

ID	Requirement
1	As a customer, I want to see the weight of the stirrup so that I could calculate the price
2	As a customer, I want to see the total length of the wire that went to the production of the stirrup so that I could calculate the price.
3	As a customer, I want to have a tutorial/video so that I could have a better understanding of how an application works.
4	As a customer, I want to have a graphical output for each length and angle pair I add to understand better how my stirrup looks.
5	As a customer, I want to update any creation step avoiding starting all over when I make a mistake in the middle of the drawing.
6	As a customer, I want to delete all the stirrup creation steps avoiding deleting items one by one if I want to start over.
7	As a customer, I want to add different stirrups to one price inquiry so that I do not need to make different price inquiries to each stirrup.
8	As a customer, I want to add the number of stirrups to the price inquiry to avoid making multiple requests for each stirrup I need.
9	As a customer, I want to choose the diameter of the wire.
10	As a customer, I want to choose the standard of bending roll.

11	As a customer, I want to choose between standardized stirrups, so I do not have to draw all the details by myself.
12	As a customer, I want to choose in which direction the wire is bent to make the right shape. (bending can only happen for 180 degrees)
13	As a customer, I want to make stirrups with rounded elements with different diameters and sizes.
14	As a customer, I want to identify if lines overlap with each other using a line slightly above the other to see that the line exists.
15	As a customer, I want a confirmation letter to be sure that my message was sent successfully
16	As a manufacturer, I want to have a stirrup recipe with the price inquiry to feed it into the machine immediately.
17	As a manufacturer, I want quotations to have information about clients (email, firm, phone number, name) so that I could contact them if needed
18	As a moderator, I want to be able to change the steel price per kg and wire information (diameter, EN and FIN standard rolls, EN and FIN length per degree, weight per meter) so that it would always be updated.
19	As an admin, I want to be able to add and delete users.

Table 2. Functional requirements.

5.2 Non-functional requirements

ID	Requirement
NF 1	Embed application into the current website, which is made with WordPress.
NF 2	Only authorized users can update tables with users, wires and price informations.

Table 3. Non functional requirments.

5.3 Client-side solution

5.3.1 Drawing shapes

One of the main functionalities of the application is drawing stirrups¹. However, before starting to draw stirrups, the users must choose the wire diameter and whether the stirrup will be bent according to English or Finnish standard. After that, users can choose to create a standard stirrup between 22 different shapes or choose a custom-shaped stirrup. The standard shape templates save time for the customer because it avoids thinking about how to draw the needed shape. If customers choose a standard shape, then all they need to do is to specify measurements and click “draw” (see Figure 11).

An example of the standard stirrup drawing is provided in Figure 11. In this figure, there is an image of standard C with letters. Each letter corresponds to one measurement that the user needs to specify to make a stirrup. The customer filled the table with sizes, and the program drew stirrup.

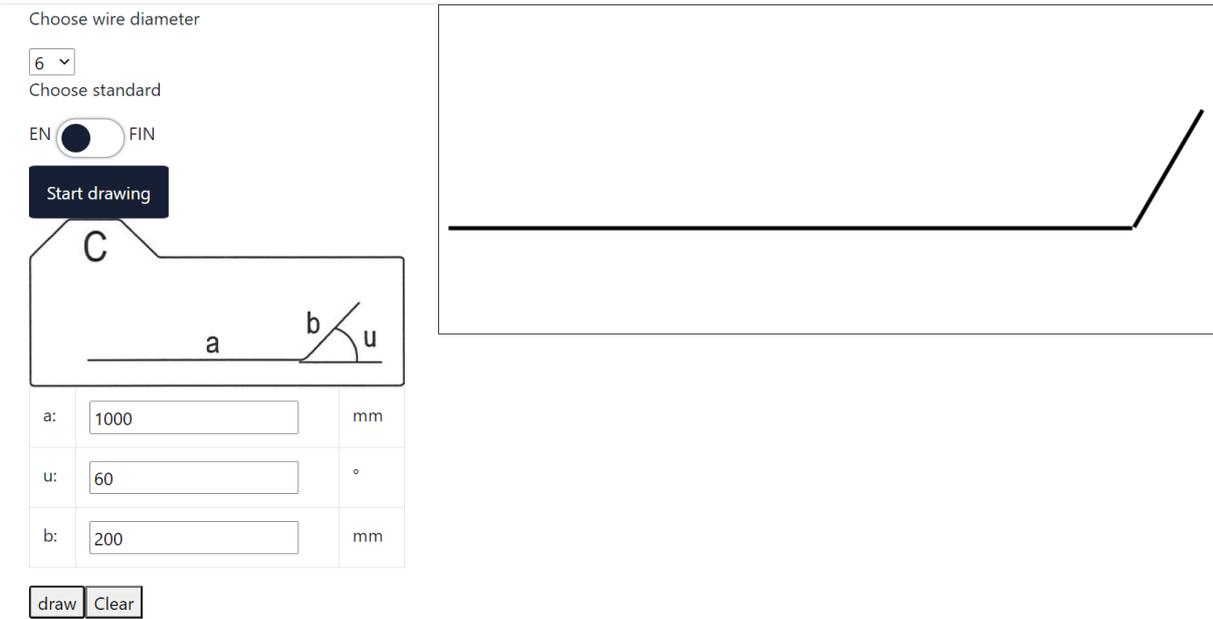


Figure 11. Stirrup with standard C

If customers can not find the needed shape, they can draw a custom shape. For that, they need to fill the table where every line corresponds to one drawn line. The table has four columns. The first column shows the row number, in the column “circle”, customers can specify if they want to make an arc or straight line. The third column “mm” specifies the length of the line in millimeters. The next column, “dw/up” gives the customer option with bending direction. In

¹ Linked functional requirements (4-6, 9-14)

this column, customers can choose if they want the wire to be bent up or down. The last column specifies how many degrees the line will be bent. If the user chooses to draw a rounded line, this column must contain information about the desired arc radius. It is also important that lines that are behind each other are not on the same line to see that they exist. An example of the customer stirrup drawing with overlapping lines and one rounded line is provided in Figure 12.

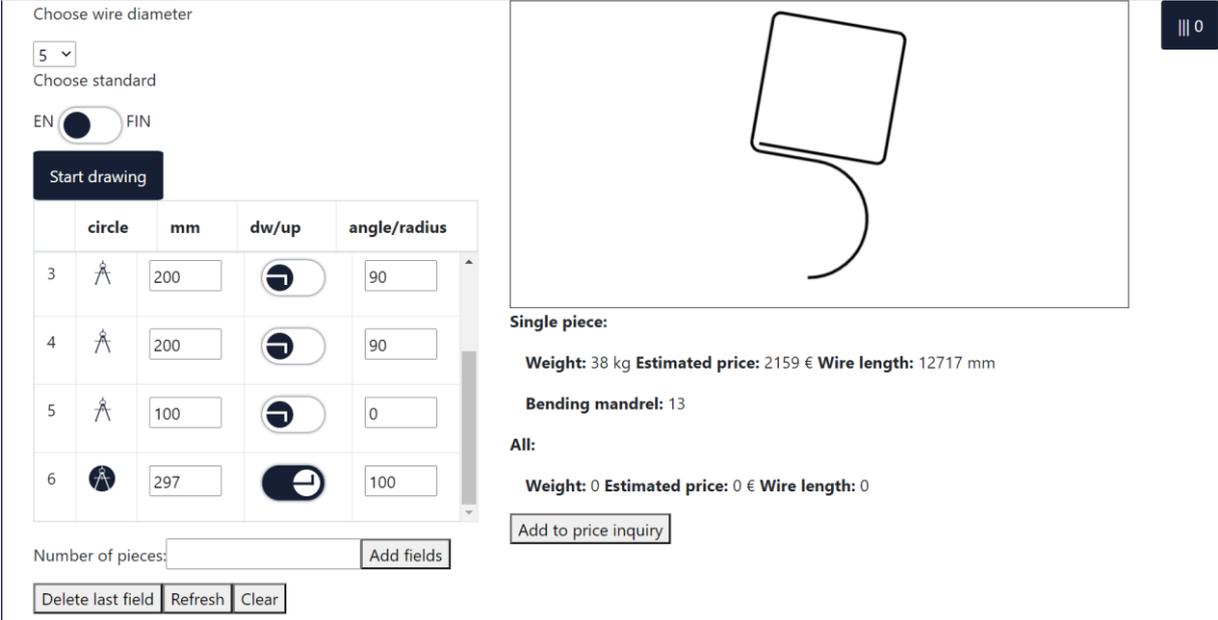


Figure 12. Custome drawing with overlapping line and arc.

5.3.2 Calculations

One big part of the application is making calculations². The application can calculate the height and width of the stirrup, the length of the wire, and the weight of the stirrup. Calculations are needed so that customers can check if all the measurements are correct and calculate prices, as the steel price changes every month. An example of the table with calculations is provided in Figure 13.

Single piece:
Weight: 27 kg **Estimated price:** 1535 € **Wire length:** 9040 mm
Bending mandrel: 13

All:
Weight: 5424 **Estimated price:** 306998 € **Wire length:** 1808000

Figure 13. Calculation table.

² Linked functional requirements (1-3)

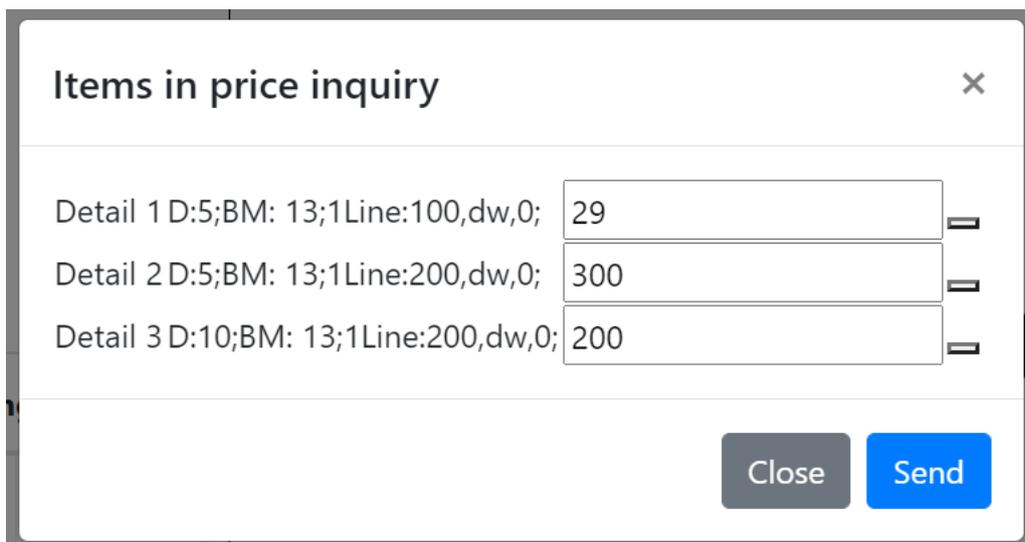
5.3.2 Sending price inquiries

To make an order customers need to send price inquiries³ to the company. Customers can add drawn stirrups to the price inquiry. For that, they need to specify the amount of the stirrups and click “Add to price inquiry”. After that, the price inquiry will be updated, and the button, which opens a detailed view of the inquiry, will show how many different stirrups are in the inquiry. An example of the price inquiry button is provided in Figure 14.



Figure 14. Button for opening price inquiry details.

When all the stirrups are added in the inquiry, customers can open the detailed view, where they can delete different stirrups and change the amount of the stirrups. An example of the price inquiry table is provided in Figure 15.

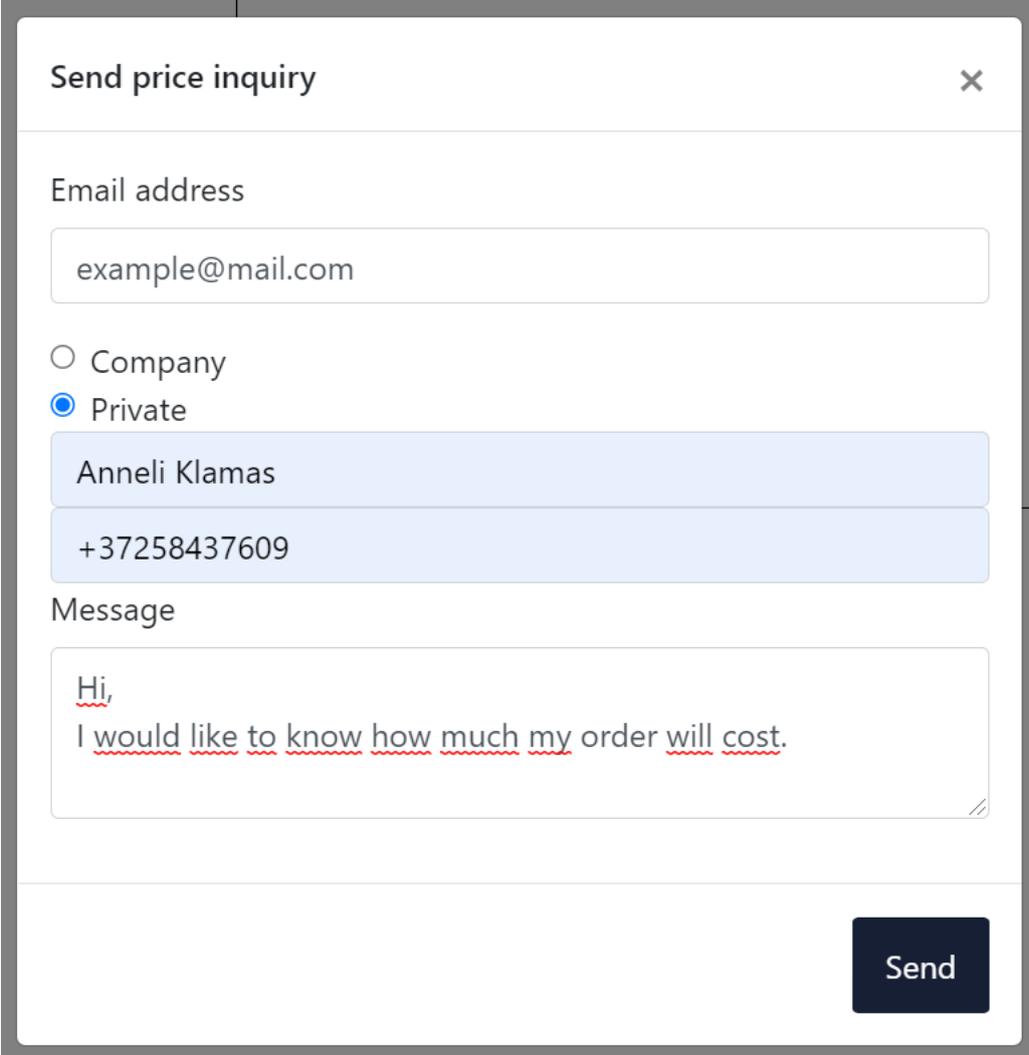


Items in price inquiry		×
Detail 1 D:5;BM: 13;1Line:100,dw,0;	<input type="text" value="29"/>	▬
Detail 2 D:5;BM: 13;1Line:200,dw,0;	<input type="text" value="300"/>	▬
Detail 3 D:10;BM: 13;1Line:200,dw,0;	<input type="text" value="200"/>	▬

Figure 15. Price inquiry table.

³ Linked functional requirements (7-8,15-17)

After checking that all the details are correct, customers can send the price inquiry. For doing that, they need to click the “Send” button and fill in the data in the contact form. An example of the filled contact form is provided in Figure 16.



The image shows a web form titled "Send price inquiry" with a close button (X) in the top right corner. The form contains the following fields and options:

- Email address:** A text input field containing "example@mail.com".
- Radio buttons:** Two options are present: "Company" (unselected) and "Private" (selected).
- Text fields:** Two light blue highlighted text boxes. The first contains "Anneli Klamas" and the second contains "+37258437609".
- Message:** A text area containing the text "Hi, I would like to know how much my order will cost." The words "I would like to know how much my order will cost." are underlined in red.
- Send button:** A dark blue button with the text "Send" in white, located at the bottom right of the form.

Figure 16. Contact form with example data.

After filling in the data in the contact form and clicking “Send,” two emails will be sent. One will go to the customer address on the form to confirm that the price inquiries were sent, and the other to the company.

5.3.3 Updating data and role-based access

To keep all data about prices, wires, and standards up to date. Workers need to be able to modify the inquire data⁴ by logging in into the settings account. Once they have successfully logged in, they can select whether they want to update wires or prices. In the case of being an admin

⁴ Linked functional requirements (18-19)

additionally, the user can update the users' rights. An example of the wires table is presented in **Tõrge! Ei leia viiteallikat..** Moderators can then click on the wire row they want to update and choose the desired action (see Figure 17).

	ID	Diameter	Weight	EN standard roll	FIN standard roll	EN length per degree	FIN length per degree
1	1	5	245.55	13	14	32	45
2	2	6	245.55	13	14	35	67
3	3	7	245.55	13	14	75	54
4	4	8	245.55	13	14	98	87
5	6	10	245.55	13	14	64	56
6	7	11	245.55	13	14	67	43
7	8	12	245.55	13	14	12	45
8	13	9	245.55	13	14	32	64

Figure 17. Wire table.

Change wire ✕

Update
Add
Delete

Close

Figure 18. Options for wire.

5.4 Server-side solution

On our server-side solution we have REST controllers, services, data transfer objects, entities, repositories, and configuration files on the server-side. Our architecture model is presented in the **Tõrge! Ei leia viiteallikat..**

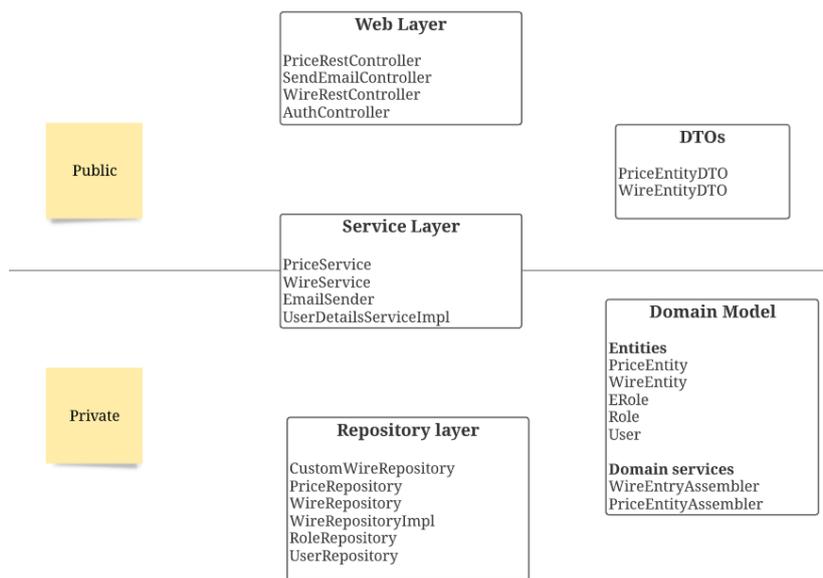


Figure 19. Architecture model.

The web layer in our solution is responsible for user interaction. This means that it will handle user inputs by processing them into corresponding outputs. Any server side exception is also handed by it. The main components in there are various controllers that are responsible for price and wire data inquiry, sending emails and authenticating users.

The service layer comes after the web layer and is responsible for being transaction layer, as explained by P. Kainulainen in his blog [31]. He says that, for this purpose it acts as a transaction boundary and contains application, infrastructure and domain services. The application services are responsible for handling the user authentication and providing public API for the service layer. According to him, the infrastructure service is responsible for communicating with external resources like file systems, databases and email services.

According to the P. Kainulainen's explanatory blog [31], the domain model consists of domain services, entities and value objects. He explains that, the domain services handle the business logic and provide operations for entities and value objects under domain model. The entity is an unchangeable identity object that remains unchanged throughout its lifecycle. As for value objects, they are entity-bound descriptions of an entity property. The repository layer is responsible for handling the communication with repositories and databases. In our solution the repository classes are each responsible for communicating with the right table in the database.

5.4.2 REST controllers

REST application programming interface, or API for short, is a set of definitions and protocols that allow for interacting with RESTful web services. It uses HTTP requests to move or manipulate data. We use Spring's RestController annotation, that is meant for building REST API. REST controllers used in our solution are responsible for handling HTTP request like GET, POST, DELETE and PATCH. These controllers are PriceRestController, SendEmailController, WireRestController and AuthController. They are responsible for forwarding user login data, email contents, wire and price inquiries to service layer.

For example the WireRestController (see **Tõrge! Ei leia viiteallikat.**) three GET method functions for getting all wires, wires by specified diameter and wires by id.

```
@CrossOrigin(origins = "http://localhost:63342")
@RestController
@RequestMapping("/wires")
public class WireRestController {

    @Autowired
    WireService wireService;

    //get all
    @GetMapping("")
    public CollectionModel<WireEntityDTO> findAll() { return wireService.findAll(); }

    //get with diameter
    @GetMapping("/diameter")
    public CollectionModel<WireEntityDTO> findWithDiameter(@RequestParam(name = "diameter" ) int diameter) {
        return wireService.findAvailableWire(diameter);
    }

    //get by id diameter
    @GetMapping("/id")
    public WireEntity findWithId(@RequestParam(name = "id" ) Long id) { return wireService.findEntryById(id); }
```

Figure 20. WireRestControllerExample.

WireRestController also has PATCH, DELETE and POST request methods. We used PATCH method to update wire, DELTE method to delete wire and POST to add wire.

5.4.3 Services

Services can be categorized as infrastructure, application or domain services. Infrastructor and application services are pary of the service layer. Domain services are part of the domain model.

One example of infrastructure service in our solution is WireService which is presented in Figure 21.

```

@Service
public class WireService {

    @Autowired
    WireRepository materialRepository;

    @Autowired
    WireEntryAssembler wireEntryAssembler;

    public WireEntity findEntryById(Long id) { return materialRepository.findById(id).orElse( other: null); }

    public CollectionModel<WireEntityDTO> findAvailableWire(int diameter) {
        List<WireEntity> entries = materialRepository.findAvailable(diameter);
        return wireEntryAssembler.toCollectionModel(entries);
    }

    public CollectionModel<WireEntityDTO> findAll() {
        List<WireEntity> entries = materialRepository.findAll();
        return wireEntryAssembler.toCollectionModel(entries);
    }
}

```

Figure 21. WireService.

5.4.3 Data Transfer Objects

Data transfer objects, or DTO for short, are a value objects that translates between different data structures in web layer and service layer. According to B. Krebs blog post [30], they can be viewed as design patterns that help reduce the number of remote calls. In our solution the DTOs are the PriceEntityDTO and the WireEntityDTO. Example of the WireEntityDTO is presented in Figure 22.

```

@Data
public class WireEntityDTO extends RepresentationModel<WireEntityDTO> {
    Long id;
    int diameter;
    Double weight;
    int enStandardRoll;
    int finStandardRoll;
    double enLengthPerDegree;
    double finLengthPerDegree;
}

```

Figure 22. WireEntityDTO.

5.4.1 Database model

In the Figure 23 database model with its relations is presented. Database has five tables: *MATERIAL_ENTITY*, *PRICE_ENTITY*, *ROLES*, *USER_ROLES*, and *USERS*.

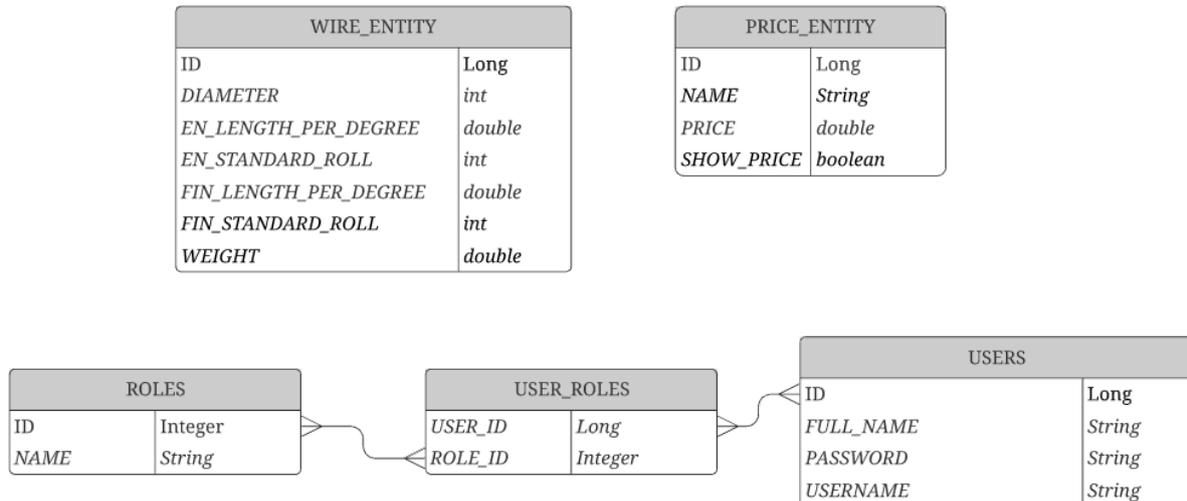


Figure 23. Database model.

MATERIAL_ENTITY table holds the information about wires. It has columns: *ID*, *DIAMETER*, *EN_LENGTH_PER_DEGREE*, *EN_STANDARD_ROLL*, *FIN_LENGTH_PER_DEGREE*, *FIN_STANDARD_ROLL*, and *WEIGHT*. The *ID* column holds value of the material id. In *DIAMETER* column we specify the diameter of the wire. The columns *EN_LENGTH_PER_DEGREE* and *FIN_LENGTH_PER_DEGREE* have information how many millimetres longer the wire gets with every degree it is bended according to its standard. In columns *EN_STANDARD_ROLL* and *FN_STANDARD_ROLL* is information about the diameter of the bending mandrel for the corresponding standard. The *WEIGHT* column has value, which shows how much one meter of the wire weights.

PRICE_ENTITY table has three columns: *ID*, *PRICE*, and *SHOW_PRICE*. The *ID* column has the id of the price. The *PRICE* column holds the price of the one kilogram of steel in euros. The *SHOW_PRICE* column has the information wheter the price should be displayed and calculated for the customer. If its value is *true*, then it will be shown to customers, if it is *false*, then not.

Tables *ROLES*, *USER_ROLES* and *USERS* are related to eachother. The *USERS* table holds user id in the *ID* column, users full name in the *FULL_NAME* column, encrypted password in its *PASSWORD* column, and *USERNAME* in its username column. The table *ROLES* holds in the *ID* column the id of the role and in the *NAME* column the name of the role. Tables *USERS*

and *ROLES* are connected with the table *USER_ROLES* which holds the information about which users which roles have.

In the future we are planning to save customers' price inquiries to the database and therefore connect *WIRE_ENTITY* with *PRICE_ENTITY* via *STIRRUP_ENTITY* (see Figure 24).

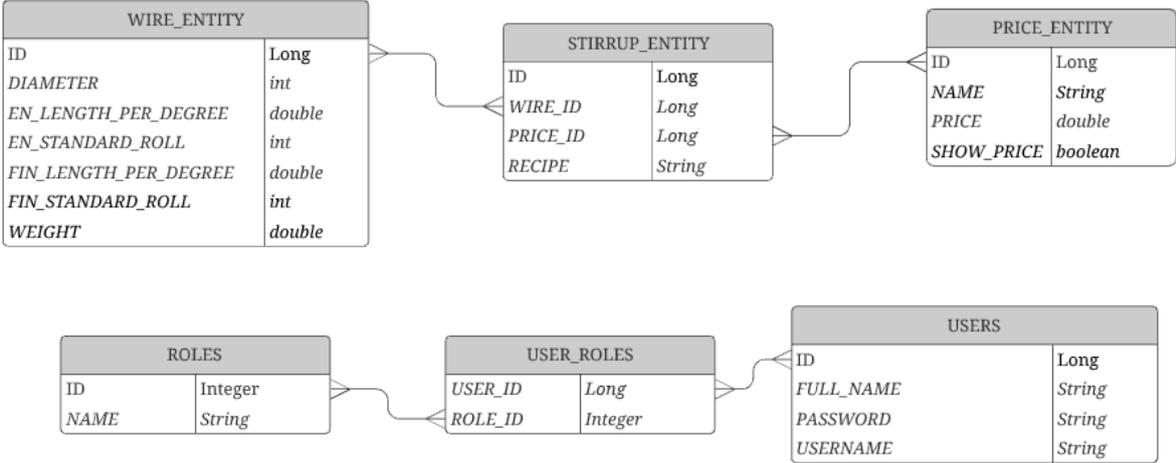


Figure 24. Future database model example.

5.5. Tests

The user acceptance tests were performed by the two salespersons. They used the application for 30 minutes each. They found it easy to use and intuitive because they know how machines take the input in and the stirrups. They were satisfied with the results.

As the development of the application continues, tests with customers will be performed.

6. Conclusion and future work

In the thesis, we describe the creation of a web application for Eesti Traat to draw standard and custom stirrups, make calculations, send price inquiries, and update data in the database.

As the main concept of the thesis was to make application for drawing custom stirrups, not all the functionalities are yet implemented. For example price inquiries for standard stirrups can not be send and not all the standard stirrupses can be made.

After the successful user acceptance test, the representatives of the Eesti Traat were satisfied with the achieved work and are willing to proceed with future implementations.

In the future, we will implement all the standard stirrup shapes. We will also add reinforcement meshes, lattice graders, and wires to the product list. We are also planning to make accounts for

customers with their order history and discounts. Furthermore, implement not just sending price inquiries but ordering and purchasing products. Also, the manufactory is improving their stock and order management program. We plan to integrate the drawing application not to have to write all the information to the program. Instead, we would generate XML specifications describing the order information that can be dragged and dropped directly to the order management program.

References

- [1] Creditinfo Eesti AS, “<https://www.e-krediidiinfo.ee/>,” Creditinfo Eesti AS, [Online]. Available: <https://www.e-krediidiinfo.ee/11195403-EESTI>. [Accessed 10 12 2020].
- [2] EESTI TRAAAT OÜ, “[eestitraad.ee](https://www.eestitraad.ee/),” EESTI TRAAAT OÜ, 2020. [Online]. Available: <https://www.eestitraad.ee/>. [Accessed 10 12 2020].
- [3] TataSteel, “What is the Main Purpose of Using Stirrups During Construction?,” [Online]. Available: <https://aashiyana.tatasteel.com/article/laying-the-foundations>. [Accessed 02 05 2021].
- [4] Schnellgroup, “Prima 12 R Bi-directional automatic stirrup bender from coil,” [Online]. Available: <https://www.schnellgroup.com/en/Used/Prima-12-R-Bi-directional-automatic-stirrup-bender-from-coil-13-mm/>. [Accessed 03 05 2021].
- [5] Schnellgroup, “Automatic stirrup bender from coil Prima 13,” [Online]. Available: <https://www.schnellgroup.com/en/Products/Automatic-stirrup-bender-from-coil-Prima-13/>. [Accessed 03 05 21].
- [6] Paolo Locatelli, “[autorebar.com](https://www.autorebar.com/),” 2020. [Online]. Available: <https://www.autorebar.com/what-is-it/>. [Accessed 10 12 2020].
- [7] Autodesk Inc, “AutoCAD,” [Online]. Available: <https://www.autodesk.com/products/autocad/overview?term=1-YEAR&support=null>. [Accessed 04 February 2021].
- [8] Schnell Software, “Schnell Software RebarWeb,” Schnell Software, 2018. [Online]. Available: <https://www.schnellsoftware.com/en/rebar-web-3/>. [Accessed 5 4 2021].
- [9] Schnell Software, “Schnell Software Graphico Pro,” Schnell Software, 2018. [Online]. Available: <https://www.schnellsoftware.com/en/graphico-pro-3/>. [Accessed 5 4 2021].

- [10 ALLPLAN Deutschland GmbH, “Allplan,” [Online]. Available:
] <https://www.allplan.com/products/allplan-2021/>. [Accessed 04 February 2021].
- [11 CADS, “RebarCAD,” CADS, 2017. [Online]. Available:
] <https://rebarcad.com/rebarcad/full-details/>. [Accessed 5 4 2021].
- [12 Autodesk, Inc, “Autodesk AutoRebar,” Autodesk, Inc, [Online]. Available:
] https://apps.autodesk.com/ACD/en/Detail/Index?id=3694790012131586411&appLang=en&os=Win32_64. [Accessed 4 February 2021].
- [13 RGS Construction Technologies, “RGS Rebar,” RGS Construction Technologies, 2021.
] [Online]. Available: <https://www.viskartech.com/global/products/rgsrebar/>. [Accessed 5 4 2021].
- [14 Autodesk, “apps.autodesk.com,” Autodesk, 2020. [Online]. Available:
] https://apps.autodesk.com/ACD/en/Detail/Index?id=3694790012131586411&appLang=en&os=Win32_64. [Accessed 10 12 2020].
- [15 Schnell Software S.L., “Creating shapes in Rebar Web,” [Online]. Available:
] [https://www.youtube.com/watch?v=kxefFCX5s1U&t=2s&ab_channel=SchnellSoftware S.L.](https://www.youtube.com/watch?v=kxefFCX5s1U&t=2s&ab_channel=SchnellSoftwareS.L) [Accessed 03 05 2021].
- [16 Bootstrap, “Bootstrap icons,” [Online]. Available: <https://icons.getbootstrap.com/>.
] [Accessed 30 04 2021].
- [17 Bootstrap, “Bootstrap website,” [Online]. Available: <https://getbootstrap.com/>. [Accessed
] 30 04 2021].
- [18 Bootstrap, “Grid system,” [Online]. Available:
] <https://getbootstrap.com/docs/4.0/layout/grid/>. [Accessed 03 05 2021].
- [19 W3Schools, “Bootstrap 4 Grid System,” [Online]. Available:
] https://www.w3schools.com/bootstrap4/bootstrap_grid_basic.asp. [Accessed 03 05 2021].

[20 W3Schools, “Bootstrap grid basic,” [Online]. Available:
] https://www.w3schools.com/bootstrap4/bootstrap_grid_basic.asp. [Accessed 03 05
2021].

[21 Bootstrap, “Grid system,” [Online]. Available:
] <https://getbootstrap.com/docs/4.0/layout/grid/> . [Accessed 03 05 2021].

[22 W3Schools, “Responsive Columns Example,” [Online]. Available:
] https://www.w3schools.com/bootstrap4/tryit.asp?filename=trybs_grid_ex1&stacked=h .
[Accessed 03 05 2021].

[23 FABricJS, “FabricJS introduction,” [Online]. Available: [http://fabricjs.com/fabric-intro-](http://fabricjs.com/fabric-intro-part-1)
] [part-1](http://fabricjs.com/fabric-intro-part-1). [Accessed 30 04 2021].

[24 FabricJs, “Introduction to Fabric.js. Part 1.,” [Online]. Available:
] <http://fabricjs.com/fabric-intro-part-1> . [Accessed 03 05 2021].

[25 Tutorialspoint, “Spring Tutorial,” [Online]. Available:
] <https://www.tutorialspoint.com/spring/index.htm>. [Accessed 06 05 2021].

[26 Tutorialspoint, “Spring - MVC Framework,” [Online]. Available:
] https://www.tutorialspoint.com/spring/spring_web_mvc_framework.htm. [Accessed 07
05 2021].

[27 Spring, “Spring overview,” [Online]. Available: [https://docs.spring.io/spring-](https://docs.spring.io/spring-framework/docs/current/reference/html/overview.html#overview)
] [framework/docs/current/reference/html/overview.html#overview](https://docs.spring.io/spring-framework/docs/current/reference/html/overview.html#overview). [Accessed 02 05 2021].

[28 Spring, “Introduction to the Spring Framework,” [Online]. Available:
] [https://docs.spring.io/spring-framework/docs/4.3.20.RELEASE/spring-framework-](https://docs.spring.io/spring-framework/docs/4.3.20.RELEASE/spring-framework-reference/html/overview.html)
[reference/html/overview.html](https://docs.spring.io/spring-framework/docs/4.3.20.RELEASE/spring-framework-reference/html/overview.html). [Accessed 07 05 2021].

[29 Vectr, “Vectr website,” [Online]. Available: <https://vectr.com/>. [Accessed 30 04 2021].
]

[30 B. Krebs, “Automatically Mapping DTO to Entity on Spring Boot APIs,” 05 09 2017.] [Online]. Available: <https://auth0.com/blog/automatically-mapping-dto-to-entity-on-spring-boot-apis/>. [Accessed 07 05 2021].

[31 B. Nadel, “Application Services vs. Infrastructure Services vs. Domain Services,” 06 06] 2012. [Online]. Available: <https://www.bennadel.com/blog/2385-application-services-vs-infrastructure-services-vs-domain-services.htm>. [Accessed 07 05 2021].

[32 P. Kainulainen, “Understanding Spring Web Application Architecture: The Classic Way,”] 19 10 2014. [Online]. Available: <https://www.petrikainulainen.net/software-development/design/understanding-spring-web-application-architecture-the-classic-way/>. [Accessed 07 05 2021].

Appendix

I. Source code

The source code can be found in thesis github repository:

<https://github.com/AnneliKlomas/stirrups-drawing-application>

For running frontend the backend solution should be running and have the imported data. The data can be found in the file imports.txt. For importing data to the dataset go to <http://localhost:8080/h2-console> click “Connect”, copy paste the content of the imports.txt to the SQL statements field.

II. Licence

I, Anneli Klamas,

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to

reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright,

Reinforcement stirrups drawing web application for Eesti Traat,

supervised by Manuel Alejandro Camargo Chávez.

2. I grant the University of Tartu a permit to make the work specified in p. 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 3.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.
3. I am aware of the fact that the author retains the rights specified in p. 1 and 2.
4. I certify that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Anneli Klamas

07/05/2021